DOI: 10.17323/2587-814X.2024.3.70.86

Designing a multi-agent system for a network enterprise

Yury F. Telnov
E-mail: Telnov.YUF@rea.ru

Vasiliy A. Kazakov
E-mail: Kazakov.VA@rea.ru

Andrey V. Danilov
E-mail: Danilov.AV@rea.ru

Plekhanov Russian University of Economics, Moscow, Russia

Abstract

The necessity to enhance the efficiency of modern network enterprises based on digital platform technologies, Digital Twins, and Digital Threads determines the relevance of implementing dynamic multi-agent technologies in production practice. The architectural complexity of existing multi-agent systems (MAS) and the lack of scientific research in the field of justifying methods and tools for their creation motivate the goal of this study to develop a comprehensive MAS design technology. This technology should encompass all architectural levels and allow for the adaptation of reference and best design practices. This article analyzes the possibilities of applying Digital Twins and Digital Threads in the creation of network enterprises and proposes methods for their implementation using MAS. A design technology for MAS has been developed in accordance with the IIRA (Industrial Internet Reference Architecture) and RAMI (Reference Architectural Model Industrie 4.0) architectural frameworks, which enables the interconnected formation and display of design results across various architectural levels. At the business level, a method is proposed for formulating business requirements for MAS based on the selection and adaptation of business models and application scenarios. At the level of constructing manufacturing and business processes, a method for formulating functional requirements for MAS is presented, revealing the transition from value networks to manufacturing and business process structures. At the level of functional design of the network enterprise's multi-agent system, a method is proposed for forming key design solutions from the perspective of implementing various service categories using AAS (Asset Administrative Shells) and their specialization. At the technological implementation design level of MAS, a method for implementing software agents using a microservice software organization is proposed. The method presented for adapting reference and best MAS design models allows for the selection of appropriate design solutions from libraries of reference models and knowledge bases for subsequent refinement. This accelerates and improves the quality of the design process. The implementation of the developed technology for designing multi-agent systems will increase the adaptability of network enterprises to dynamically changing business needs, taking into account the interests and capabilities of all stakeholders.

Keywords: multi-agent systems, digital threads, digital twins, network enterprise, asset administrative shell, project ontology, microservice architecture

Citation: Telnov Yu.F., Kazakov V.A., Danilov A.V. (2024) Designing a multi-agent system for a network enterprise. *Business Informatics*, vol. 18, no. 3, pp. 70–86. DOI: 10.17323/2587-814X.2024.3.70.86

Introduction

To produce innovative products and services for the specific needs of customers, dynamically formed flexible network or Internet-distributed enterprises can be created, requiring the implementation of new management systems based on the use of modern digital and intelligent technologies. As a result of the creation of such enterprises, the life cycle of products and services provided should be shortened both at the stage of launching into the market and at the production stage, ensuring high quality and adaptability of product configurations for various categories of consumers [1].

The creation of network or virtual enterprises based on modern digital platforms, on the one hand, leads to an increase in the level of integration and cooperation of enterprises interacting within the overall network structure [2], and on the other hand, gives rise to new tasks of creating mechanisms for coordinating participants in network enterprises and selecting reliable partners, organizing joint ownership and determining the rights to use data, the unresolved nature of which can lead to a loss of confidence of potential participants in a network enterprise regarding the possibility of joint activities within the overall network structure [3]. Solving the problems of creating network enterprises in industry becomes more complicated, as a rule, due to a large number of cooperative connections, high resource intensity and large investment cycles.

The implementation of Industrial Internet of Things (IIoT) technologies, Digital Twins and Threads into the practice of digital transformation of enterprises based on the concept of the Industrie 4.0 creates objective prerequisites for increasing the efficiency of managing network interactions of enterprises carrying out joint activities through the creation of modern digital systems [4–7]. At the same time, Digital Twins and Threads are based on digital models which represent systems of mathematical and computer models that make it possible to display the information state, predict the behavior of simulated objects in real time and formulate decisions.

One of the effective approaches to the implementation of the listed technologies is the creation of digital systems for managing manufacturing and business processes of a network enterprise based on the use of multi-agent technologies – multi-agent systems (MAS) of a network enterprise. In works [8, 9] developed computer models based on the use of agent-based and discrete-event modeling methods which are built into the structure of Digital Twins to optimize production processes at various stages of the life cycle. To increase the efficiency of using computer models of manufacturing and business processes, it is necessary to ensure their interaction with systems for operational data collection using the IIoT, as well as their integration with other intelligent technologies based on decision-making rules, analysis of big data and machine learning [10–13].

The architectural complexity of the MAS determines the goal of the study to develop a comprehensive technology for designing a software implementation that would affect all levels of the network enterprise architecture in accordance with the IIRA and RAMI architectural frameworks [14, 15] and the use of a variety of decision support tools in computer models of agents. Existing works on the design of multi-agent implementation of digital systems mainly consider the functional level of design [16–19] and practically do not consider the design of MAS at other architectural levels.

This article solves the problems of analyzing the capabilities of Digital Twins and Digital Threads for creating network enterprises, their implementation using an MAS and developing MAS design technology at the levels of business and user requirements, functional design and implementation with mutual mapping between the levels of the results obtained. A feature of the proposed technology is the linking of MAS design stages by sequential mapping of design entities (categories) between architectural levels and adaptation of design solutions based on libraries of reference solutions and knowledge bases of the best usage precedents [20–22].

1. Analysis of the possibilities of using Digital Twins and Digital Threads to create network enterprises based on multi-agent technology

Digital twin technology is widely used in industry and allows you to manage enterprise assets (products, equipment, any resources) at different stages of their life cycle. At the same time, Digital twins not only reflect the current state of assets, but also allow, using a set of procedures, to model, predict and formulate decisions to optimize their behavior. From this point of view, Digital twins are an integrated system of data, models and tools for analysis and decision-making used throughout the entire life cycle of various assets [8, 23].

Due to the need to track and manage the behavior of not only individual assets, but also the dynamic processes in which they participate, there is an objective need to implement more complex production technologies based on digitalization which are reflected in the concept of Digital Threads. The Digital Thread concept involves the use of modern modeling and management tools that link the life cycle processes of interconnected assets and make it possible to improve the manufacturability, controllability and sustainability of production systems [24]. Digital Thread in the economic sense implements value chain management. In the RAMI architectural framework [15], Digital Thread is associated with interacting assets: production chains and supply chains.

The use of Digital Twin and Digital Thread provides the flexibility and adaptability needed to quickly develop and implement products while reducing risk. Thus, data obtained from existing or designed production systems can form the basis of improved models that will allow forecasting at both the component level and the asset level as a whole. Archiving digital asset descriptions can greatly facilitate any future required redesign of the production system. The combination of Digital Twins and Digital Thread constitutes a digital system for a specific production system or an entire network enterprise.

A unified information model of Digital Twin and Digital Thread can be represented using software open source, which allows for the implementation of digital technologies in complex projects [25]. This option ensures the implementation of Digital Twin and Digital Thread not only in digitalization projects of individual enterprises, but also in the creation of network enterprises through the integration of various software systems based on a single digital platform. Using Digital Twin and Digital Thread technologies, information is transferred from individual links in the value chain to the production system of a network enterprise, which makes it possible to monitor compliance with requirements and the impact of the results obtained on the efficiency of the entire network enterprise [26].

The dynamic nature of Digital Twin and Digital Thread technologies makes it natural to use multi-agent systems to organize the interaction of Digital Twins within the Digital Thread, while the tool for implementing Digital Twins are software agents, and the Digital Thread is the MAS as a whole. The issues of implementing production systems using multi-agent interaction technologies are quite well theoretically worked out [9, 10, 13, 16, 17, 27].

The modern development of Industrie 4.0 concepts, which has led to new architectures for Digital Twins organizing in the form of Asset Administrative Shells operating on common digital platforms, provides the opportunity to develop MAS on a new technological basis, primarily using microservice implementation of mechanisms for performing agent functions [18, 19, 28].

In the listed works, the main emphasis is on the functional implementation of MAS for creating digital and network enterprises and to a lesser extent devoted to the issues of their design technology. At the same time, the construction of multi-agent systems for interaction between participants in a network enterprise makes relevant the issues of creating an MAS design technology that takes into account the level of complexity of network enterprises being created. Previously developed MAS design technologies, for example, such as ASEME [29], RTMIAS [30], X-Machine [31], etc., are local in nature, based on component technology of object-oriented design, and in some cases use ontologies for developing interaction between agents, but for such complex systems as network enterprises, they are of little use.

From the point of view of applying the concept of Industrie 4.0, the design of the MAS of a network enterprise comes down to the design of Asset Administrative Shells (AAS), which correspond to agents that implement active and proactive modes of operation [21, 28, 32], as well as designing scenarios for their interaction within the unified Digital Thread through the exchange of messages between AAS in accordance with the protocols established by FIPA [33]. A distinctive feature of the implementation of AAS on the principles of multi-agent technology is the possibility of using knowledge bases to develop solutions, machine learning and simulation mechanisms for analyzing and interpreting events.

To implement software agents and their interaction, it is necessary to develop a set of functional AAS services, and for a digital platform that serves many interacting AAS, a set of infrastructure services that ensure the creation and registration of AAS and their users, commercial and information security of use and a number of other functions.

Taking into account the complexity of the process of creating an MAS of a network enterprise, this article proposes a design technology that is based on the consistent refinement of design solutions at the levels of architecture of a network enterprise in accordance with IIRA and RAMI frameworks [14, 15] using the method for adapting reference and best design models.

2. Stages of technology for designing an MAS for a network enterprise

The stages of designing an MAS for a network enterprise are well defined by points of view on the system architecture in accordance with IIRA [14]:

- Business Modeling – Business viewpoint;
- Design of manufacturing and business processes – Usage viewpoint;

- Functional design of MAS – Functional viewpoint;
- Design of implementation technology – Implementation viewpoint.

The listed points of view on architecture or architecture levels are interrelated: each subsequent level of architecture specifies the previous level of architecture in its own specific language and confirms the possibility of implementing the requirements formulated above. A feature of the proposed MAS design technology is the sequential decomposition and detailing of design solutions and iterative repetition of stages if the need arises.

The set of digital system entities used in the proposed MAC design technology at various levels of the IIRA architecture, as well as design tools, is presented in *Table 1.*

Table 1.

Entities and tools for MAS design at various levels of IIRA architectural framework

Entity	Levels of MAS architecture of a network enterprise			
	Business level	Usage level	Functional level	Implementation level
Subject	Participant, Role	Performer role	Interface (Boundary Class)	Client application
			Subject agent (AAS)	Composite microservice
Object	Assets	Product, Resource	Object agent (AAS)	Composite microservice
			Entity class	Database
Function	Activity / Activities	Process	Operation (method, service)	Microservice
			Entity class	Microservices registry
Modeling and Design Tools				
Model type	Value Network (St. Gallen); Use-Case diagram	Activity diagram; BPMN diagram; Usage Model (IIRA)	UML diagrams: Sequence; Class; State-chart	UML diagrams: Component; Deployment

By Value Network (network enterprise) participant we mean any enterprises or organizations participating in the value chain as subjects.

By Role we mean a specific behavior scenario performed by a value network participant in manufacturing or business processes. Moreover, the same participant can play different roles in the same process; for example, an enterprise can be a supplier of equipment and a service provider, and an operating enterprise can also be an operator of a private cloud infrastructure. Roles at the usage level can be refined to the level of specific performers.

At the functional level, the role is performed directly either by an actor (organizational unit) or by a software agent (subject). In the first case, an interface must be created for the subject, implemented in the form of a client application, through which it interacts with the MAS. In the second case, a Digital Twin is created for the subject which automates a number of functions of the actor, essentially replacing it. In this regard, it is proposed to represent the subject's Digital twin at the functional level in the same way as the asset's Digital twin – in the form of AAS, which is implemented by a composite microservice in the sense that a composite software component includes a registry of services corresponding to individual functions [34].

In accordance with RAMI, an Asset of a network enterprise will be understood as any physical or software objects (products – goods, services or their components, and resources – individual devices, equipment, production lines, production systems) [15], which are represented by AAS [33]. AAS of objects (products or resources) are proposed to be implemented using microservices [35, 36]. If any part of the information about assets does not require active management, then it is technologically implemented in the form of a passive database which is local for microservice or a database which is shared across multiple microservices.

From a business point of view, the Activity of a network enterprise will be understood as a certain function that produces flows of material assets, information and value (costs). Each activity is detailed as a process executed by the MAS, which consists of individual operations. The identified operations are specified as elements of the AAS structure which link through API to an implementation in the form of a microservice. In this case, the meta-description of the operation as an entity class is placed in the service registry [21, 34].

A combination of different tools is used to model and design MAS. This article uses the object-oriented modeling language UML as a comprehensive tool to present an end-to-end example of the MAS design process.

Let us consider the stages of technology for designing an MAS for a network enterprise, corresponding to IIRA viewpoints, in more detail.

Business modeling. Business Viewpoint determines the strategy for creating and operating a network enterprise. From this position, at the business modeling stage, stakeholders and their vision of the functioning of the enterprise in the context of the use of common digital platforms, as well as the values and goals of digitalization of manufacturing and business processes are determined. At this stage, business requirements for the designed multi-agent system are formed. The most important role at this stage is played by the design of business models for the functioning of network enterprises, specified in business scenarios [21, 37]. The St. Gallen value network framework is widely used as a notation for describing business models [38]. From the perspective of multi-agent implementation, the method of determining the main activities and their actors, which are subsequently supported by software agents and their processes, becomes of utmost importance. The core activities of value networks can be captured using UML use case diagrams.

The business modeling process begins with SWOT analysis of the proposed network enterprise organization. This identifies the strengths and weaknesses of digitalization from the perspective of using internal resources, as well as opportunities and threats from the perspective of the influence of the external environment. As a result of SWOT analysis, the company's vision, the main values being formed are determined and a tree of goals is built. For goals, sets of measures are formed to achieve them, including the creation of a new or customization of an existing software and hardware platform.

The selection of business models and corresponding application implementation scenarios is carried out according to the methodology described in [22, 39]. In accordance with this methodology, depending on the type of business model (the Industrial Internet of Things platform model, the Value adding services in operation model, the Data Trustee model) and the type of business process (product lifecycle management processes, production system lifecycle management processes, supply chain management, asset service management) applied scenarios for the implementation of a network enterprise are selected (Adaptive Factory, Value Creation Network "Innovative Product Development", Value Creation Network "Order-Controlled Production", Value-Based Scenario, etc.). The selection of an application scenario for the implementation of a network enterprise is carried out on the basis of a knowledge base of typical scenario models, organized using the ontology of digital transformation, and subsequent analysis of network effects.

The selected application scenario is presented in the form of a value network model and is adapted to the operating conditions of a particular enterprise. In the value network model, the composition of enterprise participants and their roles is first determined (parent enterprise, subcontractors, functional service providers, platform operators, software developers, system integrators, etc.). The performance of activities by participants in a network enterprise in accordance with their roles can be represented in the form of a use case diagram. An example of a use case diagram for a value-based application scenario [40] is presented in *Fig. 1.*

Design of manufacturing and business processes in accordance with the point of view of system use (Usage Viewpoint). At this stage, functional requirements for the organization of manufacturing and business processes are determined in terms of specifying participants in network enterprises and their roles in various activities (processes). In this case, activities are determined from the perspective of initiation conditions, task workflow, resulting effects and restrictions on the execution of processes, and executor roles are assigned to tasks [37, 40].

The transition from the value network model to models of manufacturing and business processes is carried out in accordance with the following method: each value flow and accompanying information and financial flows correspond to manufacturing or business processes, which can be modeled in the form of an activity diagram or BPMN diagram reflecting the assignment of roles process participants for performing specific operations (services). As a result of activity diagrams design, functional requirements are formed for the future composition of software agents implementing the allocated roles, and the composition of operations (services) of the corresponding AAS.

Functional design of MAS, reflecting a functional point of view on the system architecture (Functional viewpoint). At this stage, a method is presented for the formation of basic design solutions for building an MAS of a network enterprise from the position of implementing various categories of services using AAS and

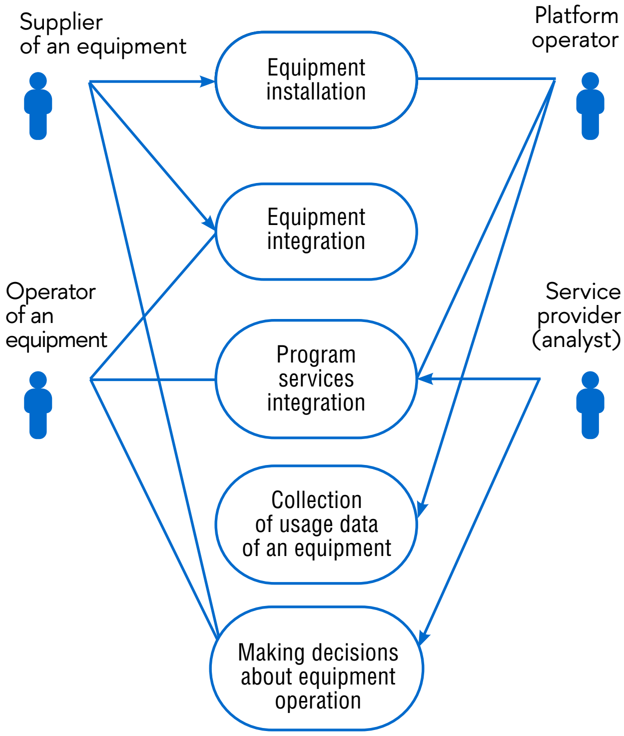

Fig. 1. Example of a use-case diagram for a value-based scenario (VBS).

their specialization: interaction with external business services (ERP, MES, PLM, etc.); applications (Application) – functional services of digital systems; system services (System Management) – platform services; asset integration services – physical devices, products (Control); communication services of digital system components (*Table 2*) [14, 34].

From a functional viewpoint, in accordance with the concept of the Platform Industrie 4.0, each component of a network enterprise represents an asset and its AAS [41, 42]. The structure of AAS includes a set of properties, operations and events which can also be divided into submodels [42]. At the same time, the information part (passive) of AAS in the form of a set of properties allows you to reflect the dynamic information model of the asset, and the operational (active) part allows you to interact with assets, other AAS and external applications, and to execute functional services. In accordance with the types of assets, software

Table 2.

Sublevels of the IIRA functional level

Functional level sublevels	Description
Business level (BUSINESS DOMAIN)	Services for exchanging information with external applications of various information systems: ERP, CRM, PLM, MES, HRM, etc.
Application level (APPLICATION DOMAIN)	Functional services performing the functions of monitoring, diagnostics, forecasting, coordination, optimization, support, management of manufacturing and business processes
Operating level (SYSTEM MANAGEMENT DOMAIN)	General infrastructure services for managing a software system (platform): deployment, configuration, monitoring, diagnostics and updating of its components, as well as orchestration for coordinating the operation of various system components
Information level (INFORMATION DOMAIN)	Services for collecting, cleaning, syntactic and semantic transformations, storing and issuing data for functional components. Auxiliary data management services: data security, data access control and data rights management, backup and recovery, etc. Services that implement universal methods of data analysis, incl. Big Data analysis methods, machine learning, simulation, knowledge extraction
Communication level	Considered between services (components) at all sublevels
Control level (CONTROL DOMAIN)	Services interaction of software components with physical devices (assets): collecting data using sensors from physical devices, monitoring operation and executing control commands on physical devices.
Physical systems	Physical devices: equipment parts, equipment, production lines, production systems (factories), network enterprises (Connection World) Software: software libraries, knowledge bases, ontologies, repositories

agents that are represented by AAS are divided into product agents and resource (equipment) agents [43]. A digital platform can be represented by AAS with a set of system services, the actor for which is the operator (administrator) of the platform. Digital Twins of subjects managing the production system are also represented by agents – AAS.

AAS as software agents are well represented by classes in the object-oriented paradigm. The interaction of software agents in the process of executing individual activities with the addition of the necessary interface classes (Boundary) and entity classes associated with the database (Entity) is displayed in the form of a sequence diagram (Sequence Diagram). *Figure 2* shows a fragment of a sequence diagram reflecting the process of generating recommendations for equipment operation in accordance with a Value-Based Scenario.

The process is launched through the boundary class – client application "Data Analysis" (Boundary), and the received recommendation is recorded in the local database through the entity class "Recommendation" (Entity).

On the generated class diagram, AAS "Analyst Agent" implements the functions of a software agent at the Functional Domain level. The service (operation) "Perform Forecast" using one of the machine learning methods, for example, a neural network, performs fault prediction, and the service "Form Recommendation" using a set of rules and/or a simulation model based on agent-based and discrete-event modeling methods. implemented, for example, in the AnyLogic system [9].

Equipment Agent AAS is a software agent that collects data into a historical dataset to predict the condition and updates the condition of the asset. In this sense, AAS performs the functions of the control level (Control Domain). Information services for converting data formats, checking access security, etc., associated with the execution of the "Data Analysis" and "Save Recommendation" services, are called within these services. Similarly, the communication service of AAS "Analyst Agent" and AAS "Equipment Agent" is called within the "Prepare Dataset" service.

Design of technological implementation (Implementation viewpoint). The point of view on the implementation of a digital system reflects the physical construction of the system from the components being created. Considering the autonomy of the main components – software agents, their distribution in a computer network, the need for independent access to infrastructure services of a common digital platform, the presence of local databases as part of the components, it is proposed to implement MAS based on microservices technology in a cloud containerization environment. Architectural patterns can be used to provide examples and references for conceptualizing real-life IIoT architectures.

The use of microservices is proposed to be carried out at two levels: at the level of AAS in the form of composite microservices and at the level of microser-

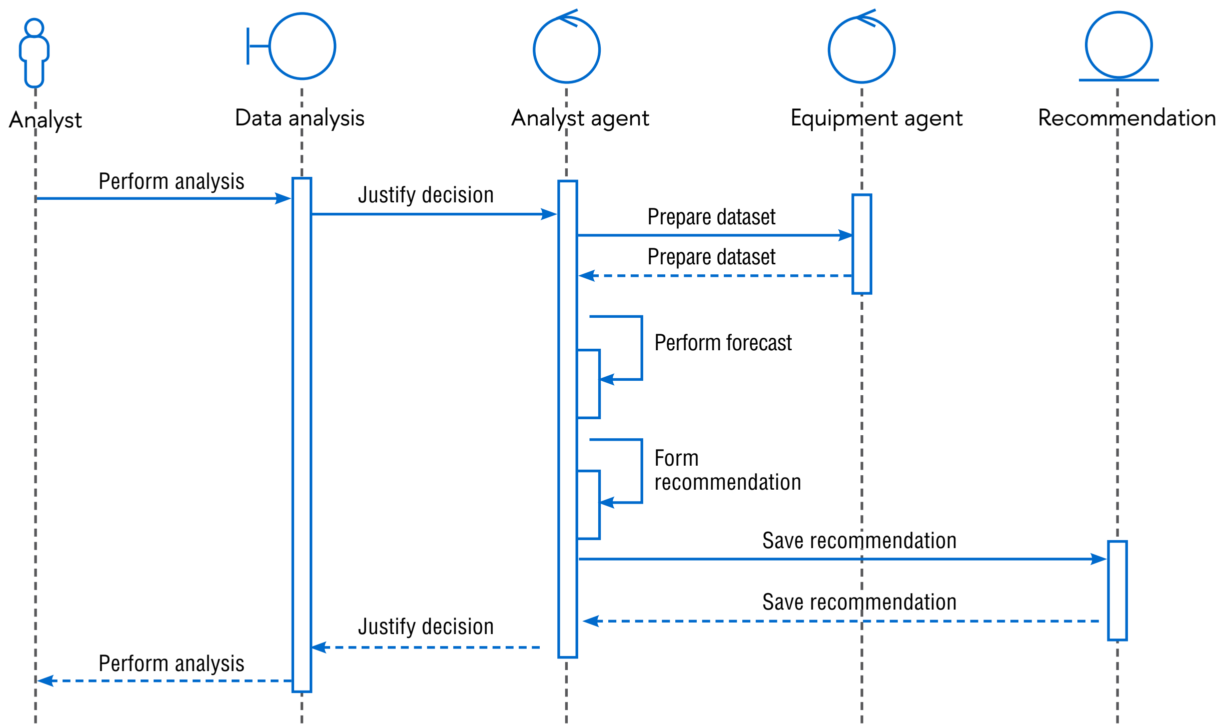

Fig. 2. Fragment of a sequence diagram for the process of recommendations formation for equipment operation.

vice implementation of operations (services, methods) of the AAS. In the second case, the operation in the information part of the AAS is connected via an API interface with a microservice which is stored in a dedicated library of the AAS, organized using a service registry (Registry) [34].

Integration and deployment of microservices is carried out on a technology platform for managing container microservices. An example of a deployment diagram of all software of a digital multi-agent system for the process of recommendation formation for equipment operation is presented in *Fig. 3*.

As an architectural pattern for implementing a network enterprise in the Industrial Internet concept (IIRA), we propose to use the Digital Twin pattern as an intermediate software layer (between the application and the physical world) [14]. This pattern implies the construction of industrial applications based on digital twins, which in turn are implemented based on standard services of the IIoT platform.

3. Method for adapting design models for a multi-agent system of a network enterprise

The MAS design technology is based on the use of a methodology for adapting design models (design patterns). For this purpose, knowledge bases (libraries) of design models are organized; they are systematized in accordance with a set of dictionaries (ontologies).

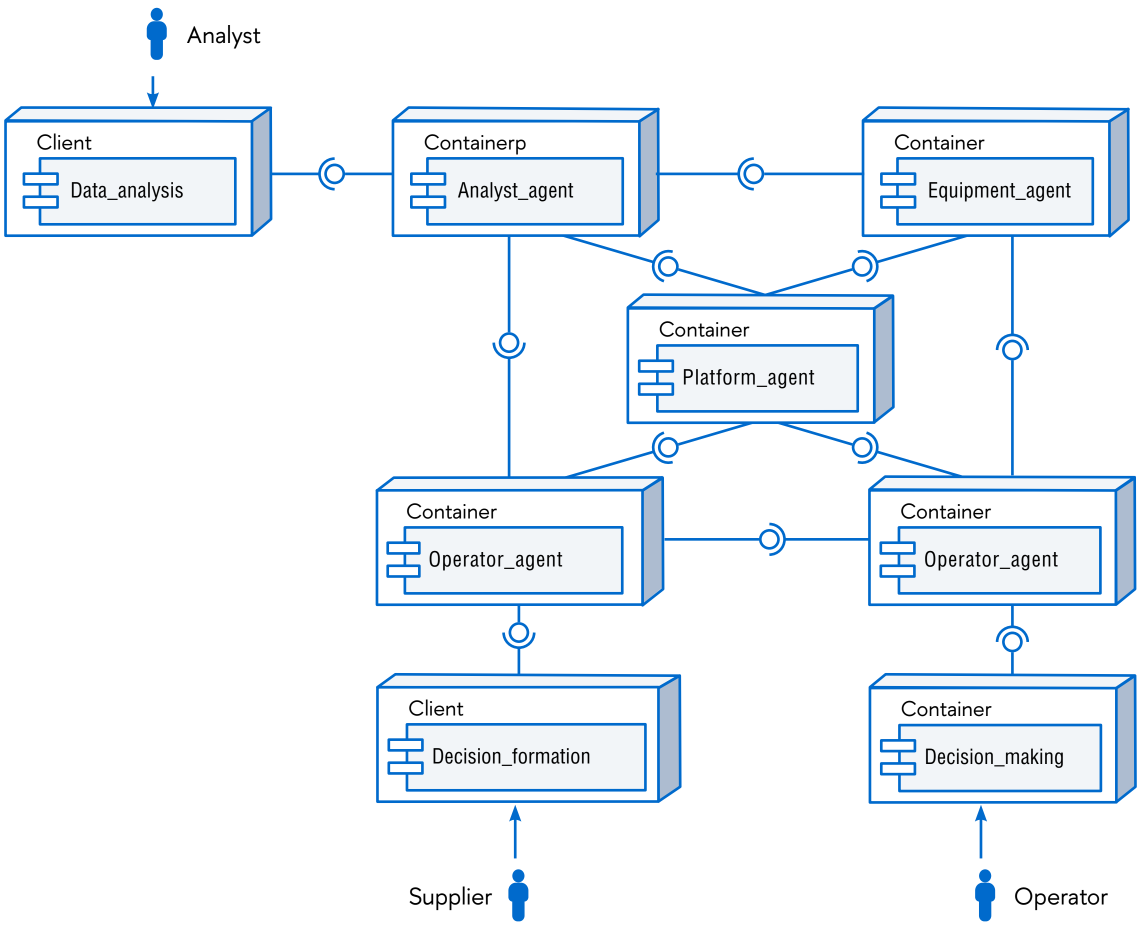

Fig. 3. MAS deployment diagram.

In the State Standard "Structure of the Digital Factory" [20], design patterns are specified in the asset class libraries of the digital factory, and a set of dictionaries is defined in accordance with which asset classes are built. In the materials of the Plattform Industrie 4.0 project [21], design models (patterns) are presented in a library of functional blocks. In [22], it was proposed to use not only reference models of application scenarios from the knowledge base, but also to accumulate models of precedents related to completed projects and use the hierarchy of ontologies to organize access to these models.

In this work, the approach to the use of MAS design patterns is developed from the point of view of their application at all stages of design technology. Method for adapting multi-agent system design models comes down to selecting precedents of appropriate models from libraries of reference models and knowledge bases and their subsequent refinement (*Fig. 4*).

To reflect the current state of the MAS creation project, a Project Repository is organized, which records the state of the project after each stage. At the same time, the Project Ontology is also clarified and developed.

Project organization at the initiation stage begins with defining the ontology of a network enterprise and selecting appropriate dictionaries from a set of ontologies that can be implemented by various standardization bodies, consortiums and research projects, and including upper ontologies, problem ontologies and domain ontologies. The selected ontologies form the prototype of the Project Ontology (or network enterprise ontology). In addition, as the project is implemented, the ontologies of external participants in the network enterprise are also connected to the existing ontology through links. A necessary condition for their unification is the alignment of external ontologies with the Project Ontology.

The subsequent development of the repository and ontology metadata involves versioning not only the data (a "snapshot" of the parameters of the system models is implemented as of a certain point in time), but also the ontology of the network enterprise (Project Ontology). This option allows you to use all the data accumulated in the historical perspective for the development of a project for the creation and operation of a network enterprise. Mechanisms that implement versioning, alignment and development of ontologies, their connection with each other, must be implemented as independent services within the platform that supports the functioning of a multi-agent system.

Each subsequent stage of designing a multi-agent system generates at the output a set of input parameters for the formation of the next stage, within which these input parameters serve as the basis for selecting new models from a library of reference models. According to the levels of model typification, which are stored in the library of reference models (design patterns), models can be both quite abstract, high-level, and specific to a particular subject area.

A distinctive feature of the proposed method for adapting MAS design models of a network enterprise is the use, along with a library of reference models (design patterns), also of a knowledge base aimed at preserving the formed and tested structures and descriptions of real business models, business scenarios, production and business processes, software agents (AAS), sets of services (precedent models).

Precedent models are presented as descriptions of cases that are stored in the system with reference to the description of the initial conditions (requirements), as well as the results of the work of the network enterprise, the quality characteristics of the products released (products) and the resulting economic effect. Thus, when searching for suitable models, not only reference models (design patterns of a certain type) are selected, but also adapted precedent models, taking into account the degree of proximity (maximum similarity value – S) of the corresponding characteristics of the models $M_i \in M$ and the available input and required output parameters that define the problem situation C^{In} [44]:

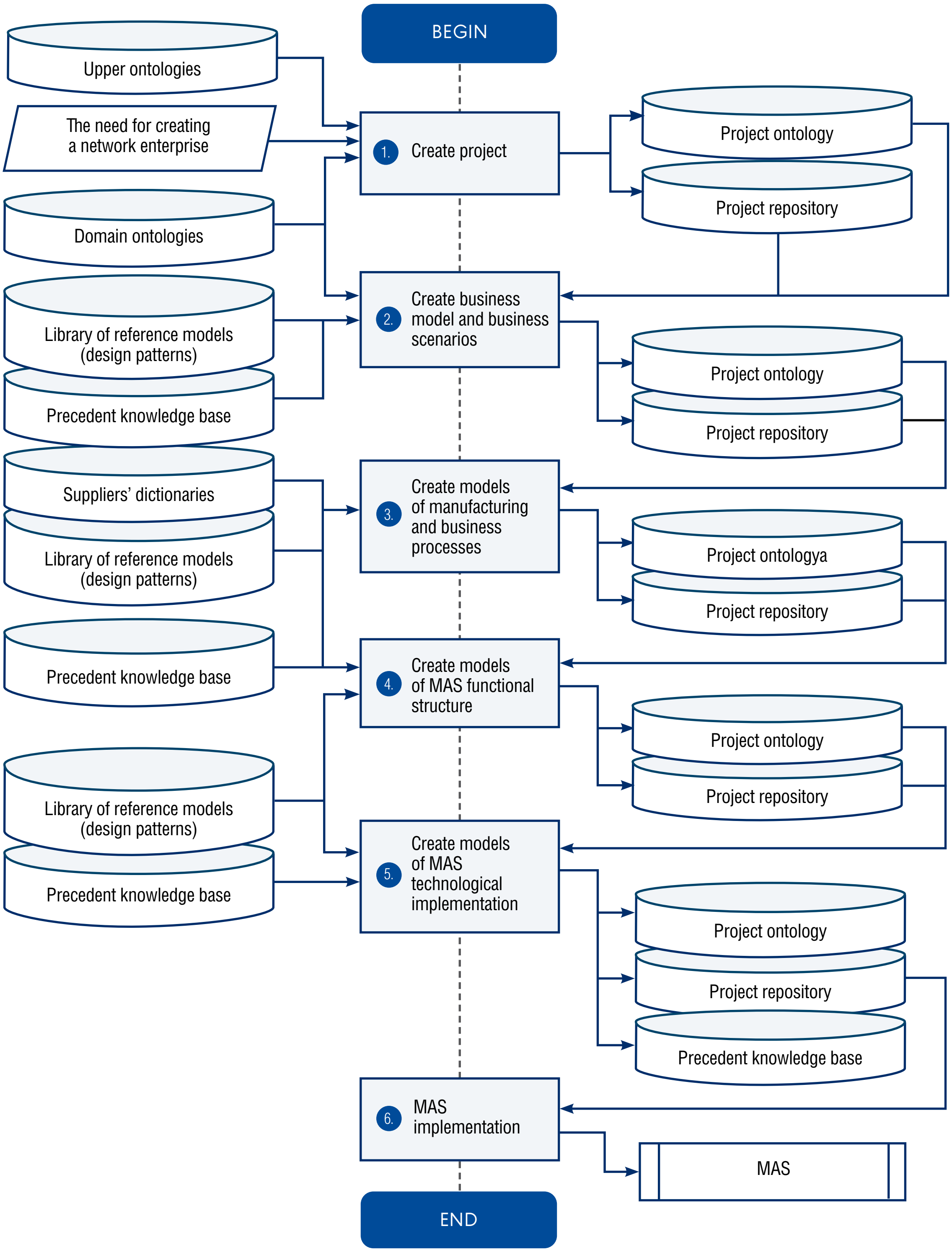

Fig. 4. Method for adapting MAS design patterns.

$$S(C^{In}, M) = \max_i \left(\frac{\sum_{j=1}^{n} w_j \cdot \mathrm{sim}(f_j^{In},\ f_{ij}^{M})}{\sum_{j=1}^{n} w_j} \right),$$

where

$i = \overline{1, N}$, where N is the total number of models available in the library and Precedent Knowledge Base;

$j = \overline{1, CR}$, where CR is a constant number of compared description elements (properties, relationships);

f_{ij}^{M} – j-th property (relationship) describing the i-th model,

f_j^{In} – j-th property (relationship), describing the set of required input or output parameters,

w_j – weight of the j-th property (relationship),

sim – similarity function f_j^{In}, f_{ij}^{M}.

After completing the design process of a multi-agent system, the design results reflected in the Project Repository are used to perform the following stages of creating an MAS: software development, testing and implementation. The result of the design process is also placed in the Precedent Knowledge Base for later use in other projects.

Conclusion

As a result of the study, we can conclude that the use of MAS fully ensures the creation of effective network enterprises based on the implementation of the principles of the Industrie 4.0 and the use of digital twins and digital thread, providing information collection, modeling and planning of asset behavior, organization and control of manufacturing and business processes.

The proposed methods and technology for designing MAS in accordance with IIRA and RAMI architectural frameworks provide, at the business and usage levels, for the construction of basic application scenarios for the use of MAS and the roles of actor-agents, the formation of structures of manufacturing and business processes; at the functional level – building a set of functional components in the form of AAS and models of their interaction in a common information space; at the technological level of implementation – adaptation of microservice implementation templates to the specific conditions of building a network enterprise.

A distinctive feature of the design technology so developed is the interconnected representation of all used categories (entities) at various levels of the architecture, which allows for a coordinated transition between the stages of MAS design.

The complex nature of the proposed technology for designing MAS interaction between participants in a network enterprise determines the organization of effective participation of all stakeholders in the creation of a network enterprise focused on the implementation of a business strategy, taking into account the adaptation of reference and best models of application scenarios, functional components and microservice structures using design patterns libraries, knowledge bases and ontologies. Consistent display of design results between different levels of architecture in the project repository allows you to fully implement functional and non-functional requirements, taking into account the available information and computing resources.

The proposed methodology for adapting design models for a multi-agent system of a network enterprise develops an approach to adapting MAS templates from libraries of reference solutions and knowledge bases of the best precedents for use at all stages of design technology.

The application of the presented technology for designing an MAS of a network enterprise will help improve the level of quality and reliability of the functioning of a network enterprise, adaptability to dynamically changing business needs and the capabilities of all stakeholders. ■

Acknowledgements

The research was supported by a grant from the Russian Science Foundation (project no. 22-11-00282[1]).

[1] https://rscf.ru/project/22-11-00282/

References

1. Matthyssens P. (2019) Reconceptualizing value innovation for Industry 4.0 and the Industrial Internet of Things. *Journal of Business & Industrial Marketing*, vol. 34, no. 6, pp. 1203–1209. https://doi.org/10.1108/JBIM-11-2018-0348

2. Feofanov A.N. Bondarchuk E.Yu., Tyasto S.A. (2018) Organization of a virtual enterprise – the future of production. *Bulletin of MSTU "Stankin"*, no. 3 (46), pp. 101–105 (in Russian).

3. Müller J.M. (2019) Antecedents to digital platform usage in Industry 4.0 by established manufacturers. *Sustainability*, vol. 11, no. 4, article 1121. https://doi.org/10.3390/su11041121

4. Golovin S.A., Lotsmanov A.N., Pozdneev B.M (2021) The Russian Federation Industry 4.0 program is a chance not to fall behind forever in the field of industrial production. *World of Information Technologies*, no. 1–2, pp. 38–40 (in Russian).

5. Borovkov A.I., Prokhorov A., Lysachev M. (2020) *Digital twin. Analysis, trends, world experience*. Moscow: Alliance Print (in Russian).

6. Rosstandart (2021) *National Standard of the Russian Federation GOST R 57700.37–2021. Computer models and simulation. Digital twins of products. General provisions* (in Russian).

7. National Institute of Standards and Technology (2018) *Digital thread for smart manufacturing*. Available at: https://www.nist.gov/programs-projects/digital-thread-smart-manufacturing (accessed 1 August 2024).

8. Makarov V.L., Bakhtizin A.R., Beklaryan G.L. (2019) Developing digital twins for production enterprises. *Business Informatics*, vol. 14, no. 1, pp. 7–16. http://doi.org/10.17323/1998-0663.2019.4.7.16

9. Makarov V.L., Bakhtizin A.R., Beklaryan G.L., Akopov A.S. (2021) Digital plant: methods of discrete-event modeling and optimization of production characteristics. *Business Informatics*, vol. 15, no. 2, pp. 7–20. http://doi.org/10.17323/2587-814X.2021.2.7.20

10 Gorodetsky V.I. (2019) Behavioral models of cyberphysical systems and group management. Basic concepts. *News of the Southern Federal University. Technical Sciences*, no. 1 (203), pp. 144–162.

11 Corsini R.R., Costa A., Fichera S., Framinan J.M. (2024) Digital twin model with machine learning and optimization for resilient production–distribution systems under disruptions. *Computers & Industrial Engineering*, vol. 191, article 110145.

12. Kabaldin Yu.G., Shatagin D.A., Anosov M.S., Kuzmishina A.M. (2019) Development of digital twin of CNC unit based on machine learning methods. *Vestnik of Don State Technical University*, vol. 19, no. 1, pp. 45–55 (in Russian). https://doi.org/10.23947/1992-5980-2019-19-1-45-55

13. Skobelev P., Mayorov I., Simonova E., Goryanin O., Zhilyaev A., Tabachinskiy A., Yalovenko V. (2020) Development of models and methods for creating a digital twin of plants within the cyber-physical system for precision farming management. *Journal of Physics: Conference Series*, vol. 1703, pp. 12–22. https://doi.org/10.1088/1742-6596/1703/1/012022

14. Industry IoT Consortium (2022) *The industrial internet reference architecture*. Available at: https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/11/IIRA-v1.10.pdf (accessed 1 August 2024).

15. Plattform Industrie 4.0 (2018) *Plattform Industry 4.0. Reference architectural model Industry 4.0 (RAMI4.0) – An introduction*. Available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.html (accessed 1 August 2024).

16. Seitz M., Gehlhoff F., Cruz Salazar L.A., Fay A., Vogel-Heuser B. (2021) Automation platform independent multi-agent system for robust networks of production resources in industry 4.0. *Journal of Intelligent Manufacturing,* vol. 32, pp. 2023–2041.

17. Karnouskos S., Leitao P., Ribeiro L., Colombo A.W. (2020) Industrial agents as a key enabler for realizing industrial cyber-physical systems: Multiagent systems entering Industry 4.0. *IEEE Industrial Electronics Magazine*, vol. 14, no. 3, pp. 18–32. https://doi.org/10.1109/MIE.2019.2962225

18. Vogel-Heuser B., Ocker F., Scheuer T. (2021) An approach for leveraging Digital Twins in agent-based production systems. *at – Automatisierungstechnik*, vol. 69, no. 12, pp. 1026–1039. https://doi.org/10.1515/auto-2021-0081

19. Telnov Yu.F., Kazakov V.A., Danilov A.V., Denisov A.A. (2022). Requirements for the software implementation of the Industrie 4.0 system for creating network enterprises. *Software & Systems*, vol. 35, no. 4, pp. 557–571 (in Russian).

20. Rosstandart (2022) *National Standard of the Russian Federation GOST R 70265.1–2022. Industrial-process measurement, control and automation. Digital factory framework. Part 1. Basic provisions* (in Russian).

21. Plattform Industrie 4.0 (2019) *Discussion Paper: Usage View of the Asset Administration Shell*. Available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/2019-usage-view-asset-administration-shell.html (accessed 1 August 2024).

22. Telnov Yu.F., Kazakov V.A., Bryzgalov A.A., Fiodorov I.G. (2023) Methods and models for substantiating application scenarios for the digitalization of manufacturing and business processes of network enterprises. *Business Informatics*, vol. 17, no. 4, pp. 73–93. http://doi.org/10.17323/2587-814X.2023.4.73.93

23. Segovia M., Garcia-Alfaro J. (2022) Design, modeling and implementation of digital twins. *Sensors*, vol. 22, no. 14, article 5396. https://doi.org/10.3390/s22145396

24. Bajaj M., Hedberg T. (2018) System lifecycle handler – Spinning a digital thread for manufacturing. *INCOSE International Symposium*, vol. 28, no. 1, pp. 1636–1650. https://doi.org/10.1002/j.2334-5837.2018.00573.x

25. Idaho National Laboratory (2020) *Deep-Lynx*. Available at: https://github.com/idaholab/Deep-Lynx (accessed 1 August 2024).

26. Bonham E., McMaster K., Thomson E., Panarotto M., Müller J.R., Isaksson O., Johansson E. (2020) Designing and integrating a digital thread system for customized additive manufacturing in multi-partner kayak production. *Systems*, vol. 8, no. 4, article 43. https://doi.org/10.3390/systems8040043

27. Tarassov V.B. (2019) Enterprise total agentification as a way to Industry 4.0: Forming artificial societies via goal-resource networks. Proceedings of the *Fourth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19). Advances in Intelligent Systems and Computing (AISC)*, vol. 1156, pp. 26–40.

28. Sakurada L., Leitao P., de la Prieta F. (2022) Agent-based asset administration shell approach for digitizing industrial assets. *IFAC-PapersOnLine*, vol. 55, no. 2, pp. 193–198.

29. Spanoudakis N.I., Moraitis P. (2007) The agent systems methodology (ASEME): A preliminary report. *Computer Science*.

30. Julian V., Botti V. (2004) Developing real-time multi-agent system. *Integrated Computer-Aided Engineering*, vol. 11, no. 2, pp. 135–149. https://doi.org /10.3233/ICA-2004-11204

31. Eleftherakis G., Kefalas P., Kehris E. (2011) A methodology for developing component-based agent focusing systems on component quality. Proceedings of the *Federated Conference on Computer Science and Information Systems (FedCSIS 2011), Szczecin, Poland, 18–21 September 2011*, pp. 561–568.

32. Plattform Industrie 4.0 (2020) *Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0*. Available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Digital-Twin-and-Asset-Administration-Shell-Concepts.pdf (accessed 1 August 2024).

33. Foundation for Intelligent Physical Agents (2002) *FIPA ACL message structure specification*. Available at: http://www.fipa.org/specs/fipa00061/SC00061G.pdf (accessed 1 August 2024).

34. Plattform Industrie 4.0 (2021) *Functional view of the asset administration shell in an Industrie 4.0 system environment*. Available at: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Functional-View.html (accessed 1 August 2024).

35. Lewis J., Fowler M. (2014) *Microservices. A definition of this new architectural term*. Available at: https://martinfowler.com/articles/microservices.html (accessed 1 August 2024).

36. Richardson C. (2018) *Microservices Patterns: With examples in Java*. Manning Publications.

37. Plattform Industrie 4.0 (2017) *Exemplification of the Industrie 4.0 application scenario value-based service following IIRA structure*. Available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/exemplification-i40-value-based-service.pdf (accessed 1 August 2024).

38. Gassmann O., Csik M., Frankenberg K. (2014) *The business model navigator: 55 models that will revolutionise your business*. FT Press.

39. Telnov Yu.F., Bryzgalov A.A., Kozyrev P.A., Koroleva D.S. (2022) Choosing the type of business model to implement the digital transformation strategy of a network enterprise. *Business Informatics*, vol. 16, no. 4, pp. 50–67. http://doi.org/10.17323/2587-814X.2022.4.50.67

40. Plattform Industrie 4.0 (2018) *Usage viewpoint of application scenario value-based service*. Available at: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/hm-2018-usage-viewpoint.html (accessed 1 August 2024).

41. Rosstandart (2021) *National Standard of the Russian Federation GOST R 59799–2021. Smart manufacturing. Reference architecture model industry 4.0 (RAMI 4.0)* (in Russian).

42. Plattform Industrie 4.0 (2022) *Details of the asset administration shell – Part 1. The exchange of information between partners in the value chain of Industrie 4.0*. Available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html (accessed 1 August 2024).

43. Telnov Yu.F., Kazakov V.A., Danilov A.V., Bryzgalov A.A. (2023) Network enterprises: Production and business process models based on multi-agent systems. *Software & Systems*, vol. 36, no. 4, pp. 632–643.

44. Kolodner J. (1993) *Case-based reasoning*. Morgan Kaufmann.

About the authors

Yury F. Telnov

Dr. Sci. (Econ.), Prof.;

Head of the Department of Applied Informatics and Information Security, Plekhanov Russian University of Economics, 36, Stremyanny Lane, Moscow 117997, Russia;

E-mail: Telnov.YUF@rea.ru

ORCID: 0000-0002-2983-8232

Vasiliy A. Kazakov

Cand. Sci. (Econ.);

Associate Professor, Department of Applied Informatics and Information Security, Plekhanov Russian University of Economics, 36, Stremyanny Lane, Moscow 117997, Russia;

E-mail: Kazakov.VA@rea.ru

ORCID: 0000-0001-8939-2087

Andrey V. Danilov

Senior Lecturer, Department of Applied Information Technologies and Information Security, Plekhanov Russian University of Economics, 36, Stremyanny Lane, Moscow 117997, Russia;

E-mail: Danilov.AV@rea.ru

ORCID: 0000-0002-0433-9701