Ontological Engineering in Complex Systems Based on Meta-Associative Graphs

V. V. Borisov[a,]* **and A. E. Misnik**[b,]**

[a] *Smolensk Branch, National Research University "Moscow Power Engineering Institute," Smolensk, 214013 Russian Federation*

[b] *Belarusian-Russian University, Mogilev, 212000 Republic of Belarus*

**e-mail: vbor67@mail.ru*

***e-mail: anton@misnik.by*

Abstract—The paper proposes a variety of metagraphs focused on the use in ontological engineering of complex systems characterized by complexity of structure, multicomponent nature, the presence of functional subsystems that solve various target tasks, and a large number of parameters characterizing the processes of the system. Ontological engineering aims to ensure the adoption of high-quality management decisions by increasing the level of integration of the necessary information, improving the search capabilities of databases and knowledge, and providing the possibility of joint work with the knowledge base, which is provided by a single semantic description of applied ontology. Meta-associative graphs allow providing a flexible representation of the ontology of the domain of a complex system suitable for modeling system processes and ensuring the functioning of information-analytical processes in the system, minimizing, in situations with no need for detail, the necessary number of steps to obtain the necessary information, but, if there is a need for detailed data, providing the ability to get all the related data. The inclusion of the process component in the ontological model of the system makes it possible to increase the overall flexibility of process modeling.

Keywords: ontological engineering, metagraphs, meta-associative graphs

DOI: 10.1134/S1054661823030045

INTRODUCTION

The efficiency of management of complex systems largely depends on properly designed and well-established system-defined processes and, accompanying them, information-analytical processes [3, 4].

The ontological approach has a number of key advantages in the design and maintenance of processes in complex systems, since the feature of such systems is a high level of process integration and the sharing of data and knowledge [14, 21].

The features of complex systems are as follows:

- complexity of structure, multicomponent nature, the presence of functional subsystems that solve various target tasks, and a large number of parameters characterizing the processes of the system;
- dynamic change in structure and parameters of the system;
- incomplete information on the functioning and state of the system;
- variety of external and anthropogenic impacts on the system;
- the presence of complex nonlinear relationships between parameters;
- difficulty in making management decisions to improve the efficiency of the system;
- limited opportunities for experimental research of the system and ongoing processes;
- impossibility of using a unified approach to the creation of models of industrial-engineering and information-analytical processes occurring in such systems [9, 13].

The conducted research made it possible to justify the use of meta-associative graphs for ontological engineering of complex systems. The requirements for the toolkit integrating ontological and metagraph representations within a single software-instrumental environment are determined, which is not achieved by the traditional unification of ontologies.

1. FEATURES OF ONTOLOGIES OF COMPLEX SYSTEMS

Ontology describes concepts in a subject area, as well as the relationships that exist between these concepts. Thus, the basic characteristics of ontology are the hierarchy of concepts/objects, which is established using various semantic links, the names of which are chosen on the basis of the accepted terminology of the subject area, in order to overcome the semantic gap between the expert and the knowledge engineer [6, 17, 20].

Received February 4, 2023; revised February 4, 2023; accepted February 4, 2023

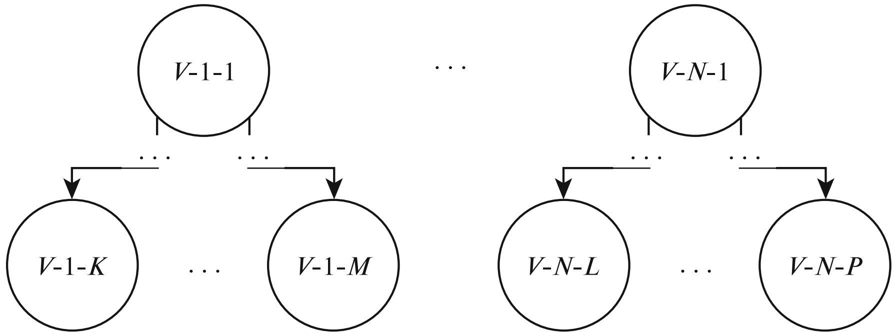

Fig. 1. Attempt to represent the ontology of a complex system in the form of classical graphs.

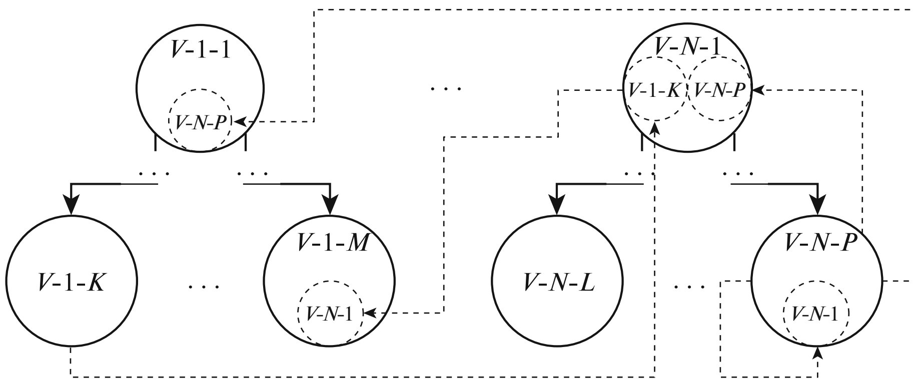

Fig. 2. Illustration of the dependences between the entities of the ontology of a complex system.

Ontological engineering involves:

• definition of classes of concepts in ontology;

• guide to taxonomy of classes;

• development of structures of concepts and situations;

• definition of the properties of concepts and meanings of these properties;

• procedures for the withdrawal and transformation of situations [11].

For practical use in the final stages of development, applied ontology should include so-called instances (entities) of created classes with specific values of their properties describing the real mapping of the applied ontology [1].

The complexity of the structure, the multicomponent nature, and the presence of functional subsystems within a complex system naturally affect the presentation of its ontology. In general, the ontology of a complex system can be considered as a specification of the processes of its construction and operation, regardless of the methods of its implementation and also regardless of the technologies used for its implementation [5, 7].

2. GRAPHS AS A MEANS OF REPRESENTING THE ONTOLOGY OF A COMPLEX SYSTEM

When trying to represent the ontology of a complex system in the form of classical graphs, it may seem that it consists of completely independent segments, each of which is a classical graph (Fig. 1).

However, the presence of complex nonlinear relationships between parameters and entities of a complex system does not allow the use of classical graph structures [10]. The ontology of a complex system can be quite complicated and the essence of one, at first glance, independent graph can be included in the essence of the same or another graph (Fig. 2).

Obviously, the toolkit of classical graphs does not make it possible to formalize such relationships between the entities of ontology of a complex system [12, 18].

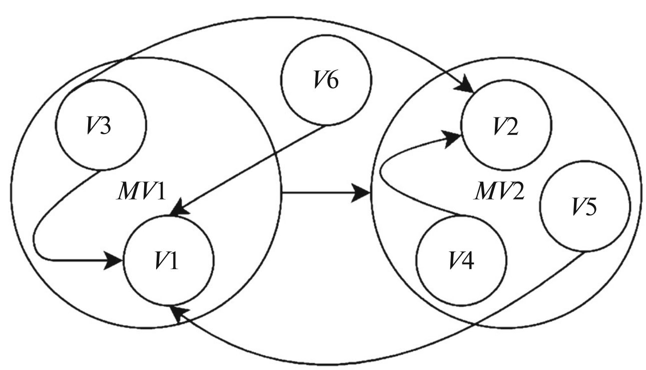

Fig. 3. Example of metagraph.

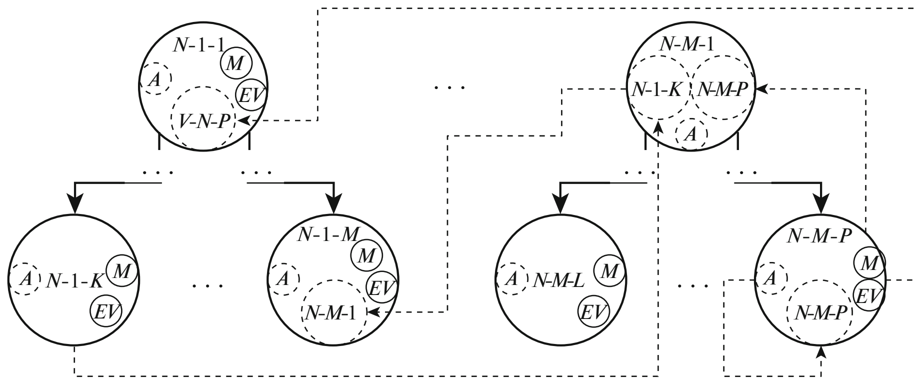

Fig. 4. Meta-associative graph.

The need to improve the adequacy of ontology leads to the search for tools to use not only binary relations between the vertices of the graph but also relations that unite more than two vertices. A hypergraph can be used as a model. The edge of the hypergraph connecting a set of vertices corresponds to a specific relationship between a given set of vertices. This, for example, may be a set of system components involved in performing a particular function, or a set of data describing a particular object, properties, and events, or a set of elements combined into one whole [8].

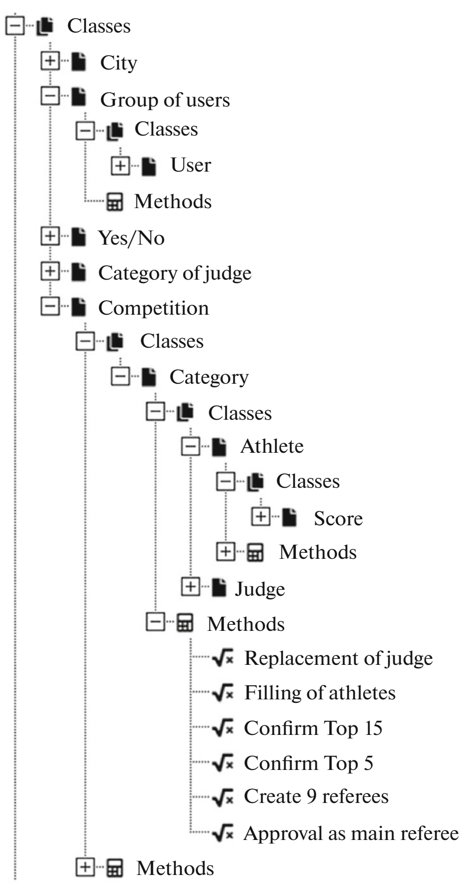

Fig. 5. Ontology of the subject area Competitions.

Currently, hypergraphs are widely used in those areas where the analysis of complex relationships between vertices of a graph is required, in particular, in the design of hardware. However, by virtue of the definition, the hypergraph does not have enough convenient capabilities for modeling hierarchies. Unlike the hypergraph, the metagraph is a model of a hierarchical object with a different level of generalization [19].

A metagraph is a collection of directional set-to-set mappings. Its promising applications are modeling data relationships, each of which is represented as mapping a set of key elements to a data set, and modeling workflow tasks, each of which can be seen as mapping a set of elements of input documents to the output set (Fig. 3).

A metagraph is defined by three sets:

$$S = \{V, MV, E\},$$

where V is the set of metagraph vertices, MV is the set of metavertices, and E is the set of edges. The metavertex of the metagraph is a vertex that includes a set of vertices v, $v \in V$.

The vertex of the metagraph can be attributive; that is, an arbitrary number of attributes can be associated with it:

$$v_i = \{A\}, \quad vi \in V,$$

where v_i is the apex of the metagraph and A is the set of attributes.

The edge of the metagraph is characterized by the origin and destination vertices and direction and connects two vertices, unlike the hyperedge of the hypergraph, which can cover several vertices. The edge can connect both vertices and metavertices of the metagraph [2].

3. META-ASSOCIATIVE GRAPHS

When describing the ontology of a complex system, a metagraph with attribute vertices, in some cases, is still not an ideal tool. The requirements for the dynamism of changes in the structure and parameters of the system, from the point of view of ontological modeling of processes in a complex system, make the differences between attributes and vertices within the metavertex rather fragile. So the vertex of the meta-

+ Add | Edit | × Delete | ↑↓

Name	uid	Data type	Array	Unique	Require	Method	LinkType
E-mail	00000000-0000-0000-0000-000000000004	text		Yes	Yes		
Surname	87f189fe-0253-4f43-af82-f1f6728d8606	text		Yes			
Given name	01fe567a-fdb0-43f8-b20a-17245559f7ad	text		Yes			
Patronym	b08c2e5e-9305-49d2-a421-49bb9c49cdbe	text		Yes			
City	6eede839-c135-4213-a298-0899dc3b7ac8	object					Yes
Referee’s category	75055304-9fc7-4c05-a9a1-2b7b90abc529	object					Yes

Fig. 6. The User class and its associations.

+ Add | Edit | × Delete | ↑↓

Name	uid	Data type	Array	Unique	Require	Method	LinkType
Referee	c7b20ea0-9f14-40d5-8f11-e4d1a9e67363	object		Yes			Yes
Top 5	b6b76ff5-05fa-417b-bff0-a7e63be930b8	object					Yes
Top 15	8ea9b6da-16c7-452a-a8a0-f93d159f6788	object					Yes
Confirm Top 15	c2050348-7e77-47d6-9fe7-39bf5d6eea78	object					Yes
Number of referee	d61d6f36-0a4b-4200-ac49-6d25f6462fe7	object					Yes
Confirm Top 5	cefadb82-a9bc-4c73-9d91-049e030d1e21	object					Yes

Fig. 7. Class Evaluation.

graph, in fact, is a degenerate case of considering a metavertex without the vertices included in it. In addition, the classical metagraph lacks natural mechanisms for identifying events and reacting to them [16].

It is proposed to generalize the concepts of the vertex and metatvertex to the concept of the node of the metagraph (N) and to supplement the definition of the node with a name and sets of events and methods belonging to the node:

$$N = \{I, AS, EV, M\},$$

where I is the name of the node, EV is the set of events associated with the node of the metagraph, and M is the set of methods associated with the node of the metagraph. AS is transformed from a set of attributes A into a set of associative attributes that are either ordinary attributes or references to nodes of the metagraph:

$$AS = \{A, N^*\}.$$

This type of metagraph makes it possible to increase the flexibility of modeling ontology, especially ontologies of processes of a complex system, since references to nodes of the metagraph, in fact, differ little from the edges of the metagraph and allow for associative connections between nodes. Any vertex potentially becomes a metavertex, because a reference to another node of the metagraph provides all the possibilities for interacting with it. On the other hand, in the absence of the need to establish an associative link, the link can be presented as a normal attribute with a value equal to the name of the node (Fig. 4).

In addition, this approach to the construction of the metagraph allows one to include the process component in the ontological model of the system as an integral part of it, which allows one to flexibly and economically form process models based on segments of the general ontological model.

4. SOFTWARE-INSTRUMENTAL TOOLS FOR ONTOLOGICAL ENGINEERING ON THE BASIS OF META-ASSOCIATIVE GRAPHS

A specialized software-instrumental environment is developed within which it is possible to design the domain ontology of the system and its system processes and implement information-analytical processes [15].

Let us consider the relatively simple task of organizing refereeing in sports competitions. For simplicity, we will limit ourselves to the consideration of processes focused only on the judges and chief judges of the competition.

The hierarchical structure of ontology is shown in Fig. 5.

Classes (nodes of the meta-associative graph) City, Yes/No, and Category of judge are independent graphs within the ontology of the subject area and, in fact, perform the functions of reference books.

The City and Judge category classes are linked through associative links to the User class (Fig. 6).

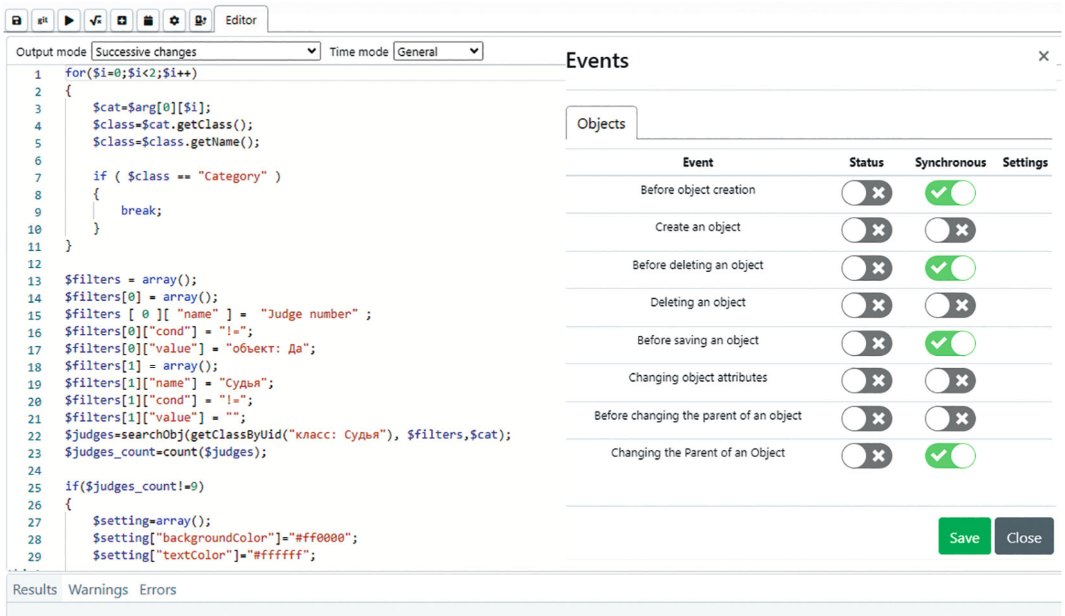

Fig. 8. Internal programming language editor and event configuration for the method.

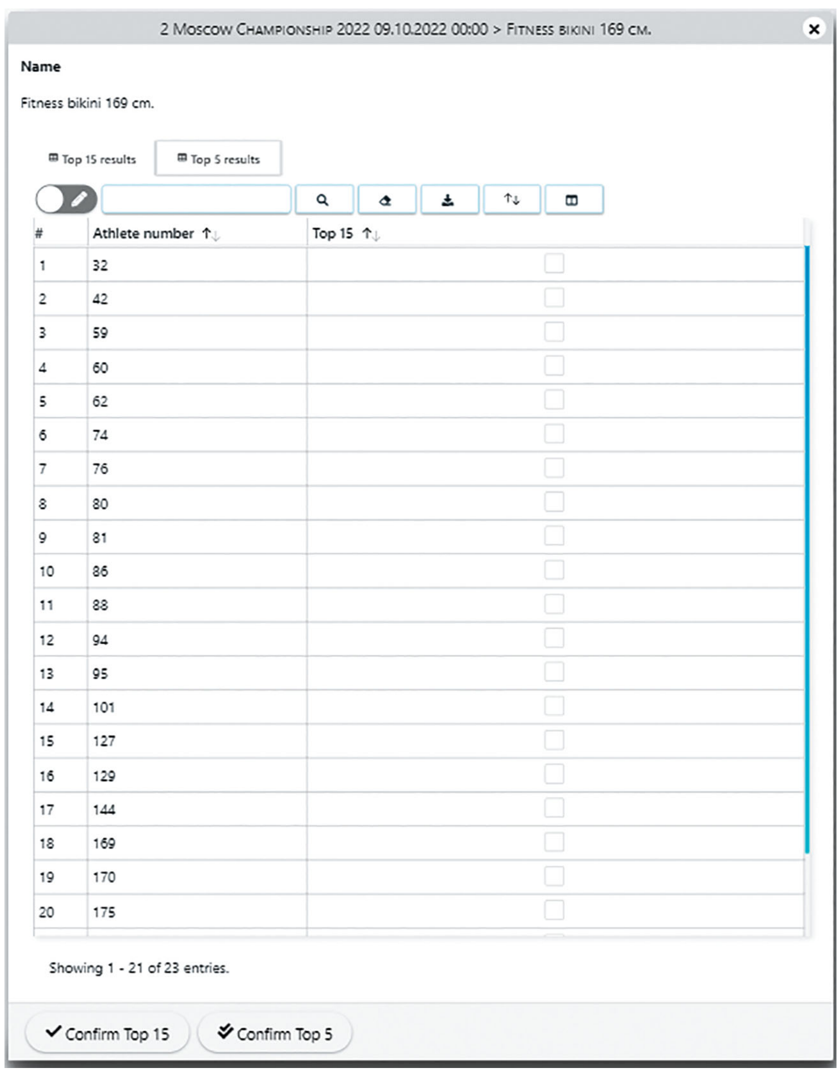

Fig. 9. Assessment by the judge.

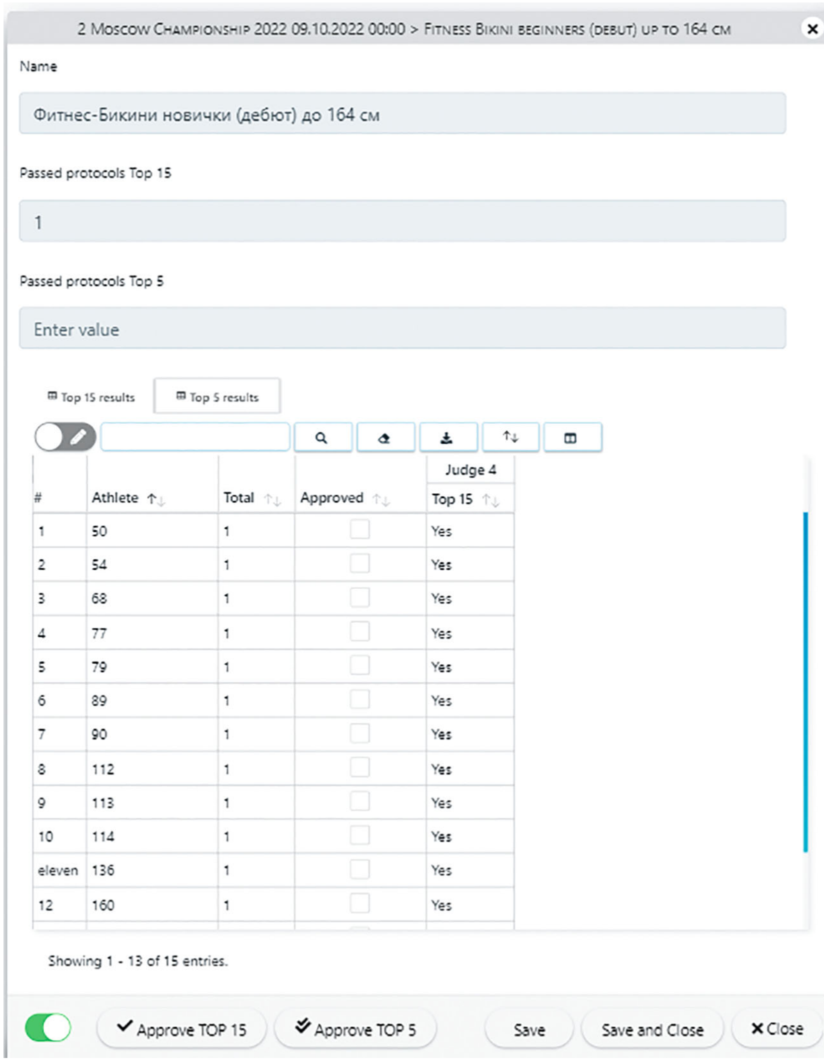

Fig. 10. Approval of the protocol by the chief judge.

The class Evaluation is of interest from the point of view that it does not contain a single independent attribute, but only associative links with other classes (Fig. 7).

Figure 5 shows that the Competition class includes a number of methods necessary for the functioning of the system processes.

The methods are implemented by means of the developed embedded programming language designed for direct use of ontology and system data.

Methods can be initiated when system events occur, or they can be executed manually. Figure 8 shows a general view of the embedded programming language editor and the configuration dialog of methods. Events can occur in both synchronous and asynchronous mode. The editor module has the functionality of tracking changes and versioning and also includes a debugger.

Let us review the basic processes of the Judge and the Chief judge implemented in domain ontology.

The task of the judge is to choose athletes in the category that he judges (Fig. 9). In this case, the associative link from the class Assessment to the class User is used to identify the belonging of the assessment to a particular judge, as well as to distinguish the access of judges to their assessments. The screen form contains the Confirm Top 15 and Confirm Top 5 methods, which the Judge must call manually after scoring.

For the chief judge, the system aggregates data on judges' assessments (Fig. 10). The software-instrumental environment allows to pivot a portion of the table data for a more convenient view. Prior to pivot, the table in Fig. 10 was close to the table in Fig. 9, with additional attributes.

Thus, the same data structure representing the domain ontology and using meta-associative connections can be presented in different ways, depending on the needs of modeling and supporting a particular process.

CONCLUSIONS

Modeling and execution of processes aimed at supporting management decisions in a complex system is, in fact, impossible without building an ontological model of its domain and processes.

The process of constructing the ontology of a complex system is a nontrivial task and requires an integrated approach—ontological engineering (to effectively solve the problem of ontological engineering, universal means of representing the hierarchy, attributes, events, and methods within the ontology are required).

The proposed new type of *meta*-associative graphs designed for ontological engineering of a complex system solves the tasks and shows its effectiveness in real applications.

The ontological model of the subject area and the models of system-defined and information-analytical processes form the basis for management in complex systems, which every year becomes more complex with increasing speed, volume and complexity of data flows accompanying the processes taking place in them. The proposed approach makes it possible to expand and flexibly rebuild the existing ontology in accordance with changing requirements.

The developed specialized software and tools for ontological engineering were tested and showed their effectiveness in solving real problems.

FUNDING

This study was carried out within the framework of the state task, project no. FSWF-2023-0012.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. G. Aguado De Cea, A. Gómez-Pérez, E. Montiel-Ponsoda, and M. C. Suárez-Figueroa, "Natural language-based approach for helping in the reuse of ontology design patterns," in *Knowledge Engineering: Practice and Patterns*, Lecture Notes in Computer Science, Vol. 5268 (Springer, Berlin, 2008), pp. 32–47. https://doi.org/10.1007/978-3-540-87696-0_6
2. A. Basu and R. W. Blanning, *Metagraphs and Their Applications*, Integrated Series in Information Systems, Vol. 15 (Springer, New York, 2007). https://doi.org/10.1007/978-0-387-37234-1
3. A. V. Bobryakov, V. V. Borisov, A. E. Misnik, and S. A. Prokopenko, "Approaches to the implementation of information-analytical processes in complex technical-organizational systems," in *V Int. Conf. on Information Technologies in Engineering Education (Inforino), Moscow, 2020* (IEEE, 2020), pp. 1–5. https://doi.org/10.1109/inforino48376.2020.9111833
4. V. Borisov, A. Fedulov, and V. Kruglov, *Fuzzy Models and Networks* (Goryachaya Liniya–Telekom, Moscow, 2018).
5. V. Borisov and V. Luferov, "Forecasting of multidimensional time series basing on fuzzy rule-based models," in *XXI Int. Conf. Complex Systems: Control and Modeling Problems (CSCMP), Samara, 2019* (IEEE, 2019), Vol. 2, pp. 217–220. https://doi.org/10.1109/cscmp45713.2019.8976821
6. M. V. Chernovalova, "Fuzzy case models for project management using a multi-ontology approach," J. Appl. Inf. **16** (2), 4–16 (2021). https://doi.org/10.37791/2687-0649-2021-16-2-4-16
7. M. V. Chernovalova, T. V. Kakatunova, I. V. Volkova, and E. A. Vlasova, "Algorithms and soft for adapting the knowledge base of project management information systems," J. Appl. Inf. **16** (4), 21–34 (2021). https://doi.org/10.37791/2687-0649-2021-16-4-21-34
8. Yu. E. Gapanyuk, T. I. Khanmurzin, A. A. Kostyan, A. A. Fadeyev, and N. R. Brysina, "Using the metagraph approach in conceptual modeling," Dinamika Slozhnykh Sistem XXI Vek **14** (2), 54–62 https://doi.org/10.18127/j19997493-202002-06
9. M. M. Gorbunov-Posadov, *Expandable Programs* (Polyptych, Moscow, 1999).
10. C. S. Jensen, M. D. Soo, and R. T. Snodgrass, "Unifying temporal data models via a conceptual model," Inf. Syst. **19**, 513–547 (1994). https://doi.org/10.1016/0306-4379(94)90013-2
11. E. F. Kendall, D. L. Mcguinness, and P. Ding, *Ontology Engineering*, Synthesis Lectures on the Semantic Web, Vol. 9 (Morgan and Claypool, 2019).
12. A. E. Misnik, S. A. Prakapenka, and V. V. Kutuzov, "Development of information-analytical processes in cyber-physical systems based on neural-fuzzy petri nets," in *Int. Multi-Conf. on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, 2020* (IEEE, 2020), pp. 1–6. https://doi.org/10.1109/FarEastCon50210.2020.9271400
13. N. G. Leveson, M. Heimdahl, H. Hildreth, and J. D. Reese, "Requirements specification for process-control systems," IEEE Trans. Software Eng. **20**, 684–707 (1994). https://doi.org/10.1109/32.317428
14. B. Meyer, *Object-Oriented Software Construction* (Prentice Hall, New York, 1988).
15. A. Misnik, S. Krutalevich, S. Prakapenka, and E. Lukjanov, "Methodology for the development of industrial analytical systems for data collection and processing," CEUR Workshop Proc. **2475**, 223–231 (2019).
16. A. E. Misnik, "Metagraphs for ontological engineering of complex systems," J. Appl. Inf. **17** (2), 120–132 (2022). https://doi.org/10.37791/2687-0649-2022-17-2-120-132
17. T. Rak, "Modeling web client and system behavior," Information **11**, 337 (2020). https://doi.org/10.3390/info11060337
18. A. S. Tanenbaum and Maarten van Steen, *Distributed Systems: Principles and Paradigms* (Prentice Hall, 2002).
19. E. N. Samokhvalov, G. I. Revunkov, and Yu. E. Gapanyuk, "Using metagraphs to describe the semantics and pragmatics of information systems," Vestn. Mosk. Gos. Tekh. Univ. Im. N.E. Baumana. Ser. Priborostr., No. 1, 83–99 (2015).
20. K. Waldén and J.-M. Nerson, *Seamless Object-Oriented Software Architecture: Analysis and Design of Reliable Systems* (Prentice-Hall, 1995).
21. K. V. Zakharchenkov, Zh. A. Mrochek, and T. V. Mrochek, "Algorithm for solution of multicriterion problem of production planning of pipes and shaped products," Sist. Anal. Prikl. Inf., No. 4, 4–10 (2018). https://doi.org/10.21122/2309-4923-2018-4-4-10

Vadim Vladimirovich Borisov graduated from the Smolensk Branch of the Moscow Power Engineering Institute in 1986. In 1991, he defended his dissertation for the degree of Candidate of Engineering Sciences in the Moscow Power Engineering Institute, and in 1997, his dissertation for the degree of Doctor of Engineering Sciences, specialization 05.13.05, on the topic "Research and Development of Multicoordinate Associative Storage Devices and Information Storage and Processing Environment."

In 2000, he was awarded the academic title of senior researcher in the specialty 20.02.12, and in 2002, the academic title of professor in the Department of Computer Engineering.

Professor of the Department of Computer Engineering, Head of the Laboratory of Intelligent Technologies and High-Performance Computing of the Smolensk Branch of the Moscow Power Engineering Institute; Professor of the Department of Management and Informatics of the Moscow Power Engineering Institute; Senior Researcher at the Research Center of the Military Academy of Military Air Defense of the Armed Forces of the Russian Federation.

Under the leadership of Borisov, two dissertations for the degree of Doctor of Engineering Sciences and more than 20 for the degree of Candidate of Engineering Sciences have been prepared and defended.

Author of more than 400 scientific works, including 15 monographs, and more than 30 patents of the Russian Federation for inventions.

On the basis of the data of the RSCI, he is included in the top-100 cited authors of the Russian Federation on the topics "Automation. Computing" and "Cybernetics."

Honorary Worker of Higher Education of the Russian Federation.

Full member of the Academy of Military Sciences of the Russian Federation.

Deputy Editor-in-Chief of the journal *Fuzzy Systems and Soft Computing*. Member of the editorial board of the journals *Neurocomputers: Development and Application*, *Hi-Tech Earth Space Research*, *Systems of Control, Communication and Security*, *Software and Systems*, *Bulletin of MPEI*, *Applied Informatics, Quantitative Philology*, and *Bulletin of the Perm University. Mathematics. Mechanics and Informatics*. Editor-in-Chief of the *International Journal of Information Technologies and Energy Efficiency*.

A well-known specialist in the field of intelligent decision support; fuzzy systems and soft computing; intelligent analysis, modeling, and forecasting of the state of complex systems and processes; and associative systems for storage and processing of information.

Anton Evgen'evich Misnik, in 2004 graduated with honors from the Belarusian-Russian University, Mogilev, Belarus, specializing in "Automated Information Processing Systems." In 2007, he received his Master's degree from the Belarusian-Russian University majoring in "System Analysis, Management, and Information Processing." From 2010 to 2013 he was a graduate student of the Smolensk Branch of the Moscow Power Engineering Institute. In 2014, he defended his Candidate's dissertation at the Moscow Power Engineering Institute, Moscow, Russia, specializing in "05.13.01—System Analysis, Management, and Information Processing" on the topic "Combined Neural Network Method, Models, and Tools for Operational Management of Complex Technical Systems." From 2020 to present, he has been studying at the doctoral level at the Moscow Power Engineering Institute in the Department of Management and Informatics.

Associate Professor of the Department of Information Technology Software, Scientific Director of the Laboratory of Cyber-Physical Systems, Belarusian-Russian University, Mogilev, Belarus.

He is the author of more than 70 scientific works, including 14 in publications indexed Scopus and Web of Science.

Specialist in the field of intellectual decision support, ontological engineering, intellectual analysis, modeling and forecasting of the state of complex systems and processes.