Forming a rule base for PLM systems

Anton Romanov[1[0000−0001−5275−7628]], Aleksey Filippov[1[0000−0003−0008−5035]], Nadezhda Yarushkina[1[0000−0002−5718−8732]], and Liliya Kamaletdinova[1[0009−0006−2929−9757]]

Ulyanovsk State Technical University, Ulyanovsk, Russian Federation
romanov73@gmail.com
https://ulstu.ru/

Abstract. In production, there are tasks of the integration of a PLM systems and a third-party information system. This system may not be part of the PLM complex, but provides information support for managing production processes. An analyst is currently responsible for carrying out the integration. He must form a structural and process models of the integrated information system to make interaction rules with the system. During the operation process, besides the analyst, there is an operator involved, who performs maintaining the relevance of the integrated information system data and the entire complex, and a decision maker, who performs managing the integrated information system, namely maintaining its operational state. An approach is to reduce the load both on the analyst, who configures the interaction and display of data, and on the operator, who is involved in ensuring the relevance of data structures, and on the decision-maker, who is involved in making important decisions related to risks in production. We propose using data-based management by forming a data meta-model of the integrated information system based on the analysis of its storage; mapping of the data of PLM systems and the integrated information system on the enterprise through the use of a rule base for the behavior of the integrated information system.

Keywords: Rule base · PLM systems · Data-driven management

1 Introduction

As of today, many large manufacturing facilities use PLM systems [1] to manage the product lifecycle. PLM systems are used to control data flows in production, including tasks such as storing, integrating, and maintaining data in each information subsystem within the complex. The data bus plays a key role in this [2]. Each complex of systems with a data bus has its own implementation features [3]. However, the common feature of data bus-based integration is the use of rule-based interaction. To configure the interaction between subsystems, analysts perform the following tasks:

– building a model of the information system behavior,
– identifying key features,

– forming system behavior rules.

This work considers not only the approach to forming interactions between systems, but also the approach to managing PLM subsystems. There is often a need to integrate a PLM system with an external information system (referred to here as the integrable IS) that is not part of the PLM complex but is involved in tasks related to supporting information for managing production processes. At present, an analyst or decision-maker (DM) performs integrating the two systems. The analyst must create structural and process models of the integrable IS based on which interaction rules with the system are formed. During operation, the analyst's (DM) tasks include managing the integrable IS, ensuring the data is up to date, and maintaining the system's operational state.

The figure 1 shows the state graph of the integrable information system.

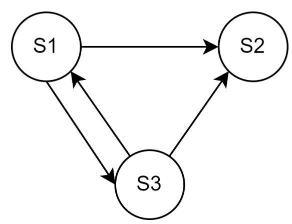

Fig. 1. State graph of an integrated information system

The figure presents the major states of the integrable IS used in this work. $S1$ represents the operational state of the system, $S2$ represents the non-operational state or failure state, and $S3$ represents the limit state of the system [4]. The operational state of the system is when the system continues to perform its primary tasks while its data is accurate. The limit state of the system is when the system continues to perform its primary tasks while the data is in a boundary state. The non-operational state or failure state of the system is when the system cannot continue performing its primary tasks and requires intervention from the DM. While in states $S1$ and $S3$, the system can transition to the failure state, so the main goal of management is to keep the system in state $S1$ and allow it to return to this state without risks if it transitions to state $S3$.

It is worth separately considering the process where the interaction between the integrable IS and the entire PLM complex involves the operator. In this process, the amount of work required by the operator for setting up interaction, data display, and maintaining the system's operational state is considerable. The role of the operator in this process is shown in figure 2.

The analyst must take into account changes in the data structure of the integrable IS. The operator monitors potential risks associated with ensuring the data matches the PLM complex. The DM makes decisions regarding minimizing possible risks. An approach is proposed that helps reduce the workload on both the analyst responsible for setting up interaction and data display and the op-

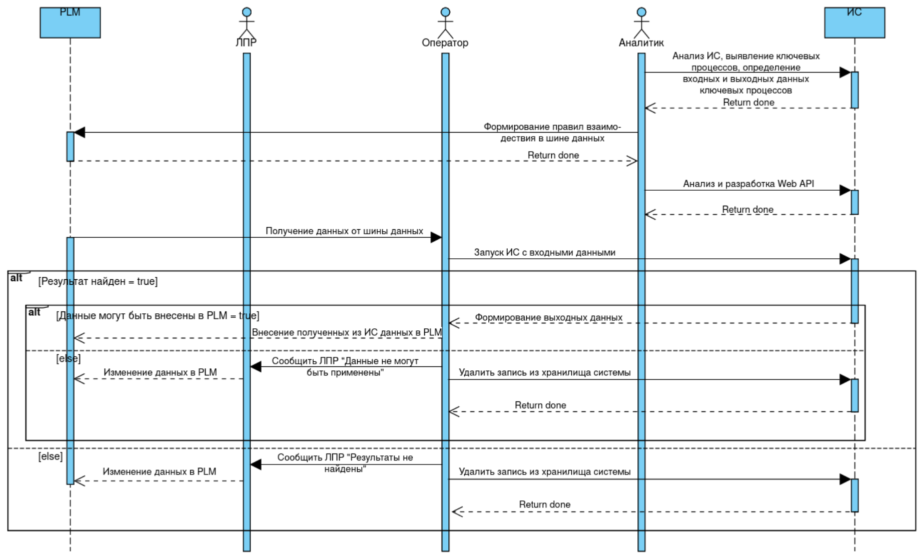

Fig. 2. The process of organizing interaction and maintaining a system in a state of operability with human participation

erator involved in ensuring the data structure's timeliness, and the DM making crucial decisions related to potential risks in production.

1.1 Overview of existing methods and approaches to management

Currently, approaches to managing production information systems are discussed in the following sources: [5–10]. Sources [5, 6, 8] suggest using an integrated IS model. These approaches directly depend on the accuracy and completeness of the model, with the model being formulated either by an analyst [5, 8] or automatically [6], which does not always guarantee the adequacy and accuracy of the model. Source [5] presents classical control based on the model of processes and data in the information system, where the role of the analyst is significant. Source [6] suggests using a framework for model formation. Despite the fact that the model formation process occurs automatically, the controlling component still depends on the accuracy and completeness of the model. Source [7] proposes using a training sample and unsupervised machine learning. However, depending on the completeness and quality of the data sample, the model of the integrated IS being formed may lead to misinterpretation of the results of control mechanisms forecasting, especially if the data contains errors or is heavily noisy. In source [8], a digital twins approach is proposed. Despite the advantages of this approach overall, in the context of the set task, namely integration of information systems, data representation, and reducing the workload of analysts, operators, and managers, the digital twins approach requires additional qualitative analysis and building a model of the information system, leading to an increased workload on the analyst.

In source [9], an approach to human-in-the-loop management is described. The main idea of this approach is complete control of the control system behavior by a human and boils down to supervised machine learning. This approach is the safest in terms of preventing production-related risks, but it is more complex and requires the involvement of managers at all stages of the control system lifecycle, from forming the training sample to monitoring the control system behavior. Therefore, a data-driven control approach is proposed [10]. It is assumed to reduce the workload of the analyst and operator by integrating and partially managing the tasks of the production information system with the developed control system. The data-driven control approach implies:

1. performing system modeling, but not in the classical sense [5], but by forming a data metamodel of the integrated IS based on analyzing its repository;
2. data representation in the integration of PLM systems and the integrated IS of the enterprise by using the rules base of the behavior of the integrated IS;
3. reducing the risks of transitioning to a failure state 1 and maintaining the system in a working state without the involvement of managers in this process.

1.2 Overview of existing approaches to rule base formation

To solve the problem, it is proposed to use a production rule-based model. Classical fuzzy systems are based on the Mamdani approach [11]. In such systems, there are 2 modules for converting regular data into fuzzy data. The fuzzification module establishes a correspondence between real values of input data and fuzzy values, based on the membership function. On the other hand, the defuzzification module establishes a correspondence between fuzzy values and real values of output data of the subject area. Fuzzy rule bases (systems based on fuzzy rules) are based on the principle of converting crisp values into fuzzy values. In this case, rules represent a set of linguistic terms, and output data is associated with them; for example, a rule may have multiple input data and only a certain value of the output parameter. There are several variants of classical systems based on fuzzy rules:

1. Mamdani-type fuzzy rules system [12]. Each rule variable represents a value from a set of linguistic terms, for example. If X_i is a variable represented by a set $\{l_1, l_2, l_3\}$, then in the rule variable X_i can be represented as $\{l_1, l_2\}$. The variable can belong to a set of linguistic terms in the rule. This helps reduce the number of rules to avoid increasing the size of the rule base. Thus, a rule can look like: $x_1 = \{l_{11}, l_{12}\} \wedge x_2 = \{l_{23}\} \wedge ... \wedge x_n = \{l_{n1}, l_{n2}\} \rightarrow y = Y$.
2. Approximate systems based on Mamdani fuzzy rules [13]. Such systems include several elements of term sets, which can reduce the interpretability of the output. Systems can achieve greater accuracy at the cost of losing interpretability. Each rule has its own fuzzy set instead of using linguistic terms.

This approach generates semantically free rules and has more expressiveness due to the use of various fuzzy sets in each rule. It can take different numbers of rules depending on the complexity of the problem. As for the drawbacks, such systems suffer from a loss of interpretability and may overfit training data, performing poorly on unseen data.

3. Classification systems based on fuzzy rules [14]. A classification system based on fuzzy rules is a system that uses fuzzy rules as a means of learning. In classical systems based on Mamdani fuzzy rules, input data is matched with a single-dimensional output, but in this case, input data is matched with one of the class labels. The rule structure looks as follows: $x_1 = l_{1i} \wedge x_2 = l_{2i} \wedge ... \wedge x_n = l_{ni} \rightarrow y = c$.

There are variants of non-classical systems based on fuzzy rules:

1. Hierarchical fuzzy systems [15] consist of several low-dimensional fuzzy systems arranged hierarchically. Rules in hierarchical fuzzy systems are grouped into modules according to their roles in the system. Each module calculates a partial solution, which is then passed to modules of the next level. Although each module is a fuzzy system, it generates a significantly smaller number of rules than a flat fuzzy system. Despite the widespread use of hierarchical rule bases when working with big data, there are a number of drawbacks to such systems. The article [16] provides examples of optimizing hierarchical fuzzy systems using a genetic algorithm. The article [17] provides examples of optimizing cascading hierarchical fuzzy systems using neural networks, but because the proposed method involves all input variables, the advantage of reducing the number of rules is lost.
2. Neuro-fuzzy systems [18] are a merge of systems based on fuzzy rules with artificial neural networks. The main idea of these systems is the ability to make decisions based on given rules and to learn by using neural systems. The rule base is flat (one-dimensional, unlike hierarchical fuzzy systems), which increases the size of the rule base but simplifies the process of training the neural network during system operation. Neuro-fuzzy systems consist of two modules, the first one is responsible for tuning and structuring the rule condition, and the second one is responsible for forming the consequence. Based on existing rules, the neural network adjusts the rule condition and partially calculates the membership function corresponding to the fuzzy input data. In the second stage, the consequence of the rule is calculated from the fuzzy set of consequences to precise output values.
3. Evolutionary fuzzy systems [19]. Genetic algorithms are used to form the rule base. The difference of this type of fuzzy rule bases is that they are self-learning and self-optimizing. Implementation of this approach is done in two stages:
   - in the first stage, a genetic algorithm (GA) is used to find candidates of fuzzy rules in the knowledge base;
   - in the second stage, GA is used for the knowledge base optimization procedure to exclude the worst rules that have little impact on achieving the final solution.

To solve the research problem in data organization, it was decided to combine the hierarchical fuzzy rule base and neuro-fuzzy rule base approaches with a result based on Mamdani fuzzy rules, using an evolutionary algorithm approach for rule formation based on changes in the metamodel. Thus, the rule base will be structured as shown in Figure 3.

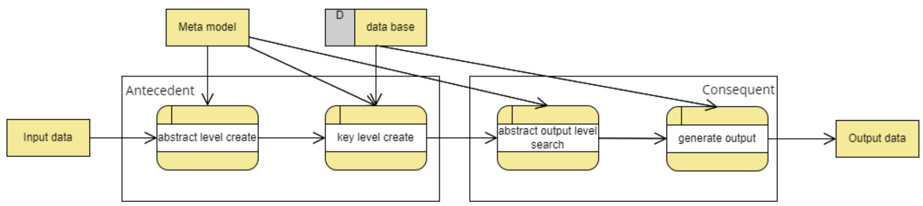

Fig. 3. Structure of the hybrid base with fuzzy logical inference

Thus, according to the diagram, the knowledge base consists of several levels of conditions taking into account possible changes in the metamodel and in the database of the integrated information system, and several levels of consequences forming a clear logical inference based on the linguistic representation of rules in the rule base.

2 Model of the knowledge base of behavior of the integrated information system

According to figure 3, the rule base will be represented as a hierarchical structure with two levels of rules. Previously, the authors obtained a structural model of the metadata M of the integrated IS [20]. Thus, the first level will be represented as rules consisting of linguistic terms and will depend on changes in the metamodel. The second level of rules will be dynamically formed based on the results obtained at the first level.

Let $INP = \{INP_1, INP_2, ..., INP_z\}, z \in N$ be the set of linguistic terms representing the input data of the metadata model M, and $OUT = \{OUT_1, OUT_2, ..., OUT_w\}, w \in N$ be the set of linguistic terms representing the key processes of the metadata model M. Hence, the rule describing the first level will have a set-theoretic representation as follows:

$$P(INP) \to \{INP^{OUT_s}\}, OUT_s, \tag{1}$$

where OUT_s is a linguistic term reflecting a specific key process of the metadata model M, and $\{INP^{OUT_s}\}$ is a set of linguistic terms reflecting the input data for a specific key process of the metadata model M.

Table 1. Tabular representation of the input data for the second level rule

$\mathbf{X_1}$	$\mathbf{X_2}$	...	$\mathbf{X_m}$	$\mathbf{Y}$
v_1^1	v_2^1	...	v_m^1	y^1
v_1^2	v_2^1	...	v_m^2	y^2
...	...	...	...	...
v_1^n	v_2^n	...	v_m^n	y^n

Let's represent $\{INP^{OUT_s}\}$ as X and OUT_s as Y. Table 1 shows the input data for the second-level rule, which will be used to generate the final values of the behavior of the integrated IS.

The columns 1-4 represent the input data values for the key processes of the information system, and column 5 represents the values that are the system's reaction to the input data values.

To form second-level rules of controlling the integrated IS, let's define the following functionality:

$$p(X, V) \to Y, \tag{2}$$

where $V = \{\{v_1^1, ..., v_m^1\}, \{v_1^2, ..., v_m^2\}, ..., \{v_1^n, ..., v_m^n\}\}$ - the input values of m parameters $X = \{x_k\}, k = [1, m], m \in N, n \in N$, $Y = \{y^i\}, i = [1, n], n \in N$ - n states of the integrated IS.

The system state y^i is determined by the vector of input values $\{v_1^i, ..., v_m^i\}$. Thus, to form a rule for controlling the system to transition to a state (issuing control actions) y^i, it is necessary to include a comparison of the parameter vector X with the values $\{v_1^i, ..., v_m^i\}$ in the rule's antecedent as shown in equation 3.

$$p^i(X, \{v_1^i, ..., v_m^i\}) \to y^i. \tag{3}$$

During the operation of the integrated IS, there may be situations where different input parameter values lead to the same state y^i. In this case, they should be grouped, explicitly specifying the same output state.

To account for uncertainty in input values, fuzzy membership functions of triangular form $\mu_(y^i)(x^i)$ [11] will be used. This function of input parameter values x^i inherent in system state i allows for logical inference even when the input value vector contains values that do not precisely match the values used in the rule antecedents.

3 Algorithm for generating output data based on a hierarchical rule base

Figure 4 depicts the decision-making algorithm using a hierarchical fuzzy rule base with fuzzy logical inference based on the Mamdani approach [11].

The algorithm presented in Figure 4 consists of the following steps:

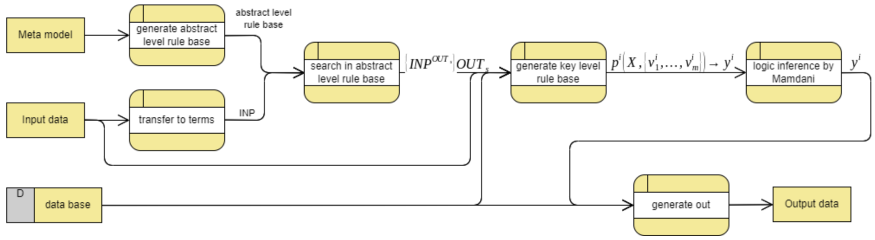

Fig. 4. Decision making algorithm

- Input data, represented as key-value data tuples ($inp1 = 7$) of different types (integer, string, date, and boolean variables), are transformed into linguistic terms represented as $INP = \{INP_1, INP_2, ..., INP_z\}, z \in N$.
- Using the first level rule base (abstract level rule base) and the transformed input data (INP), a logical inference is performed, represented as $\{\{INP^{OUT_s}\}, OUT_s\}, s \in N$.
- The result of the first level rule execution ($\{INP^{OUT_s}\}, OUT_s$), the original input data (key-value data tuples such as $inp1 = 7$) of different types, and the data base of the integrated IS participate in the dynamic formation of second-level rules (generate key level rule base), whose mathematical representation is given in equation 3.
- Through fuzzy logical inference (logic inference by Mamdani) based on the Mamdani approach, the result of rule execution (y^i) is obtained based on the second-level rule base.
- In the final stage (generate out), suitable output data are formed as key-value data tuples ($out1 = 7$) of different types (integer, string, date, and boolean variables). The final stage uses the data base of the integrated IS and the result of rule execution (y^i) based on the second-level rule base as input data.

Thus, the decision-making process is based on a hierarchical fuzzy rule base with fuzzy logical inference.

4 Illustrative example of a hierarchical rule base

As an example, let's consider the rule base obtained based on an information system designed for processing requests, Faveo Service Desk [21]. "

5 Control system in the overall control complex

The proposed approach involves using a control system to reduce the time and effort of the operator involved in the control process. The control system (CS) is a software tool that implements the proposed management approach based on data [20]. Control involves configuration and data exchange, in this case, through the use of the metamodel of the integrated IS and a hierarchical rule base with fuzzy logic inference. The control system acts as an adapter to facilitate the interaction of the integrated IS with the PLM software complex.

Figure 5 shows the process of organizing the management of an information system with the involvement of a control system.

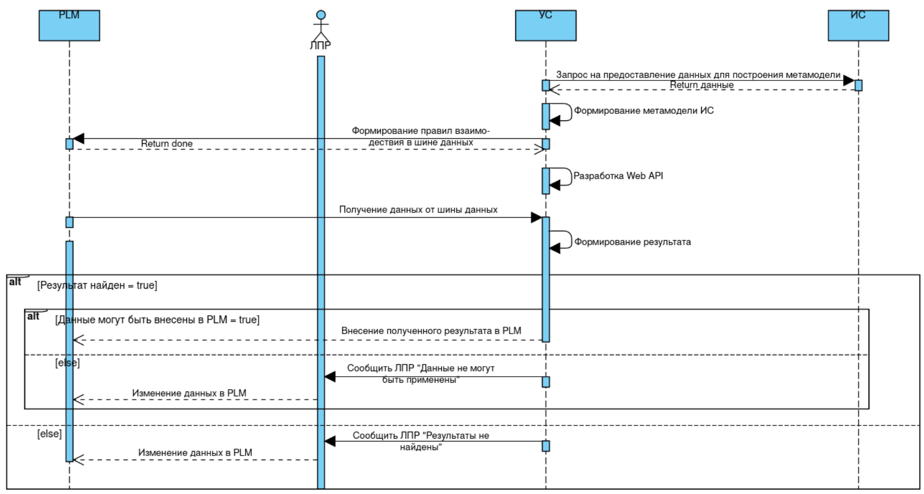

Fig. 5. The process of organizing interaction and maintaining the system in a state of operability with the participation of the management system and the decision maker

When comparing Figures 2 and 5, it is noted that the role of the analyst and the operator is excluded from the control process, their functions are performed by the CS. It is also worth noting that in the management process, there are several conditions in both cases:

1. The possible reaction of the integrated IS to input data exists;
2. The influence of the received reaction on the state of the PLM system (the possibility of adding data to the complex without errors and risks to the entire production complex).

Thus, it can be seen that the main task of the CS is to adjust the data of the overall PLM complex without interacting with the integrated information system. The overall percentage of scenarios in which a person should be involved in working with the system without the CS is 85%, while the total number of scenarios when working with the CS is 18%. The main task, in the case of using the CS, is to react to errors in generating results from the integrated IS. This results in a 67% reduction in the role of the decision maker.

6 Conclusion

The article presents a data-driven management approach applied to production processes. The approach involves using the data metamodel of an information system and a hierarchical rule base for the behavior of the information system with fuzzy logic inference. The main conclusions are provided, confirming the feasibility of using this approach in tasks related to data representation in information systems and managing information systems to reduce the human factor in organizing control of external systems in production. The tasks are relevant in many production facilities of the Russian Federation, including industrial enterprises in the Ulyanovsk region.

Acknowledgments. This study was supported the Ministry of Science and Higher Education of Russia in framework of project No. 075-03-2023-143 "The study of intelligent predictive analytics based on the integration of methods for constructing features of heterogeneous dynamic data for machine learn-ing and methods of predictive multimodal data analysis".

References

1. Terzi S., Bouras A., Dutta D., Garetti M., Kiritsis D. Product Lifecycle Management — from its History to its New Role. International Journal Product Lifecycle Management, 2010, vol. 4, no. 4, pp. 360–389.
2. Menge F. Enterprise Service Bus. Free and Open Source Software Conference, 2007. Available at: https://programm.froscon.org/2007/attachments/15-falko_menge_-_enterpise_service_bus.pdf
3. Shvedenko V. N., Shchekochikhin O. V. Architecture of an integrated information system that ensures behavioral properties (In russian) // Scientific and technical bulletin of information technologies, mechanics and optics. – 2016. – V. 16. – No. 6. – pp. 1078-1083.
4. Chekal E. G. Reliability of information systems: textbook: in 2 parts. Part 1 (In russian) / E. G. Chekal, A. A. Chichev. – Ulyanovsk: UlGU, 2012. – 118 p.
5. Feng, Gang. "A survey on analysis and design of model-based fuzzy control systems." IEEE Transactions on Fuzzy systems 14.5 (2006): 676-697.
6. R. Sudarsan, S.J. Fenves, R.D. Sriram, F. Wang, A product information modeling framework for product lifecycle management / Computer-Aided Design 37 (2005) 1399–1411
7. Alahi MEE, Sukkuea A, Tina FW, Nag A, Kurdthongmee W, Suwannarat K, Mukhopadhyay SC. Integration of IoT-Enabled Technologies and Artificial Intelligence (AI) for Smart City Scenario: Recent Advancements and Future Trends. Sensors. 2023; 23(11):5206. https://doi.org/10.3390/s23115206
8. Sacks R, Brilakis I, Pikas E, Xie HS, Girolami M. Construction with digital twin information systems. Data-Centric Engineering. 2020;1:e14. doi:10.1017/dce.2020.16
9. Ch. Cimini, F. Pirola, R. Pinto, S. Cavalieri, A human-in-the-loop manufacturing control architecture for the next generation of production systems // Journal of Manufacturing Systems. - 2020. - vol. 54. - pp. 258-271
10. Hou Z.S., Xu J.X. On Data-driven Control Theory: the State of the Art and Perspective. Acta Automatica Sinica, 2009, vol. 35, no. 6, pp. 650–667.

11. Mamdani, E.H.: Application of fuzzy algorithms for control of simple dynamic plant. In: Proceedings of the Institution of Electrical Engineers, vol. 121, pp. 1585–1588. IET (1974).
12. Gonza´lez, A., Pe´rez, R., Verdegay, J.L.: Learning the structure of a fuzzy rule: a genetic approach. Fuzzy Syst. Artif. Intell. 3(1), 57–70 (1994)
13. Duckstein, L., et al.: Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological, and Engineering Systems, vol. 8. CRC Press, Boca Raton (1995)
14. Chi, Z., Yan, H., Pham, T.: Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition, vol. 10. World Scientific, Singapore (1996)
15. Razak, T.R., Fauzi, S.S.M., Gining, R.A.J., Ismail, M.H., Maskat, R.: Hierarchical fuzzy systems: interpretability and complexity. Indones. J. Electr. Eng. Inform. 9(2), 478–489 (2021)
16. Zhang, X., Onieva, E., Perallos, A., Osaba, E., Lee, V.C.: Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction. Transport. Res. C: Emerg. Technol. 43, 127–142 (2014)
17. Duan, J.-C., Chung, F.-L.: Cascaded fuzzy neural network model based on syllogistic fuzzy reasoning. IEEE Trans. Fuzzy Syst. 9(2), 293–306 (2001)
18. Jang, J.-S.R., et al.: Fuzzy modeling using generalized neural networks and Kalman filter algorithm. AAAI 91, 762–767 (1991)
19. Komartsova L. G. Evolutionary methods for forming fuzzy rule bases. – 2011.
20. Kamaletdinova, Lilia & Romanov, Anton. (2024). Control System Design (In russian). System Analysis & Mathematical Modeling. 6. 60-77. 10.17150/2713-1734.2024.6(1).60-77.
21. Faveo Service Desk - URL: https://www.faveohelpdesk.com/servicedesk/?lang=ru (accessed at 1.05.2024)