

[DOI: 10.17323/2587-814X.2021.4.76.92](https://doi.org/10.17323/2587-814X.2021.4.76.92)

Technology for designing innovative processes for creating products and services of a network enterprise using an i4.0 knowledge-based system

Yury F. Telnov 

E-mail: Telnov.YUF@rea.ru

Vasily A. Kazakov 

E-mail: Kazakov.VA@rea.ru

Andrey V. Danilov 

E-mail: Danilov.AV@rea.ru

Plekhanov Russian University of Economics
Address: 36, Stremyanny Lane, Moscow 117997, Russia

Abstract

The creation of network enterprises based on the digital technologies of the Industrie 4.0 (the 4th Industrial Revolution, i4.0) opens broad opportunities for increasing production flexibility, customer focus and continuous innovation in products and services provided. At the same time, new opportunities necessitate the development of new methods and technologies for designing innovative processes in the context of digital i4.0 platforms, all of which highlights the relevance of the presented research topic. This work aims to define technologies for designing innovative processes to create products and services using i4.0 systems which are based on multi-agent interaction of asset administration shells (AAS), displaying digital twins of product components, and the use of ontological and cognitive methods for forming and justifying design decisions. The work presented here uses the Domain-Driven Design approach, an architectural framework for building i4.0 systems, methods of ontological engineering, quality function deployment (QFD), analysis of the types and consequences of potential inconsistencies (FMEA) and processing of fuzzy sets. The paper proposes principles for identifying bounded contexts of the domain under the design activities for the stages of the life cycle and products' subsystems (components). For bounded contexts of the domain, it is envisaged to create AAS of i4.0 systems, with the help of which the innovative process is supported and the multi-agent interaction of its

participants is carried out. As cognitive tools for making design decisions, we proposed to use services for assessing the importance of the determined quality characteristics of products and minimizing deviations of the proposed solutions from the formed functional and non-functional requirements. The methods of ontological engineering and data modelling allow us to dynamically develop an innovative project and support various versions of the project in the design process. Application of the proposed technology for designing innovative processes to create products and services at network enterprises using i4.0 systems will improve the quality of design decisions, increase the dynamism and continuous design of innovative projects.

Key words: innovative process; digital twin; domain-driven design; bounded context; value-added chain; ontology; multi-agent interaction; i4.0 system; i4.0 platform; asset administration shell (AAS).

Citation: Telnov Yu.F., Kazakov V.A., Danilov A.V. (2021) Technology for designing innovative processes for creating products and services of a network enterprise using an i4.0 knowledge-based system. *Business Informatics*, vol. 15, no 4, pp. 76–92. DOI: 10.17323/2587-814X.2021.4.76.92

Introduction

The innovative process for creating products and services in a network enterprise is an iterative process of designing a value-added chain in which the conceptual study of the product design is accompanied by the selection of project participants and agreeing with them the conditions for performing work at the implementation stage. The architecture of a knowledge-based system (KBS) that supports the innovative process for creating products and services was proposed by the authors in [1]. Its distinctive feature is the use of digital threads [2, 3] and digital twins [4–6] technologies for reflecting the design process in the system of product models and related processes, multi-agent technology for organizing the interaction of participants in the value-added chain [7–9] and the ontological approach [10–12] for automating the search for relevant information sources and organizing information exchange between software agents. For further development of technology to design innovative processes that create products and services, it is necessary to justify the choice of a software platform for multi-agent

implementation of a knowledge-based system and determine the design technology in more detail, considering the selected software implementation environment.

The methodological foundations of the conceptual design of such complex innovative processes of product and service design are laid down in the Domain-Driven Design approach [13, 14], which allows one to structure the subject area, considering its objective structure and subjective component that determines the organization of design teams working efficiently. In this regard, this article develops the application of the Domain-Driven Design approach to identifying bounded contexts of the innovative process under the stages of the life cycle and with the value-added chain for decomposing the process into separate components and establishing various types of interfaces between them following the needs of multi-agent interaction of the innovative process participants.

The multi-agent implementation of a knowledge-based system to support the innovative process for creating products and services is aimed at ensuring the stability of the entire structure and the autonomy and interoperabil-

ity of its components, distributed in a computing system, and operating on a single software platform. Since the proposed KBS implements the technology of digital threads and digital twins [1], the article discusses the use of the 4th generation industry software platform (the i4.0 platform) [15] and asset administration shells (AAS) [16, 17] to represent the bounded contexts of software agents corresponding to individual components of the value-added chain. At the same time, the composition of functional and platform (infrastructure) services necessary for the implementation of the innovative process to create products and services in a networked distributed environment is determined.

To develop mechanisms for supporting planning processes and organizing a value-added chain which is implemented through the interaction of software agents (AAS), the article proposes to use the ontological approach [11, 12] and cognitive approach [18, 19]. They both provide algorithms formalization for choosing business partners and coordination of conditions of interaction based on semantic search methods and fuzzy assessment of requirements correspondence and their fulfilment possibilities. The co-use of ontological and cognitive approaches will allow us to build effective value-added chains aimed at promising and executable product design implementation. The mechanisms of multi-agent interaction of business partners based on digital twins' technology on the i4.0 platform will reduce the time and increase the flexibility of designing innovative processes.

1. The domain-driven design approach for the structuring of innovative processes domain design

During the design of innovative processes to create products and services, there are problems of rational definition of the boundaries of project tasks, decision-making during the for-

mation of project groups, organization of interaction between project groups. As a methodology for designing innovative processes to create products and services, it is proposed to use the domain-driven design approach that is best focused on solving cognitive problems in project teams that arise during the coordination and justification of project solutions [13, 14].

The domain-driven design approach to innovative design aims at conducting an effective decomposition of the design process according to the subject and contextual (cognitive) principles and allows one to develop a common language of interaction in project groups of diverse specialists. From our point of view, this language and corresponding design solution concepts can be reflected in computer-supported models in the form of ontologies.

At its core, the domain-driven design approach is designed for the development of a special type of innovative product (software) and the use of flexible iterative software development technologies in which each iteration involves the creation of a working version of the software. Considering that modern production of conventional products is becoming digital (that is, products, on the one hand, are saturated with embedded software, and on the other hand, are supported by remote computer models – digital twins) then the principles of domain-driven design can be fully extended to traditional products and services with a cascade (sequential) development model.

The decomposition and modelling of the subject area for the innovative processes design are considered from two points of view: from the side of individual subsystems, the so-called subdomains, and from the point of view of solving individual tasks, the so-called bounded contexts, for the joint solution of which project groups are created.

From the point of view of the allocation of subsystems (subdomains) and following modelling, the functional principle of highlighting

the functions that the product should perform is usually applied; cross-functional areas of integration can be distinguished in addition. Often, the functional subsystems development results in the corresponding physical parts of the product development. For example, for a car design, the chassis, fuel system, electrical equipment, security system and other car subsystems are allocated as subdomains, which can be divided into components corresponding to individual nodes or parts. In *Figure 1*, subsystems and components (subdomains) are shown with rounded rectangles, and the arrows reflect the decomposition order of the subject area.

Bounded contexts are associated with specific tasks to be solved and have clearly defined boundaries. Specialists, primarily of two types, are involved in solving innovative tasks in a bounded context: developers and domain experts. Depending on the task complexity, developers and experts of different profiles can be involved in the innovative process.

In the innovative processes of product creation, it is advisable to allocate bounded contexts according to the life cycle stages: concept formation, product requirements formation, structural design, value-added chain design, prototype production and testing, industrial production and maintenance, disposal. At the same time, if the same group is engaged in several works throughout the life cycle, these works can be combined into one bounded context. For example, the work on the concept (product quality characteristics) formation and requirements formation, structural design and value-added chain design can be combined in the same bounded context, provided that the same specialists work together on the listed tasks. In *Figure 1*, individual works are shown as ovals, and their associations into bounded contexts are shown as rectangles. At the same time, the implementation of bounded contexts may differ significantly from each other depending on the functional subsystems or even the components of the product.

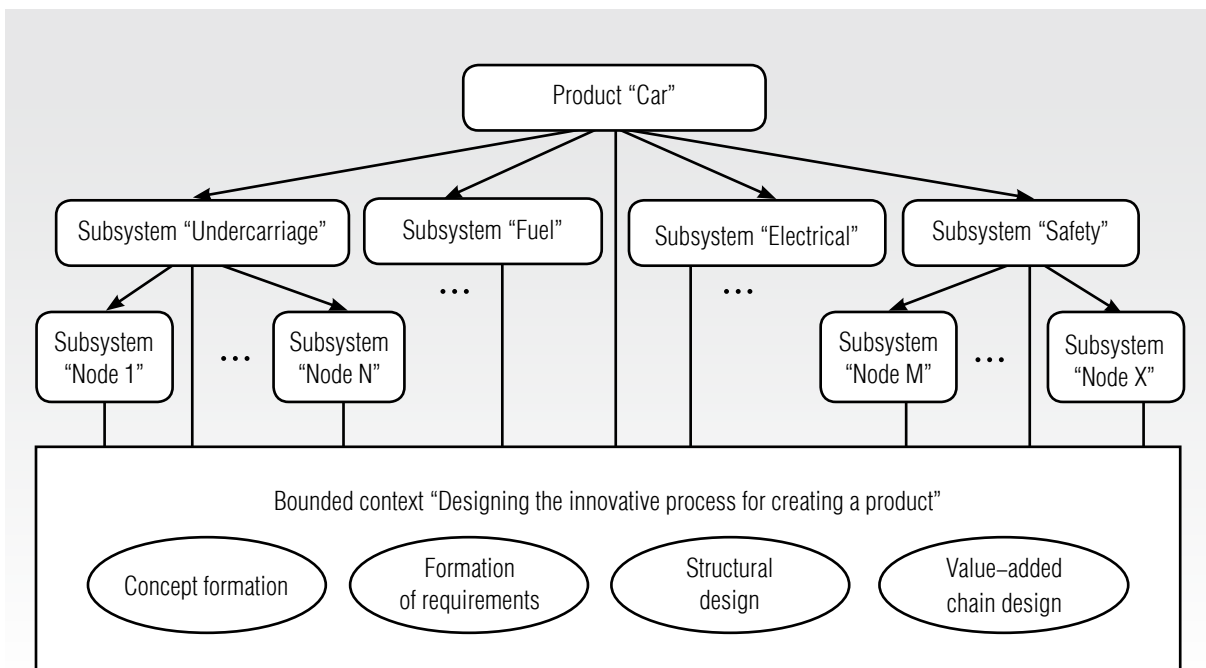


Fig. 1. Example of interaction between subsystems and bounded contexts

Within one bounded context, the participants in the product creation process work on an equal basis, forming sets of properties and rules (procedures) of behavior. At the same time, standard ontologies, reference books, and any other external information resources can be used within one bounded context. In the process of joint work, a common language of interaction is formed, formalized in the form of an iteratively developed ontology. In the course of work, mathematical and simulation models can be developed to justify the adoption of optimal (or rational) design decisions.

Bounded contexts of the innovative process for creating a product can be set both for the entire product and its components (for a detailed study). In the bounded context of product design, the value-added chain is being worked out, causing the need to select the product components contractors. In this case, interfaces between bounded contexts for the entire product and its components should be organized, as a rule, in the “requirement – execution” paradigm.

Interfaces between bounded contexts can be implemented in various ways [14]:

- ◆ **Partnership.** In this case, groups of process performers working on related subject sub-domains actively exchange the results of joint work. Group members work on common parts of the ontology and should exchange both the parts of the ontology being developed and the project results.
- ◆ **Shared kernel.** This method of interaction can be considered as a special case of partnership organization. In this case, a common bounded context is organized for several groups that have access to it with different modes of resource sharing and results adjustment.
- ◆ **Customer – supplier development.** This method can be considered as a way of interaction along the value-added chain when a higher-level product group representing the customer forms specifications of requirements for the contractor of the order (supplier). The supplier must cognitively and economically evaluate the received requirements and, following the requirements, must give consent to the execution and fulfil them accordingly. In this case, it is possible to interact with external suppliers (manufacturers) in the mode of sending messages and deferred decision-making. To coordinate requirements between customers and suppliers, it is also necessary to coordinate interacting fragments of ontologies.
- ◆ **Conformist.** This method of interaction is a specific case of the customer-supplier interface. In this case, the contractor is in a subordinate position and fulfils the customer’s requirements without agreeing with the latter on the possibilities of fulfilling the order. This way of establishing the interface is characteristic of the internal relations of the company’s divisions when a superior structure is well aware of the capabilities of subordinate structures, and subordinate structures fully own the ontology or are directly involved in its development.
- ◆ **Anticorruption layer.** This method of interaction is also a specific case of the customer-supplier interface. But in this case, direct access to the supplier’s data for criterion selection is prohibited. Getting answers to the queries of interest is possible only through sending messages, usually in offline mode. Thus, the subordinate level of the organization protects itself from possible customer access to confidential information. In this case, requirements adjustment through ontology is the most difficult and requires the translation of the shared knowledge representation often.
- ◆ **Open host service.** In this case, a protocol is used that provides access to a bounded context through a set of services that are expanded and refined as the project develops.

- ◆ **Published language.** This method is a specific case of the “service with an open protocol” method. The use of a public language involves the use of a common project ontology. In some cases, it is possible to translate concepts in both directions of interaction.
- ◆ **Separate ways.** The unrelated use of bounded contexts may be due to competing product and service development. In this case, decisions are made as the project is completed.
- ◆ **Big ball of mud.** This is the interaction of a new product with an already existing part of the product which can be performed without complying with the requirements of subject-oriented design.

From the point of view of creating some material product, the structural model of bounded context is transformed into the program code of a digital twin. At the same time, the digital twin will correspond to one bounded context, reflecting either the entire product or its specific component and represent an autonomous software agent in a multi-agent system based on knowledge to solve the problems of designing the innovative process for creating products and providing services. At the same time, the presence of several subsystems (subdomains) in the product makes it possible to specialize the development of software modules for the corresponding subdomains and the need to obtain integrated solutions at the level of the entire product.

2. Software implementation of a knowledge-based multi-agent system for designing innovative processes using the asset administration shell

The knowledge-based system proposed in [1] for designing innovative processes to create products and services can be implemented in the architecture of the 4th generation industry system (the i4.0 system) [20–22], which

includes a set of interacting i4.0 components – Asset Administration Shells (AAS), which are accessed by i4.0 applications of innovative processes participants. AAS is a software implementation of the concept of digital twins of created products or parts thereof, as well as related resources: machine tools, production lines, supply chains, actors executing production processes, information assets and organizational and management assets. From the point of view of the domain-driven design approach, AAS of products and their components within the design of innovative processes for creating products and services will constitute the semantic core of KMS, and AAS of the remaining resources will be auxiliary entities.

AAS services are categorized into application component services and infrastructure services of the AAS platform. *Figure 2* shows the structure of an asset administration shell for products (product components) interacting with the i4.0 platform.

An asset administration shell of a digital twin reflects data about a specific physical entity – any resource and includes a set of services that implement functional modules for solving design problems.

Asset data determine the state of this entity at various stages of the life cycle. It can be planned, actual, predicted, simulated, etc. Data can be categorized into sub-models, such as identification, technical, operational and documentation sub-models:

- ◆ identification data, which must be immutable throughout the entire life cycle of the project;
- ◆ technical data – quantitative and qualitative characteristics that meet the business requirements for the product to perform the functions of the product;
- ◆ functional data – requirements for the functional subsystems of the product in case of a complex structure;

- ◆ project documented data that reflects the characteristics of the implementation of the functional requirements in terms of the technologies used and process executors;
- ◆ operational data that reflects the performance of products at the operational stage or the results of simulation and mathematical modelling at the design and development stages.

AAS application component services correspond to functional modules of applications. In the case of a bounded context associated with the design of innovative processes to create products and services, services of application components call application modules for the formation of quality characteristics, product requirements, design of structures and

value-added chains, as well as a planning module that coordinates the execution of all functional modules [1].

Infrastructure services of the i4.0-platform are auxiliary. They ensure the functioning of the AAS itself and the interaction between AAS, and from this point of view, they are a platform and standardized. Infrastructure services refer to the i4.0 platform. The i4.0 system is developing on the i4.0 platform basis.

First of all, infrastructure services include AAS management services, which perform the functions of creating AAS components, registering and creating service registries, searching and presenting services, searching and presenting AAS data and submodels.

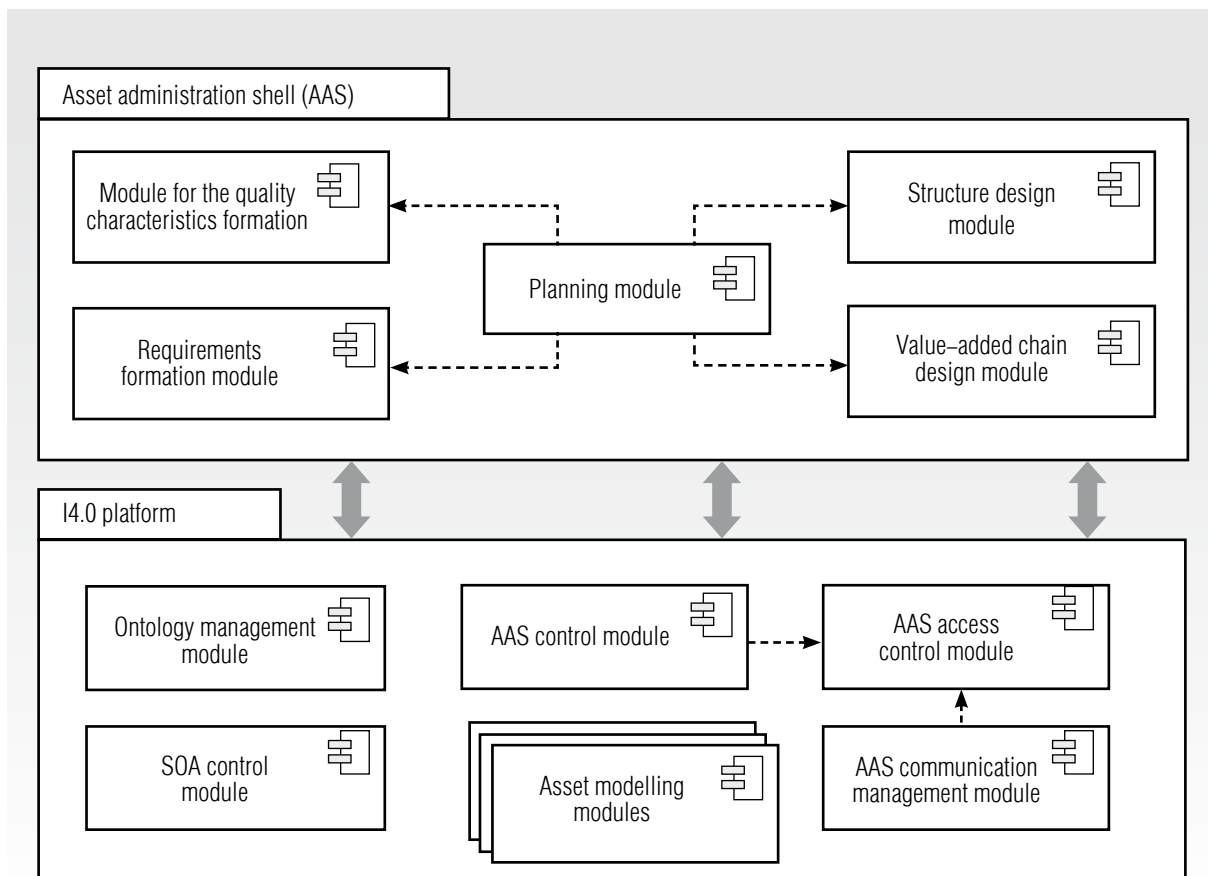


Fig. 2. Structure of the asset administration shell for products (product components) interacting with the i4.0-platform

A set of standardized asset modelling services enables the generation and use of asset data using modelling tools. Standard modelling tools are used that implement:

- ◆ Quality Function Deployment (QFD) methods to prioritize business and functional requirements implementation;
- ◆ Failure Mode and Effects Analysis (FMEA) methods for analyzing the types and consequences of potential non-conformities;
- ◆ methods of processing fuzzy sets;
- ◆ simulation methods;
- ◆ statistical modelling methods;
- ◆ machine learning and neural network modelling, etc.

Thus, on the i4.0 platform, you can implement the general functionality of various types of modelling (asset modelling modules) required to build a value-added chain - simulation mechanisms, QFD and FMEA algorithms, and other mathematical and statistical modelling algorithms. At the same time, any AAS participating in various stages of the value-added chain can call these services to implement the corresponding functionality (*Figure 2*), substituting the necessary input data for modelling from its submodels and obtaining required results.

Infrastructure services of the i4.0 platform are used to control access of application components services to data and models in specific AAS. Infrastructure services set access restrictions for various application components, access ontologies, determine the context for sharing application components, check (classify) the compliance of calls to application components services. In *Figure 2*, the infrastructure platform services implement the functions of ontology management, service-oriented architecture, interoperability with other AAS and access control to AAS.

AAS can provide service interfaces for software applications to access their data and invoke commands or run models. Service interfaces (for example, implemented in RESTful API) provide communication and interoperability between AAS and applications.

Typically, modern industrial products have a complex structure. In this case, for the entire product and its specific parts, separate i4.0 components (AAS) are created, which are connected by service interfaces (*Figure 3*). Both external subcontractors and internal executors can act as launch actors for functional applications and corresponding i4.0 components.

These i4.0 components exchange information with each other using three modes of interaction:

- ◆ passive mode – file exchange;
- ◆ reactive mode – by organizing data exchange and calling services through API;
- ◆ proactive mode (peer-to-peer), when digital twins automatically jointly carry out some production processes.

Since the article deals only with the initial stages of creating products and services associated with concept formation and value-added chain design, then either passive or reactive interaction modes are considered as modes of interaction, involving the active participation of people in the design process. From the point of view of the multi-agent implementation of the interaction process of i4.0 components belonging to external participants in the value-added chain, the reactive interaction mode corresponds to various modes of sending messages and subscribing to information, as well as service interaction (see the section “The domain-driven design approach for the structuring of designing innovative processes domain”).

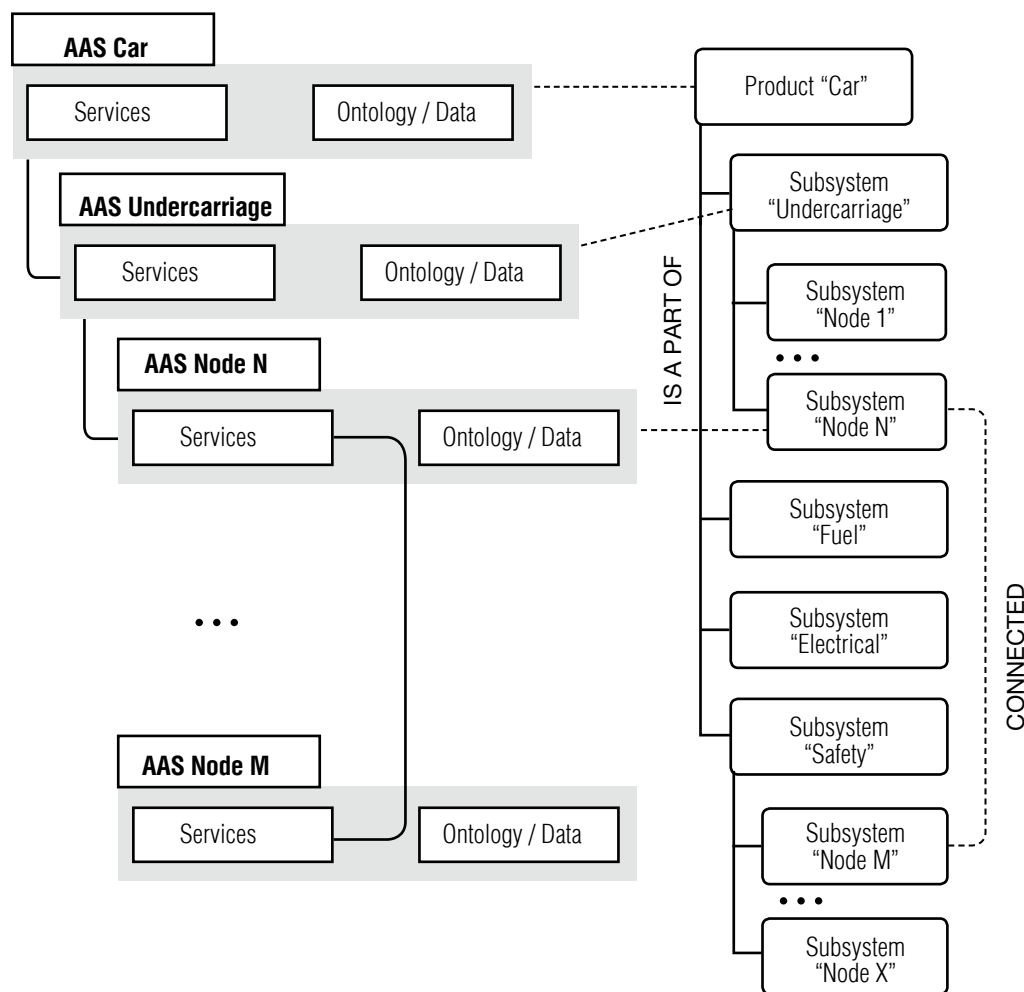


Fig. 3. An example of a layered AAS system (based on [21])

3. Technology for designing innovative processes to create products using ontological and cognitive approaches in the i4.0 system

The scheme of interaction between participants of a network enterprise in the i4.0 system which implements the functions of a knowledge-based system to support innovative processes of products creation is shown in *Figure 4*. Based on the reference model of the domain ontology, project ontologies are formed both for the entire project and its components. As noted in the first section, ontologies are formed iteratively and are used to

solve search problems in knowledge bases and exchange knowledge between separate AAS. Standardized services of QFD [23], FMEA [24], fuzzy set processing [25] are used in functional modules of AAS, realizing cognitive decision-making mechanisms. Let's consider the technology for designing innovative processes to create products in the i4.0 system in detail.

The core of AAS is a **planning module** that implements interface services for interacting with participants of an innovative process of product creation and provides coordination of functional services. In particular, the module supports the decision-making process

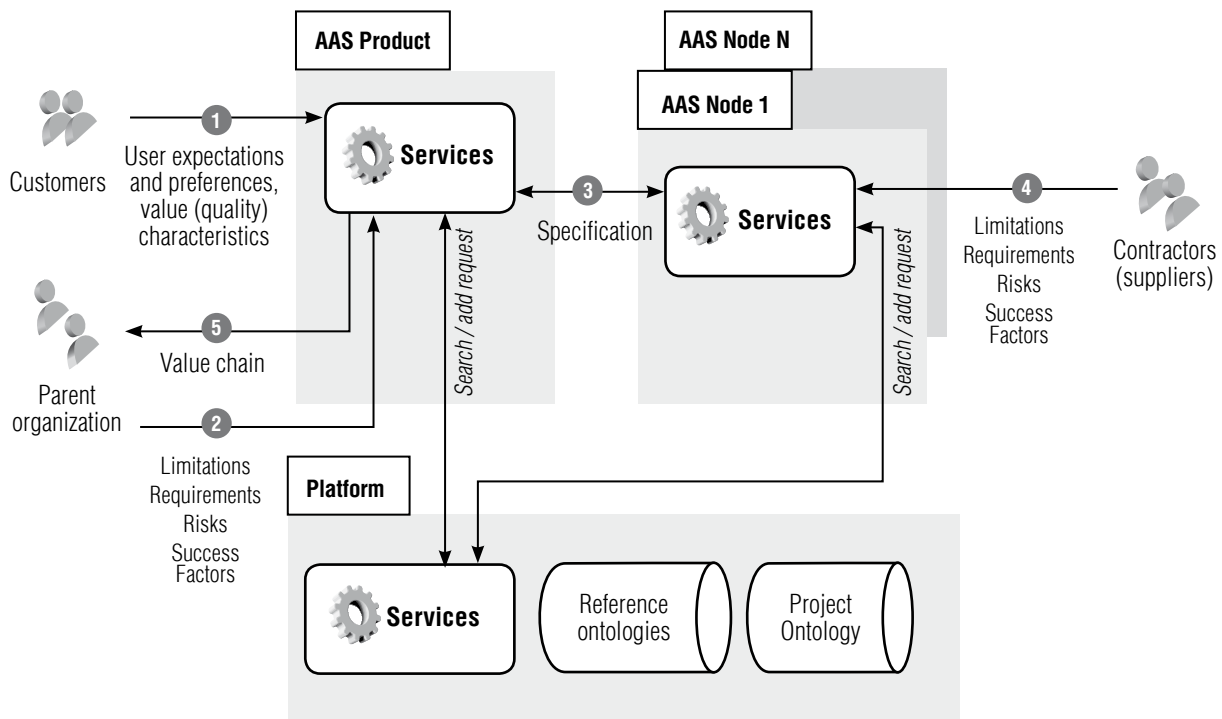


Fig. 4. Scheme of the interaction between participants of a network enterprise in the i4.0 system

for the choice of product design and value-added chain alternatives based on a combination of QFD and fuzzy logic methods. Within the QFD method, when describing each value (quality) characteristic (business requirement), a priority (weight) is established in an expert way, where the most important characteristic with the greatest consumer interest has the maximum weight. The further procedure involves the collection of requirements and an assessment of their impact on value (quality) characteristics, and the selection of the most important requirements. The subsequent assessment of structural elements in terms of their impact on the implementation of requirements is associated with the choice of a product design alternative that better meets the selected requirements. The assessment of risks associated with suppliers and(or) contractors takes into account the generated design alternatives. As a result, the choice of the optimal

value-added chain alternative is carried out, in which requirements and associated risks are determined for all participants, and access to the data of AAS models (distributed repository) is organized. A detailed description of the methodology for using the QFD method is presented in the article [26].

The formation of product design alternatives is initiated in the AAS and goes through a series of iterations [1, 26], within which calls to the services implemented by the AAS modules are made.

The product value for the consumer is determined through the value (quality) characteristics (or business requirements), which are described in the project ontology using the services of the **quality characteristics formation module**, as well as a set of functional and non-functional requirements, the criteria base for assessing the implementation

of requirements, which are described using the services of **the requirements formation module**.

An integrated expression of the cognitive understanding of the market situation for the specific product type finds its representation in the SWOT analysis methodology, which reveals the advantages and disadvantages of the specific product type manufacturing in generalized form, as well as the opportunities and threats of the market implementation of the project [27].

The formation of product quality characteristics is a poorly formalized task of analyzing the external and internal factors of the specific market for products and services. Specialists of various profiles take part in this process: marketers, researchers of new technologies and materials, designers and technologists in the area of similar product types. Solving the problem of selecting the most significant properties of a designed product can use the methods of Big Data analysis of markets of products, technologies and materials, predictive forecasting of development trends, analysis of resource constraints and risks of mastering new types of products and technologies.

Product classification, standard (mandatory and optional) value characteristics, product parameters and some logical rules presented in the reference ontology are used to describe an innovative product (a new class of products) in the project ontology. The product inherits some value characteristics of the category to which it belongs and/or has some unique characteristics not typical for this product category.

With the help of existing services, expected and attractive characteristics are evaluated additionally and classified in the project ontology (according to Kano [28]). Furthermore, the presence of indifferent, contradictory and repellent characteristics is minimized. In this case, the highest priority (weight) according to

QFD will have the mandatory (expected) and attractive characteristics, and the contradictory or repellent characteristics – the least. In the course of the project, some of the optional characteristics, the implementation of which leads to significant project costs, as well as repellent and contradictory characteristics that reduce the value of the product for the consumer, can be excluded from the set.

Information about the value (quality) product characteristics can be refined during the project and is used in the **requirements formation module**, where for each of the identified value (quality) characteristics, at least one requirement is determined, the implementation of which provides the corresponding characteristic. Functional requirements should be distributed across subsystems (subdomains of the subject area).

Typical requirements inherent in specific product categories and specified by standards (for example, technical specifications and GOSTs) are transferred to the project ontology from the reference ontology, and the original requirements and project constraints derived from these requirements (or independent of them) are introduced.

To identify mandatory requirements, the degree of connection of each requirement with the implementation of a quality characteristic is assessed. As a result of using the fuzzification function, the confidence coefficient for assessing such a relationship is reduced to the scale [0, 1]. The total importance for the implementation of the entire set of quality characteristics is calculated for each requirement using the formula [29]:

$$\hat{T}_j = \sum_f \varphi_f(C_{jf} P_f), \quad (1)$$

where \hat{T}_j – fuzzy assessment of the importance of the j -th requirement for the implementation of quality characteristics (business requirements) of an innovative product;

C_{jf} – level of connectivity of the j -th requirement and the f -th characteristic on the scale [0, 1, 3, 9];

P_f – preset priority of the f -th characteristic on the scale [1, N];

φ_j – fuzzification function on the scale [0, 1]: the more the importance of the j -th requirement for the implementation of the f -th characteristic, the closer to 1;

Σ – operation of additive summation of fuzzy numbers.

To determine the final set of groups of mandatory and optional requirements which must ensure the presence of all mandatory and expected characteristics, the specified threshold values of the assessment \hat{T}_j are used. Expected and optional requirements that are below the threshold are not considered further.

The requirements are detailed down to the level of specific functions and structural elements. They are described in the project ontology within **the structure design module**.

A condition for including a structural element in a product design alternative (bill of materials, BOM) is to ensure the ability to perform some product functions or to implement mandatory interface requirements. The product breakdown structure (PBS) can also be inherited from the reference product ontology. Thus, each design element and function of a product is directly or indirectly related to one or more requirements which define the product design alternative.

Ranking (comparing) different product design alternatives is possible by using QFD services to assess the impact of structural elements on meeting the requirements. In this case, the estimates of the importance of requirements obtained according to the above formula are considered when calculating the specific structural element estimates.

BOM specification is a poorly formalized, iterative and innovative task. In the course of this task, the BOM is defined in such a way as to ensure, on the one hand, full compliance with existing requirements (a design alternative must always include at least one element for each mandatory requirement) and, on the other hand, an appropriate level of costs. A threshold value can be set and used to select optional elements and/or elements with high costs.

The determining factor is the possibility of purchasing the selected types of structural elements and/or their production. To assess a possibility, the generated design variant is passed to the planning module, which calls the services implemented by the value-added chain design module. The market analysis, the choosing of the innovative process alternatives and the selection of its participants are carried out with the help of these services.

The absence from suppliers of a product design structural elements related to mandatory requirements can lead to one of the following organizational scenarios triggered by the planning module:

- ◆ searching for contractors who could provide the manufacturing of the missing structural element using the specified technologies, tools and design constraints;
- ◆ selection according to the rank of another product design alternative, which does not contain the missing elements;
- ◆ revision of the product concept to clarify the characteristics and requirements to eliminate the missing structural element;
- ◆ termination of work due to the lack of the possibility of revising the concept of the product.

To identify businesses that are capable of custom-made missing structural elements, it is necessary to form a value-added chain alternative that includes manufacturing, design, assembly, processing, setting up, maintenance

and delivery of products. The set and connections of activities are contained in the project ontology, where possible tools and technologies can also be determined, as well as compliance with the specified design constraints and established requirements.

Assessment of the total impact of all activities in the value-added chain on ensuring quality characteristics and meeting requirements allows one to select participants in the process, taking into account their competencies (ability to perform certain types of work, possession of tools and technologies). The formula for calculating the deviations of requirements from the capabilities of the enterprise, based on fuzzy logic, is considered below while determining the possibility of performing some types of work on their own or outsourcing them [29]:

$$\hat{V}_{iu} = \sum_j \theta_j (T_{ij} - R_{ij}), \quad (2)$$

where \hat{V}_{iu} – fuzzy assessment of the competitiveness of the u -th product component on the market supplied (produced) by the i -th contractor;

T_{ij} – a given level of the j -th requirement for the implementation of the u -th component by the i -th contractor;

R_{ij} – level of implementation of the j -th requirement for the u -th component by the i -th contractor;

θ_j – fuzzification function (0, 1) on the scale, the smaller the deviation, the higher the competitiveness;

\sum – operation of additive summation of fuzzy numbers.

The level of implementation of the j -th requirement for the u -th component by the i -th contractor is determined in the process of the requirement adjustment with the contractor by interpreting the data received from the contractor (on a natural or fuzzy scale [0, 1]).

Accordingly, the choice of the i -th contractor (supplier or manufacturer) is carried out according to the maximum assessment of the competitiveness of the u -th supplied (produced) component in the market:

$$\hat{V}_u^* = \max_i \hat{V}_{iu} \hat{K}_{iu}, \quad (3)$$

where \hat{V}_u^* – maximum assessment of the competitiveness of the u -th supplied (produced) component;

\hat{V}_{iu} – fuzzy assessment of the competitiveness of the u -th product component on the market supplied (produced) by the i -th contractor;

\hat{K}_{iu} – risk ratio related with delivery of the u -th component by the i -th contractor.

The risk ratio is calculated as a multiplicative product of fuzzy numbers corresponding to specific risk types, which are determined on a fuzzy scale [0, 1]:

$$\hat{K}_{iu} = \prod_k K_{iuk}, \quad (4)$$

where \hat{K}_{iu} – risk ratio related with delivery of the u -th component by the i -th contractor;

K_{iuk} – risk ratio for the k -th type of risk related with delivery of the u -th component by the i -th contractor;

\prod – fuzzy multiplication operation.

During the search for contractors, **platform services** for ensuring the interaction of AAS are used. By means of them, requests are sent to the contractors. In particular, the specification of the structural elements set in which the initiating organization is interested, the description of the corresponding tools and technologies, the structural element requirements and the design constraints are sent. Through an existing or created AAS, the contractor uses calculation models and the information provided in the enterprise databases to assess the production possibilities. For this, similar services are called, which are imple-

mented at the structural element level by the module for the formation of quality characteristics, the requirements formation module, the structure design module.

If the enterprise does not fully meet the specified conditions, then the appropriate response is sent through the platform services. Otherwise, the results of the calculation models and the structural element characteristics that can be developed are sent. The absence of positive responses to the mailing from the contractors is the basis for switching to other process alternatives according to the previously determined rank.

The selected process alternative with a list of activities and participants, a detailed description of the product and structural elements are saved by platform services in the submodel of the project documentary data of AAS. They are used at the production stage to coordinate work (direct communication between the process participants services), as well as during the maintenance of the product and during the development of new types of products.

Conclusion

As a result of the research into the technology for designing innovative processes to create products of a network enterprise using the i4.0 knowledge-based system, the following conclusions can be drawn:

It is advisable to decompose the subject area of designing innovative processes for creating products and services based on the Domain-driven design approach, which provides, on the one hand, an objective reflection of the structure of products and services and value-added chains, and on the other hand, the formation of project teams capable of effectively interacting within the assigned tasks.

The knowledge-based system for designing innovative processes to create products and

services can be based on Plattform Industrie 4.0 (i4.0 platform), which provides a multi-agent implementation of the design process in a distributed network architecture using digital twins and digital threads technologies.

Domain-driven design allows us to highlight the bounded contexts of creating products and services by stages of the life cycle, and for the design phase - the sequence of activities of concept formation, functional and non-functional requirements formation, product design and value-added chain design. Accordingly, the structure of an i4.0 system based on domain-driven design decomposition will include sets of interacting Asset Administration Shells (AAS) that implement bounded contexts for design.

The feature of the proposed technology for designing innovative processes to create products and services is its iterative nature. During the process, AAS are dynamically created for digital twins of products and their components, forming and evaluating alternatives of value-added chains for the best fulfilment of functional and non-functional requirements.

The use of standardized ontological services of the i4.0 platform provides dynamic creation of ontologies from reference models and their actualization as an innovative project of creating products and services develops.

Standardized methods of the i4.0 platform in terms of information exchange between AAS allow for active interaction between participants in the innovative process, aimed at well-grounded and coordinated design alternatives.

The functional modules of AAS are based on standardized services that implement cognitive methods: QFD to determine priorities of business requirements and functional requirement implementation; FMEA to risk assessment; a method for processing fuzzy sets to obtain multi-criteria assessments of the advis-

ability of making specific design decisions. The application of these methods allows for multi-criteria optimization of the innovative design process.

Characterizing the proposed technology for designing innovative processes to create products and services, we note its novelty in terms of the proposed methods for implementing multi-agent interaction between participants in the innovative process within the dynamic construction of an i4.0 system and the use of ontological and cognitive methods for devel-

oping design solutions. In the future, there are plans to continue research into the application of the developed methods and mechanisms of i4.0 systems for the implementation of subsequent stages of the life cycle of creating products and services. ■

Acknowledgments

This study was funded by the Russian Foundation for Basic Research according to the research projects No. 19-07-01137a.

References

1. Telnov Yu.F., Kazakov V.A., Trembach V.M. (2020) Developing a knowledge-based system for the design of innovative product creation processes for network enterprises. *Business Informatics*, vol. 14, no 3, pp. 35–53. DOI: 10.17323/2587-814X.2020.3.35.53.
2. Bajaj M., Hedberg T. (2018) System lifecycle handler – Spinning a digital thread for manufacturing. Proceedings of the *28th Annual INCOSE International Symposium. Washington, DC, 7–12 July 2018*, vol. 28, pp. 1636–1650. DOI: 10.1002/j.2334-5837.2018.00573.x.
3. NIST (2020) *Digital thread for smart manufacturing*. Available at: <https://www.nist.gov/programs-projects/digital-thread-smart-manufacturing> (accessed 26 July 2021).
4. Grieves M. (2016) *Origins of the digital twin concept*. Available at: https://www.researchgate.net/publication/307509727_Origins_of_the_Digital_Twin_Concept (accessed 26 July 2021). DOI: 10.13140/RG.2.2.26367.61609.
5. Minaev V.A., Mazin A.V., Zdiruk K.B., Kulikov L.S. (2019) Digital twins of objects in the solution of control problems. *Radioindustry*, vol. 29, no 3, pp. 68–78 (in Russian). DOI: 10.21778/2413-9599-2019-29-3-68-78.
6. NTI (2019) *Digital twins in the high-tech industry. Expert and analytical report*. Available at: http://assets.fea.ru/uploads/fea/news/2019/12_december/28/cifrovoy_dvoinik.pdf (accessed 26 July 2021).
7. Wooldridge M. (2009) *An introduction to multi-agent systems*. New York: Wiley, 2009.
8. Gorodetskiy V.I., Grushinskiy M.S., Khabalov A.V. (1998) Multi-agent systems (review). *Artificial Intelligence News*, no 2, pp. 64–116 (in Russian).
9. Telnov Yu.F., Danilov A.V., Diveev R.I., Kazakov V.A., Yaroshenko E.V. (2018) Development of a prototype of multi-agent system of network interaction of educational institutions. *Open Education*, vol. 22, no 6, pp. 14–26 (in Russian). DOI: 10.21686/1818-4243-2018-6-14-26.
10. Uschold M., King M., Moralee S., Zorgios Y. (1998) The enterprise ontology. *Knowledge Engineer Review*, vol. 13, no 1, pp. 31–89. DOI: 10.1017/S0269888998001088.
11. Gavrilova T.A., Kudryavtsev D.V., Muromtsev D.I. (2016) *Knowledge engineering. Models and methods*. Saint Petersburg: Lan' (in Russian).
12. Efimenko I.V., Khoroshevsky V.F. (2011) *Ontological modeling of enterprises and markets of modern Russia. Part 1. Ontological modeling: Approaches, models, methods, tools, solutions. Working paper WP7/2011/08 (part 1)*. Moscow: HSE Publishing House (in Russian).

13. Evans E. (2003) *Domain-driven design: Tackling complexity in the heart of software*. Boston: Addison-Wesley.
14. Vernon V. (2016) *Domain driven design distilled*. Boston: Addison-Wesley.
15. Industrial Internet Consortium, Plattform Industrie 4.0 (2017) *Architecture alignment and interoperability. Joint Whitepaper*. Available at: https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf (accessed 26 July 2021).
16. Plattform Industrie 4.0. (2019) *Details of the asset administration shell from idea to implementation*. Available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/vws-in-detail-presentation.pdf?__blob=publicationFile&v=12 (accessed 26 July 2021).
17. Industrial Internet Consortium, Plattform Industrie 4.0 (2020) *Digital twin and asset administration shell concepts and application in the industrial internet and Industrie 4.0. Joint Whitepaper*. Available at: <https://www.iiconsortium.org/pdf/Digital-Twin-and-Asset-Administration-Shell-Concepts-and-Application-Joint-Whitepaper.pdf> (accessed 26 July 2021).
18. Kuznetsov O.P. (2012) Cognitive semantics and artificial intelligence. *Artificial Intelligence and Decision Making*, no 4, pp. 95–105 (in Russian).
19. Trembach V.M. (2017) Cognitive approach to developing the intelligent modules of organizational and technical systems. *Open Education*, vol. 21, no 2, pp. 78–87 (in Russian). DOI: 10.21686/1818-4243-2017-2-78-87.
20. The Plattform Industrie 4.0 (2020) *Details of the asset administration shell. Part 1 – The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01)*. Available at: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf?__blob=publicationFile&v=5 (accessed 26 July 2021).
21. Plattform Industrie 4.0 (2019) *Usage view of asset administration shell. Discussion paper*. Available at: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/2019-usage-view-asset-administration-shell.pdf?__blob=publicationFile&v=4 (accessed 26 July 2021).
22. Plattform Industrie 4.0 (2019) *Functional view of the asset administration shell in an Industrie 4.0 system environment. Discussion paper*. Available at: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Functional-View.pdf?__blob=publicationFile&v=5 (accessed 26 July 2021).
23. Vashukov Yu.A., Dmitriyev A.Ya., Mitroshkina T.A. (2012) *QFD: Product and process development based on customer requirements and expectations: A methodological guide*. Samara: SSAU (in Russian).
24. Vashukov Yu.A., Dmitriyev A.Ya., Mitroshkina T.A. (2008) *Type, consequence and cause analysis of potential nonconformities (FMEA): A methodological guide*. Samara: SSAU (in Russian).
25. Averkin A.N. (2013) *Fuzzy sets in management and artificial intelligence models*. Moscow: Kniga po Trebovaniyu (in Russian).
26. Telnov Yu.F., Trembach V.M., Danilov A.V., Yaroshenko E.V., Kazakov V.A., Kozlova O.A. (2019) Constructing network enterprise structure to create innovative products. *Open Education*, vol. 23, no 6, pp. 59–73 (in Russian). DOI: 10.21686/1818-4243-2019-6-59-73.
27. Thompson A.A. Jr., Strickland III A.J. (2003) *Strategic management: Concepts and cases*. Boston, Mass.: McGraw-Hill/Irvin.
28. Kano N., Seraku N., Takahashi F., Tsuji S. (1984) Attractive quality and must-be quality. *Journal of the Japanese Society for Quality Control*, vol. 14, no 2, pp. 147–156. DOI: 10.20684/quality.14.2_147.
29. Kalachikhin P.A., Telnov Yu.F. (2018) Value-added chains formation in network structures of interaction based on intelligent technologies. Proceedings of the XVI National Scientific Conference on Artificial Intelligence with International Participation (KII–2018), Moscow, 24–27 September 2018, vol. 1, pp. 106–115 (in Russian).

About the authors

Yury F. Telnov

Dr. Sci. (Econ.), Professor;

Head of the Department of Applied Information Technologies 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

Vasily A. Kazakov

Cand. Sci. (Econ.);

Leading Researcher, Research Institute “Strategic Information Technologies”, 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