Intelligent System for Simulation Modeling and Research of Information Objects

Yuliia Kostiuk[1,†], Pavlo Skladannyi[1,†], Volodymyr Sokolov[1,*,†] and Svitlana Rzaieva[1,†]

[1] *Borys Grinchenko Kyiv Metropolitan University, 18/2 Bulvarno-Kudryavska str., 04053 Kyiv, Ukraine*

Abstract

The paper discusses an intelligent system for simulation-based modeling of information entities focused on improving the efficiency and accessibility of modeling complex information systems. The system includes an adaptive interface, automated modules designed to generate analytical descriptions and calculation modules, and a mechanism for in-depth analysis of results, which reduces the time for user training and increases the accuracy of the results. One of the key features is the use of logical and linguistic models to select optimal analytical descriptions and create procedural models, which significantly improves the simulation process. Using modern methods of mathematical logic, the system analyzes the structure of analytical formalizations of information objects. It selects optimal solutions based on the similarity of data elements stored in the database. Specialized mathematical models are used to implement operations of logical sequences and build fuzzy relations, significantly improving decision-making accuracy and speed while providing high flexibility in settings. The approach allows automation of the processes of developing and adapting models, reducing the time spent on model revision. As a result, the intelligent system ensures high efficiency of the simulation-based modeling of information entities. This is critical in today's digital environment, where speed and accuracy are the main factors of successful work.

Keywords [1]

Intelligent systems, simulation modeling, information objects, analytical descriptions, fuzzy relations, artificial intelligence.

1. Introduction

Given information technologies' rapid development and integration into all spheres of life, modeling complex systems is becoming increasingly relevant [1, 2]. Simulation modeling, which allows the creation of virtual models of real processes and objects to analyze and predict their behavior, is gaining importance in various fields, including engineering, economics, medicine, and science [3]. However, even with the emergence of robust software solutions such as MATLAB, Maple, or the latest tools for developing complex simulation models, these products require significant training, high user qualifications, and time to master [4]. Modern software products have significant limitations: they require in-depth knowledge of programming and algorithmization and are not intuitive enough for a wide range of users [5]. In addition, they often do not provide flexibility in expanding and supplementing existing models and do not have an effective mechanism for searching for or creating new analytical descriptions [6]. All this complicates the simulation modeling process and limits the speed of obtaining results [7]. Given these problems, there is a need to develop an intelligent system for simulation-based modeling of information entities to provide users with ease of use and flexibility in settings [8]. Such a system

Workshop "Software engineering and semantic technologies" SEST, co-located with 15th International Scientific and Practical Programming Conference UkrPROG'2025, May 13–14, 2025, Kyiv, Ukraine

* Corresponding author.
† These authors contributed equally.
y.kostiuk@kubg.edu.ua (Y. Kostiuk); p.skladannyi@kubg.edu.ua (P. Skladannyi); v.sokolov@kubg.edu.ua (V. Sokolov); s.rzaieva@kubg.edu.ua (S. Rzaieva)
0000-0001-5423-0985 (Y. Kostiuk); 0000-0002-7775-6039 (P. Skladannyi); 0000-0002-9349-7946 (V. Sokolov); 0000-0002-7589-2045 (S. Rzaieva)

should have an intuitive interface to help the user at every simulation stage, providing tips and recommendations for further actions [9]. In addition, the system should support a centralized database containing analytical descriptions, calculation modules, and ready-made procedural models, which will reduce the time needed to develop new models and analytical descriptions when solving similar problems [10]. As a result, such an intelligent system will significantly increase the efficiency of simulation modeling, reducing the need for highly skilled users and the time spent on training and development [11]. All these aspects are essential to ensure that research can be conducted quickly, which is a key factor in today's dynamic digital environment [12].

2. Literature review

With the development of information technology and growing interest in simulation modeling, scientists are actively improving systems for modeling complex objects. Well-known platforms such as MATLAB, Simulink, and Ansys can provide a wide range of simulations but have limitations due to their narrow focus on specific types of tasks. In addition, they require high user qualifications and do not automatically generate analytical descriptions and store results in centralized databases, which complicates the work. These problems can be solved by developing intelligent systems that automate the modeling process, including generating analytical descriptions and model optimization without the need for deep programming knowledge. An important area is integrating artificial intelligence and adaptive interfaces to improve the accuracy of results and facilitate user training.

Well-known scientists who have significantly contributed to developing such intelligent systems are Liu and Shi [1], who studied the role of adaptive interfaces in facilitating user training and automating data analysis. They noted that intelligent interfaces can significantly improve modeling accuracy by automating analysis and generating analytical descriptions. Another area, explored by Lellis Rossi [2], focuses on using neural networks for risk assessment in information systems, which speeds up the modeling process and significantly improves the quality of results due to the ability to adapt to changing object parameters.

An essential role in developing intelligent systems is played by the work of S. Oh and J. Byun [13], who have developed methods for contextual risk assessment in cybersecurity, which allows them to be effectively used in modeling complex information objects. These methods will enable the integration of mathematical logic and artificial intelligence to create reliable and fast models. Also worth noting is the study by H. Rong and Z. Yang [14], who applied fuzzy logic to risk management in information systems, which allows for more accurate models for complex and uncertain processes.

Modern research in this area focuses on creating flexible and adaptive systems that can automate many modeling stages and improve the accuracy and speed of decision-making. This is critically important in the rapid development of digital technologies [15, 16]. The combination of modern approaches to mathematical modeling, intelligent systems, and adaptive interfaces is the basis for creating the latest tools for simulation-based modeling of information entities, which can significantly increase efficiency and reduce complexity.

3. Research methods

To solve the task set, the paper uses modern methods that ensure effective modeling and analysis of information objects. In particular, system analysis methods are used to evaluate and optimize complex information systems, as well as simulation modeling to study various scenarios of information objects' behavior under conditions of uncertainty and variable parameters. The theory of fuzzy sets allows us to consider uncertainty in data and decision-making, providing flexibility in processing incomplete or fuzzy information resources. Numerical analysis was used to perform calculations and evaluate the modeling results, which allowed us to obtain accurate numerical indicators of the effectiveness of various decision options. In addition, artificial intelligence methods, such as machine learning and deep learning, are used to automate the decision-making process and predict results, significantly increasing the accuracy and efficiency of simulation-based modeling of information entities.

4. Summary of the primary material

An intelligent system (IS) for simulation-based modeling of information entities is a comprehensive tool that allows for simulation research and informed decision-making in an interactive mode. Due to the interactive approach to modeling, the system provides the ability to constantly offer the user additional information, instructions, and recommendations for practical work with models, divide the modeling process into separate stages, allowing an individual approach to each stage (automated or manual), which allows for more flexible control of the process and adaptation to specific conditions; reduce the overall time for research by automating routine stages and providing quick access.

The system is structured based on an information array that includes databases such as the analytical descriptions database (ADB), the results database (RD), the calculation modules database (CMDB), and the knowledge base (KB). This array provides storage and processing of data required for efficient simulation studies and real-time decision-making. The IS is developed using the UML language, which allows it to clearly describe its structure and interaction between components [2]. The interaction diagram and functional diagram are shown in Figs. 1 and 2 demonstrate the main components and principles of interaction between them. Such an information structure ensures efficient storage and management of data, which is necessary for the successful performance of simulation modeling within an intelligent system [8].

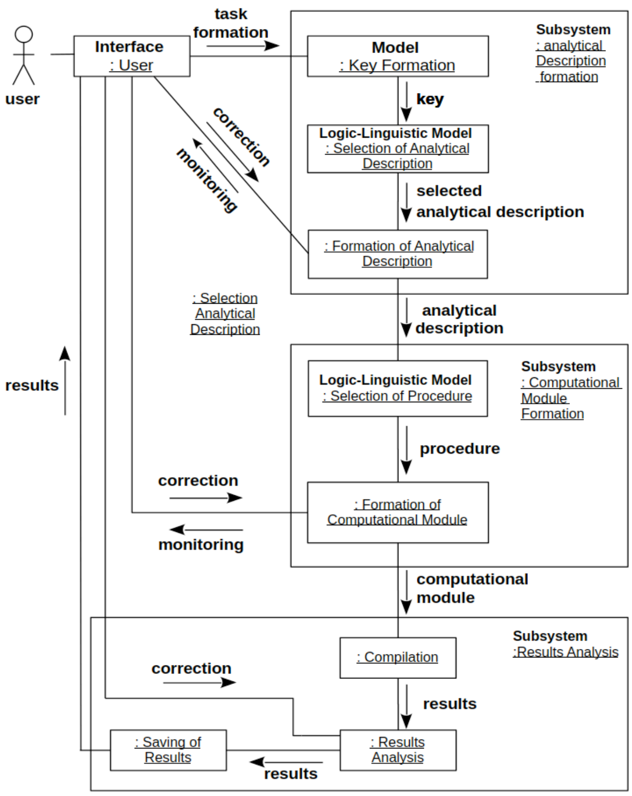

Figure 1: UML diagram of interaction between components of an intelligent system for simulation-based modeling of information entities

The interface of an intelligent system includes two main components: an innovative user and a database interface. The clever interface adapts the information from the system components into a user-friendly form and provides an interactive dialog between the user and the system [6]. The database interface provides opportunities for extracting, editing, and deleting data from various

databases (models, knowledge, calculation modules, and results), effectively manipulating the information required for analysis [10, 17].
The database of analytical descriptions is used to store analytical descriptions necessary for building models and conducting research. The database of calculation modules contains algorithms, ready-made calculation modules, and individual functions, procedures, and source codes in various programming languages [18]. The results database stores the results of completed tasks in multiple formats (text, graphics, tables), which allows for storing essential data about the conditions and initial parameters of the functions. The knowledge base includes fuzzy rules for selecting calculation procedures and analytical descriptions, which allows the system to adapt flexibly to changing research conditions [17]. Databases (database of analytical descriptions, the database of calculation modules, database of results) are interconnected, which ensures efficient information processing and preservation of logical relationships between tables [19].

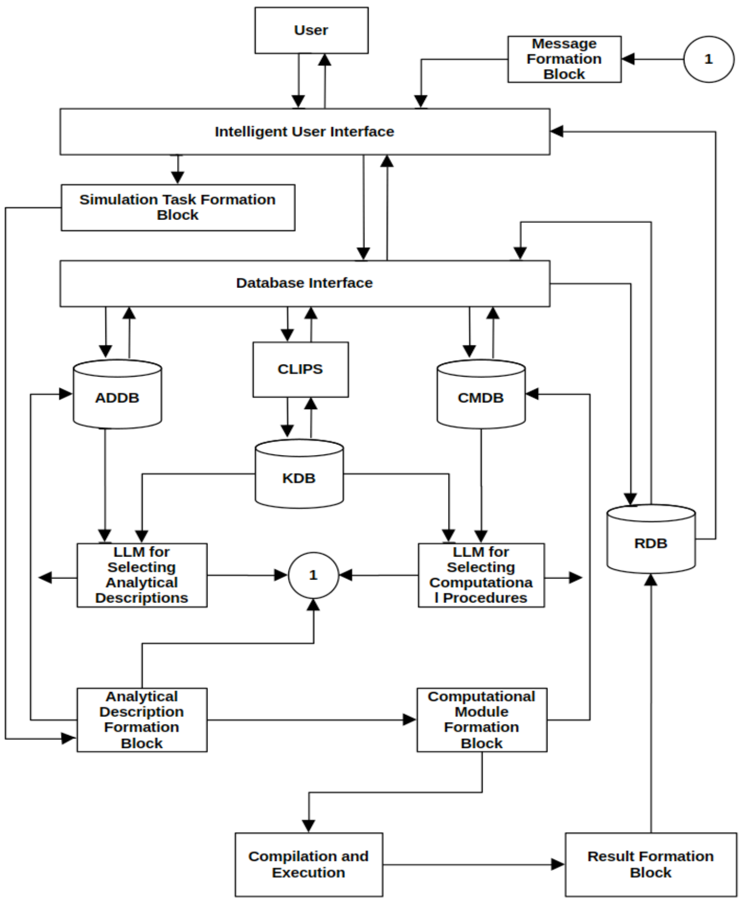

Figure 2: Functional diagram of an intelligent system

An intelligent system for simulation-based modeling of information entities consists of several integrated subsystems: formation of analytical descriptions, creation of calculation modules, and analysis of results. The procedural model of the intelligent system defines data processing algorithms, control logic, decision-making rules, user interaction, and monitoring mechanisms that are used in decision support systems, automated control systems, robotics, expert systems, cybersecurity, and big data analysis (Fig. 3). The modeling process is organized in such a way that at each stage the system adapts to the specific requirements of the study, ensuring the effective use of mathematical and linguistic models to select the most appropriate analytical tools. The analytical descriptions generation subsystem selects and generates descriptions necessary for solving the tasks of the simulation study based on a logical and linguistic model that allows for effective extraction and analysis of available data from the database of analytical descriptions [1, 20, 21]. This process can include using ready-made descriptions and creating new ones in cases where the previously proposed options require further development.

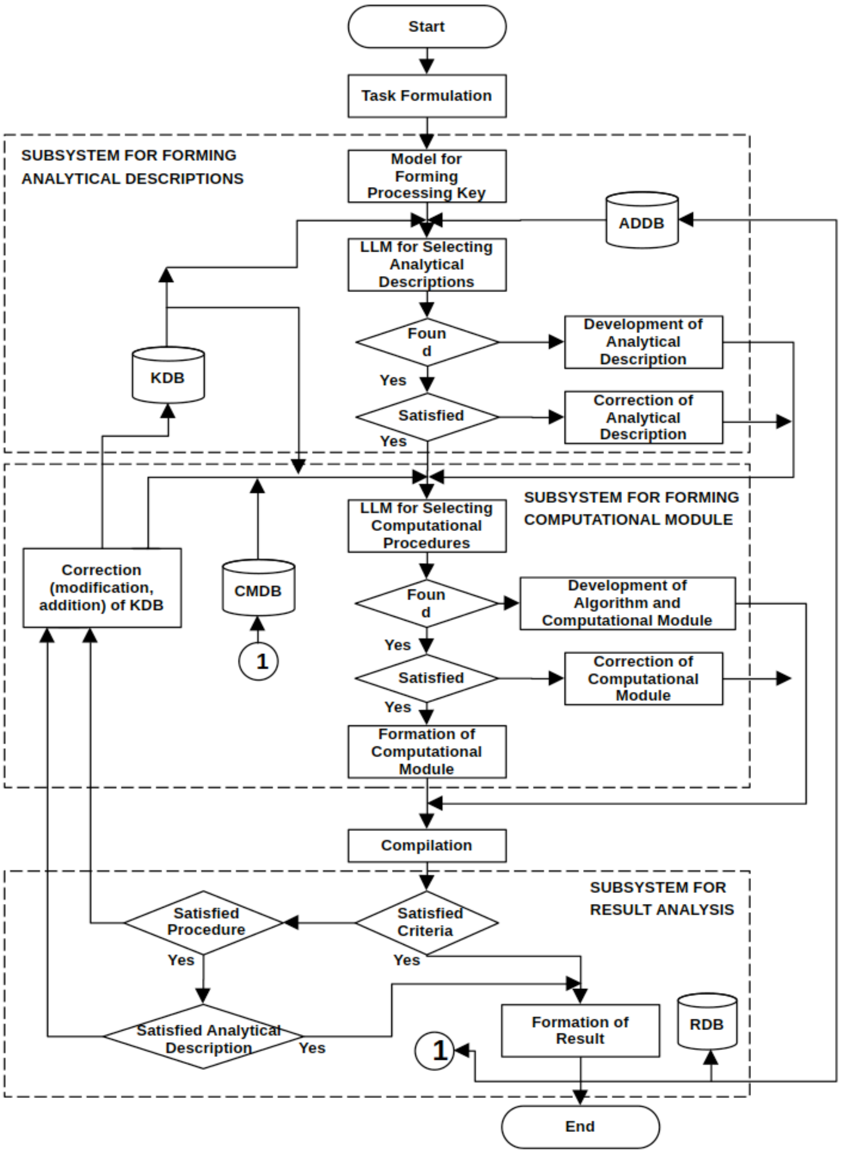

Figure 3: Procedural model of an intelligent system

Based on the logical-linguistic model (LLM), the calculation module formation subsystem selects and adapts procedural models from the database of calculation modules. Without the necessary procedures, the user can independently develop the required mathematical models or adjust the existing ones to the specifics of the problem. Thanks to this structure, the calculation module can be formed based on ready-made components or a new one that meets the requirements of the study [2, 13, 14]. In addition, the user can use functional elements stored in the database of calculation modules to optimize the development process.

After the computational module is generated, it is compiled and run before being used in the results analysis subsystem. At the analysis stage, the user evaluates the results obtained and, if they are unsatisfactory, makes adjustments to the analytical description, calculation procedure, or the module itself, after which the compilation and execution process is repeated until the optimal result is achieved. At the final stage, all the data obtained are stored in the result databases, and analytical descriptions, calculation procedures, and modules—in the corresponding databases of analytical descriptions and databases of calculation modules, thereby providing the possibility of further processing and use of research results [3, 6, 17].

To illustrate the mechanisms of analysis and generation of calculation procedures in an intelligent system, advanced mathematical models can be used to assess the similarity of analytical descriptions and adaptation of calculation procedures. This makes it possible to detail the process of selecting the most relevant tools and modules and optimizing modeling efficiency. A multi-level equation-type index is used to assess the similarity between analytical descriptions [5, 10, 20]. The mathematical model that takes into account the weighting coefficients of each level is as follows:

$$S = \sum_{i=1}^{n} \omega_i \cdot \left| I_1^{(i)} - I_2^{(i)} \right|, \qquad (1)$$

where S is the total difference between the indices of the two analytical descriptions, $I_1^{(i)}$ and $I_2^{(i)}$ are the values of the equation type indices for level i for the two analytical descriptions, ω_i is the weighting factor for level i, which reflects its importance in the overall assessment, n is the number of levels of the equation type index. The smaller the value of S, the more similar these descriptions are. In cases where the value of S is close to zero, we can assume that the descriptions are almost identical. On the contrary, a large value of S indicates significant differences between the descriptions, which may require a deeper analysis or modification of one of the descriptions. In this formula, each level of the hierarchy is taken into account with a given weight value ω_i, which allows us to customize the model to meet specific requirements for assessing similarity. For example, if certain levels are critical to the analysis, their weights can be increased. The multi-level structure of the equation type index allows us to detail and classify analytical descriptions by different levels of detail, taking into account the specifics of each level. Each level corresponds to a specific aspect of the description, for example, the general type of equation, its parametric structure, or functional features. Taking these levels into account provides flexibility and accuracy when comparing different descriptions. Thus, this model is a universal and adaptive tool for classifying, comparing, and evaluating analytical descriptions in complex information systems. It ensures not only the efficiency of the modeling process but also the possibility of its adaptation to the specific needs of the user or task [6].

The LLM for selecting analytical descriptions (LLMVAO) provides an efficient search and adaptation of analytical descriptions in the database to solve specific modeling or analysis tasks. The main goal of this model is to optimize the process of finding appropriate solutions by systematizing and evaluating the parameters that characterize each description. The model's input parameters are variables that reflect various aspects of the description's relevance to the problem. These are the degree of correspondence to the type of equation (TU), the degree of correspondence to the subject area (PO), the degree of correspondence to the input (IN) and output (OUT) parameters, and the degree of correspondence to the keywords (KW) [14]. The generalized evaluation of the task description relevance is calculated using the formula:

$$Q = \omega_{TU} \cdot TU + \omega_{PO} \cdot PO + \omega_{IN} \cdot IN + \omega_{OUT} \cdot OUT + \omega_{KW} \cdot KW, \qquad (2)$$

where Q is a generalized conformity assessment, ω_{TU}, ω_{PO}, ω_{IN}, ω_{OUT} and ω_{KW} are the weighting coefficients that determine the significance of each criterion, and TU, PO, IN, OUT, KW is the value of the relevant criteria. The weighting coefficients are set so that their sum equals one $\omega_{TU} + \omega_{PO} + \omega_{IN} + \omega_{OUT} + \omega_{KW} = 1$, and the criteria values are normalized from 0 to 1 [6].

To adapt the computational procedures, a model is used that considers the sum of the squares of deviations between the values of the input and output parameters of the problem and the selected analytical description. The corresponding formula has the form [14]:

$$\Delta P = \sqrt{\sum_{i=1}^{n} \left(IN_i - IN_i^{/}\right)^2 + \sum_{j=1}^{m} \left(OUT_j - OUT_j^{/}\right)^2}, \qquad (3)$$

where ΔP is the overall assessment of the description adaptation, IN_i, OUT_j is the input and output parameters of the problem, and $IN_i^{/}, OUT_j^{/}$ is the corresponding parameters of the selected description. Suppose the value of ΔP exceeds the permissible threshold. In that case, this indicates a significant discrepancy between the task parameters and the chosen analytical description. It

requires mandatory adjustment of the current description or, if it is not appropriate, additional analysis to find an alternative description that better meets the established criteria.
Thus, the LLM for selecting analytical descriptions has several significant advantages that make it particularly effective for modeling tasks. In particular, it provides a high-efficiency level due to the ability to automate the processes of selecting analytical descriptions, which can significantly reduce the time and resources required to analyze large amounts of information. At the same time, the model demonstrates high flexibility by adjusting the weighting coefficients, and it can be adapted to the specific requirements of the task, taking into account the different significance of criteria, such as compliance with the type of equation, subject area, or keywords. In addition, the modular structure of the LLMBAO allows for easy integration of new evaluation criteria or changes to existing ones, which will enable it to be quickly adapted to new conditions or changes in task requirements. Thus, LLMBAO is a versatile tool capable of supporting all stages of working with analytical descriptions, from their search to adaptation and improvement, which is especially important for solving complex and dynamic modeling problems in modern conditions.

The effectiveness of the LLM for selecting analytical descriptions largely depends on its ability to find relevant descriptions and adapt them to the specific requirements of the task. Adjusting calculation procedures or developing new ones is often necessary when working with a large amount of data and various analytical descriptions. To ensure such adaptability, an approach is used that considers the descriptions' main characteristics and ensures optimal compliance with the task at hand. To adapt computational procedures based on LLMs, a model can be used that includes a set of criteria that take into account the type of equation, subject area, input and output parameters, and keywords. The mathematical model of adaptation can be described as follows:

$$A = \alpha \cdot f_1(TU) + \beta \cdot f_2(PO) + \gamma \cdot f_3(IN, OUT) + \delta \cdot f_4(KW) + \epsilon \cdot g(TU, PO, IN, OUT, KW), \quad (4)$$

where A is adapted calculation procedures, $f_1(TU)$ is a function for assessing the correspondence of the type of equation (TU), $f_2(PO)$ is a function for assessing the correspondence of the subject area (PO), $f_3(IN, OUT)$ is a function that takes into account the relationship between input (IN) and output (OUT) parameters, $f_4(KW)$ is keyword matching function, $KWg(TU, PO, IN, OUT, KW)$ is an integral function that takes into account all the relationships between the parameters,$\alpha, \beta, \gamma, \delta, \epsilon$ are weighting coefficients that determine the importance of the corresponding functions in the overall model [13]. The f_1, f_2, f_3, f_4, and g function can be non-linear, which allows for complex dependencies between parameters. For example, the function $f_3(IN, OUT)$ can be represented as:

$$f_3(IN, OUT) = \frac{\sum_{i=1}^{m} \omega_i \cdot |IN_i - OUT_i|}{m}, \quad (5)$$

where IN_i,OUT_i is the value of the input and output parameters for the$i$ level of the problem, ω_i is the weighting coefficient for the$i$ level, $g(TU, PO, IN, OUT, KW)$ is an integral function that takes into account all the relationships between the parameters,m is the number of parameter levels [10]. The values of the weighting coefficients $(\alpha, \beta, \gamma, \delta, \epsilon)$ are determined empirically or based on historical data, which allows the model to be adapted to specific conditions and tasks. This formula describes a multifactorial process of adaptation of calculation procedures in which model parameters affect the final result individually and through interaction. Depending on the values of the weighting coefficients, the system can prioritize specific criteria, ensuring the model's flexibility.

The integral function g allows us to consider non-linear dependencies that may arise in real-world problems. Thus, the model improves the efficiency of procedure selection and enables the development of new ones that meet complex research requirements. The following model can be used to evaluate the modeling results and make adjustments:

$$R = g\left(P_{\text{output}}, P_{\text{expected}}\right), \quad (6)$$

where R is the result of the evaluation, which determines the degree of compliance of the obtained results with the expected ones, P_{output} is the obtained results of the simulation, P_{expected} is the expected result. The$g$ function defines a metric that can be used to compare the two data sets, such as the mean square error (MSE) or other relevant criteria. In the case of unsatisfactory results, the system can return to previous steps, such as adjusting the analytical description or computational procedures, to improve the result.

In the modeling process, after each stage of compilation and execution, it is necessary to evaluate the results and decide on further steps. This can be described as an iterative process:

$$P_{\text{final}} = \lim_{k \to \infty} (f_k), \tag{7}$$

where P_{final} is the optimal modeling result, f_k is a function representing the$k$ step in adaptation and correction, and k is the iteration index. The iterative process will continue until the desired result is achieved, with the system constantly adapting based on the received at each stage.

A similarity index is used to evaluate the similarity between analytical descriptions, considering the difference between the corresponding elements of the analytical description and the type of equation. This approach allows us to quantify the degree of similarity between the descriptions. If the index value is small, this indicates a high correspondence between the descriptions. The improved mathematical model of the similarity index is as follows [3, 6]:

$$I = \sum_{i=1}^{n} \omega_i \cdot |E_i - T_i|^P, \tag{8}$$

where I is the similarity index that reflects the difference between two analytical descriptions, E_i is the equation element of the i^{th} analytical description that is compared to the corresponding component of the equation type, T_i is the equation type element to be compared to the description element,n is the number of elements in the equation, ω_i is the equation type element to be compared to the description element, p is a parameter that determines the degree of influence of the difference (for example, p = 1 for a linear relationship or p = 2 for a quadratic relationship). The formula for the similarity index considers all elements of the analytical description. It compares them with the equation type's corresponding elements, ensuring the analysis's completeness. To improve the accuracy of the assessment, we use the weighting coefficient ω_i, which allows different values to be set for each equation element. This makes it possible to focus on the parameters that have a greater impact on the similarity of the descriptions. Additionally, the model includes *the p* parameter, which determines the degree of influence of the difference between the elements. For example, choosing a quadratic influence (p = 2) allows for a more substantial penalty for large deviations between items, while a linear influence (p = 1) makes the model more flexible to relatively small differences. The value of the$I$ index serves as a quantitative measure of the degree of similarity between two analytical descriptions. The lower the value of I, the greater the similarity between the descriptions. On the contrary, high index values indicate significant differences between the compared descriptions. This model provides a basis for automating processes to find, adapt, and improve analytical descriptions, which is especially important in complex modeling tasks.

The elements of the two methods (original and adapted) are compared to adjust the calculation procedures. If the difference between them exceeds a specified threshold, the user can adjust the corresponding procedures [2, 17]:

$$\Delta P = \sum_{j=1}^{m} |R_j - C_j|, \tag{9}$$

where ΔP is the difference between the two procedures, R_j is the element of the calculation procedure of the j type, C_j is the element of the adapted calculation procedure,m is the number of elements in the procedure. If the difference ΔP exceeds a certain threshold, it signals that the procedure needs to be adjusted. The user can make changes to the calculation procedure using functional elements from the calculation module database.

We use the optimality index, which compares the calculated results with theoretical values to evaluate the effectiveness of the calculated results. The greater the difference between them, the less optimal the results are [7, 19]:

$$O = \frac{1}{K}\sum_{l=1}^{k}(R_l - T_l), \tag{10}$$

where O is the optimality index that measures the accuracy of the calculated results, R_l is the calculated result at stage l, T_l is the theoretical result at stage l, k is the number of stages in the calculation process. The optimality indicator allows us to determine how closely the results correspond to the theoretical values. The optimality value will be high if the calculated results are close to the theoretical ones [17, 22, 23].

Each analytical description is formed through equations, input, and output variables to perform simulation-based modeling of information entities. The classification of analytical descriptions by the type of equations is carried out using a logical model of key search, which allows the assessment of the degree of similarity between two descriptions and determines the optimal parameters for their combination within the simulation modeling (Fig. 4). This ensures automated selection of the most relevant models for further modeling and analysis of information objects.

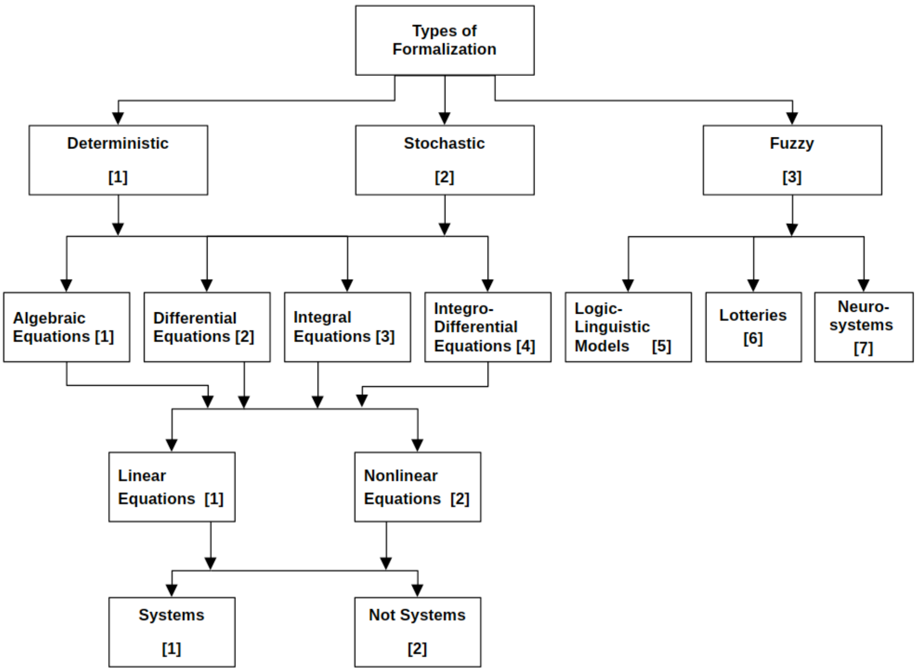

Figure 4: Model of logical key formation

The logical key search model improves the accuracy of simulation modeling by optimally matching input and output variables.

For this purpose, each analytical description is assigned an equation-type index, which is determined by the hierarchical distribution of equations. At the first level of the hierarchy, the type of equation is determined, after which the first digit of the index is set; similarly, at each subsequent level, the following numbers of the index are determined. For example, for deterministic algebraic linear systems of equations, the equation type index will be "1111" [1, 2, 13, 14]. Determining the similarity between two analytical descriptions by equation type involves calculating the difference between the indices modulo. If the number of bits of the index does not match, it should be supplemented with zeros. A more minor difference between the indices indicates a greater similarity of the descriptions, which avoids a lengthy process of developing a new description and improves the existing ones by making only minor adjustments [17].

Each analytical description is assigned an index that determines the type of equation. The index is formed through the hierarchical distribution of equations, where at each level the index number is determined according to the type of equation [5, 12]:

$$I = (i_1, i_2, \dots, i_k), \tag{11}$$

where I is an index of the equation type consisting of k digits, i_j is a digit that determines the type of equation at the j level. Each analytical description is formed as a set of equations with input and output variables. For each level of this hierarchy, a specific index number is defined. For example, for a deterministic algebraic linear system of equations, the index can be “1111,” where each corresponds to a specific type of equation at its level [4, 10].

The difference between their equation-type indices determines the similarity between the two analytical descriptions. The difference is calculated modulo [8, 11]:

$$D = \sum_{j=1}^{k} |i_1(j) - i_2(j)|, \tag{12}$$

where D is the total difference between the equation-type indices of the two descriptions, $i_1(j)$ and $i_2(j)$ are the index numbers at the j level for the two descriptions, and k is the number of levels in the hierarchy. The difference is calculated between the corresponding digits of the indices i_1 and i_2 for the two descriptions at each level. This allows us to measure the distance between the indices and assess the similarity of the descriptions. The smaller the difference, the greater the similarity.

If the number of index bits does not match, it is necessary to supplement them with zeros of the same length to compare them correctly [7, 17, 22, 23]

$$I_1 = (i_1, i_2, \dots, i_k), I_2 = (j_1, j_2, \dots, j_l). \tag{13}$$

If $k \neq l$, we add zeros to the lowercase index:

$$I_1 = (i_1, i_2, \dots, i_k, 0, 0, \dots, 0), I_2 = (j_1, j_2, \dots, j_l, 0, 0, \dots, 0). \tag{14}$$

If the indices have a different number of digits, zeros are added to the smaller index to ensure a correct comparison. This avoids errors in the similarity assessment since both indexes have the same number of elements.

A more minor difference between the indices indicates greater similarity between analytical descriptions. The D difference determines how close a description is to another, and this allows us to optimize the process of developing descriptions:

$$S = \frac{1}{1 + D}, \tag{15}$$

where S is the similarity score between two analytical descriptions, and D is the difference between the equation type indices. The similarity score S can be defined as the inverse function of the difference D. The smaller the difference, the higher the similarity between the descriptions. This definition helps to reduce the time required to develop new descriptions because, with high similarity, the system can use existing descriptions with only minor adjustments.

The development of intelligent systems for simulation-based modeling of information entities involves using modern mathematical tools, including fuzzy logic, to formalize complex processes and make decisions in a multifactorial environment. One of the key steps in building such a solution is to define a membership function that allows us to display a fuzzy relation for each rule. This relation is defined as:

$$\mu_{ij}(X \to Y) = T(\mu_A(X), \mu_B(Y)), \tag{16}$$

where X is a parameter vector with the following components

$$X = \{TU, PO, IN, OUT, KW\}, \tag{17}$$

and Y belongs to the set D, and the dimension of the vector X is defined as $\dim(X) = 5$.

T-implication is a more general and flexible type of implication that is often used in fuzzy sets and fuzzy logic systems to model logical relationships between elements. It includes mathematical operations that take into account fuzziness and fuzzy relationships between components and is formalized as follows:

$$T(\mu_A(X), \mu_B(Y)) = \frac{\min(\mu_A(X), \mu_B(Y))}{\max(\mu_A(X), \mu_B(Y))}, \tag{18}$$

where $j = 1, \ldots, \dim(X)$, and Y is a parameter whose value is determined by solving an optimization problem, which, having an optimal value, ensures accurate adaptation of the model to changing conditions.

The membership function for the fuzzy set B_i is given as:

$$\mu_{B_i}(Y) = \sup_{X \in X} T(\mu_A(X), \mu_B(Y)), \tag{19}$$

where T is the norm, the expression is used, subject to $Y > 0$

$$T(\mu_A(X), \mu_B(Y)) = \frac{\mu_A(X) \cdot \mu_B(X,Y)}{Y + (1 - Y) \cdot (\mu_A(X) \cdot (\mu_B(X,Y))}, \tag{20}$$

The resulting membership function that describes the output of the model is defined as:

$$\mu_{\text{output}} = \max_{i=1,\ldots,\dim(X)} \min T(\mu_A(X), \mu_B(Y)), \tag{21}$$

The proposed model allows for effective adaptation to changing conditions and optimization of system parameters. Thanks to modern algorithms and fuzzy logic methods, the system ensures accuracy and reliability in the simulation-based modeling of information entities, which is vital for analyzing complex technical systems and decision-making.

Simulation-based modeling of information entities is a crucial tool for analyzing complex systems. A logic-linguistic model (LLM) is used to automate the selection of calculation algorithms, which is based on the use of rules, fuzzy sets, and mathematical formulas to determine optimal solutions. Its advantages include fast decision-making, high accuracy due to the adaptation of algorithms, and flexibility in selecting procedures depending on input conditions. The LLM works with four main parameters:

1. TU (logical key) defines the structure and organization of the data used in the system. It allows the system to identify the information required for computation and analysis. TU is critical for correctly selecting algorithms based on a particular data type.

2. TCH (method accuracy) shows how accurate the algorithm results are. This parameter depends on the absolute error. The lower the absolute error, the higher the accuracy, which allows the model to reflect real processes more accurately

$$TCH = \frac{1}{1 + Absolute\ Error}. \tag{22}$$

3. TM (calculation procedure execution time) determines the performance of the algorithm. The execution time characterizes the efficiency of using computing resources. The following formula is used to calculate it

$$TM = \frac{1}{1 + e^{-(t-t_0)}}, \tag{23}$$

where t is the actual execution time, t_0 is the optimal time for this task.

4. TAO (compliance of the procedure with the analytical description) reflects the level of coincidence between theoretical calculations and practical results. The integral indicator calculates it

$$TAO = \int_{x \in X} \mu(x)\,dx, \tag{24}$$

where $\mu(x)$ is a membership function that characterizes the quality of the matching results.

Analyzing the values of these parameters allows us to determine one of three possible scenarios of system operation:

1. If the values of all parameters are high, the system operates in optimal conditions. In this case, a ready-made calculation procedure is used, which ensures fast and efficient execution of the task without the need for additional configuration.

2. If one or more parameters have average values, the system suggests an algorithm that needs to be adjusted. This allows us to adapt the existing approach to the current conditions and ensure it meets the objectives.

3. If all parameters have low values, this indicates significant deviations from optimal conditions. In such a situation, the system recommends developing a new algorithm that considers the task's specifics and current constraints [1, 2, 13].

Integrating such mechanisms into simulation modeling provides efficiency and flexibility in solving complex problems, allowing algorithms to adapt to changing conditions and achieve high accuracy of results [14].

The optimality criterion is used to determine the optimal algorithm:

$$K = \alpha_1 \cdot TCH + \alpha_2 \cdot TM + \alpha_3 \cdot TAO, \tag{25}$$

where α_1, α_2, and α_3 are weighting factors that reflect the importance of each parameter [3].
The decision to choose an algorithm can be represented as an equation:

$$D = \omega_1 \cdot TU + \omega_2 \cdot TCH + \omega_3 \cdot TM + \omega_4 \cdot TAO + \in, \tag{26}$$

where $\omega_1, \omega_2, \omega_3$, and ω_4 are the weighting coefficients of the parameters, $\in$ is the model error [6].

The system's adaptability is also taken into account, which characterizes its ability to respond quickly to changes in input parameters:

$$A = \frac{\Delta TCH}{\Delta t}, \tag{27}$$

where A is the algorithm adaptation speed

For decision-making, LLM uses membership functions based on fuzzy sets [5]:

$$\mu_{ij}(x, y) = T\left(\mu_{A_i}(x), \mu_{B_j}(y)\right), x \in \{TU, TCH, TM, TAO\}, y = D, T(a, b), \tag{28}$$

where $T(a.b)$ is determined by the following formula:

$$T(a.b) = \max\left(1 - [(1-a)^{\gamma} + (1-b)^{\gamma}]^{\frac{1}{\gamma}}, 0\right), \gamma > 1. \tag{29}$$

In this case, the fuzzy relation is used to calculate the supremum:

$$\mu_{B_j}(y) = \min_{x \in X}[\ \mu_{A_i}(x) \wedge T\left(\mu_{A_i}(x), \mu_{B_j}(y)\right), \tag{30}$$

where Λ is the intersection operation of fuzzy sets [20].

Normalization of input parameters is a crucial step to ensure the correct use of TU, TCH, TM, and TAO values. For this purpose, the parameter values are normalized on the interval [0,1]:

$$\hat{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{31}$$

where x_i is the value of the parameter, $x_{\max}, x_{\min}$ are the minimum and maximum values, respectively [12].

The triangular membership function is used to evaluate the boolean key TU:

$$\mu_{TU}(x) = \begin{cases} 0, & IF\ x \leq a\ OR\ x \geq c, \\ \frac{x-a}{b-a}, & IF\ a < x\ \leq b, \\ \frac{c-x}{c-b}, & IF\ b < x\ < c, \end{cases} \tag{32}$$

where a, b, c are the points that define the shape of the triangle.

To improve the stability of the model, a modified T-implication formula is used:

$$\acute{T}(a,b) = a \cdot b - (1-a)(1-b)^{\gamma}, \tag{33}$$

where γ is a parameter that controls the level of smoothing [18].

The aggregation of membership functions to determine the complex membership of all parameters is carried out by the method of summing weight functions [9]:

$$\mu_{agr}(x) = \sum_{i=1}^{n} \omega_i \cdot \mu_i(x), \tag{34}$$

where ω_i is the weighting factor of the parameter i, where $\sum \omega_i = 1$.

The weighting coefficients are calculated based on the Analytic Hierarchy Process (AHP) [22]:

$$\omega_i = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_{j_i}}, \tag{35}$$

where λ_i is the parameter importance score.

Shannon entropy is used to determine the level of uncertainty of the result [19]:

$$H = -\sum_{i=1}^{n} \mu_i(x)\, In\, \mu_i(x). \tag{36}$$

The gradient descent method is used to select the optimal values of the model parameters:

$$\theta_{t+1} = \theta_t - \eta \frac{\partial L}{\partial \theta}, \tag{37}$$

where θ are the model parameters, η is the learning coefficient, and L is the loss function.

The quadratic loss function is used to assess the model's accuracy:

$$L = \frac{1}{2}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{38}$$

where y_i is the real value, $\hat{y}_i$ is the predicted value.

A heuristic criterion is used to determine the best solution:

$$E = \sum_{i=1}^{n} (\omega_i \cdot x_i), x_i \in \{TU, TCH, TM, TAO\}. \tag{39}$$

In general, the LLM ensures the efficiency and accuracy of the choice of algorithms for simulation modeling. It is based on clearly defined rules, mathematical models, and criteria that allow adapting solutions to the specific conditions of the task [17].

The development of an intelligent system for simulation-based modeling of information entities includes hardware, software, and information software, as well as integration with modern web technologies for access via the Internet. The system is implemented through a web interface, which ensures cross-platform compatibility with various operating systems [4, 20]. The MySQL database provides information support for storing and searching analytical descriptions, calculation modules, and results. The system interface allows the user to generate analytical descriptions quickly, select calculation procedures, and enter parameters, ensuring the system's flexibility and adaptability. Using LLM based on fuzzy criteria ensures accuracy and speed of decision-making, and the database structure optimizes modeling and research processes, reducing task execution time and increasing work efficiency.

Conclusion

Developing an intelligent system for simulation-based modeling of information entities is essential for optimizing processes and improving decision-making efficiency. The basis of such a system is the integration of modern technologies, including adaptive interfaces and automated modules designed to generate analytical descriptions and calculation modules, significantly reducing time costs and user qualification requirements. This provides quick access to customized solutions for new tasks and increases the efficiency of simulation modeling. One of the key innovations is the model of logical key formation, which is based on the analysis of analytical formalizations of research objects and uses modern mathematical logic methods to optimize data use. This ensures a reliable and flexible approach to information processing, automating critical research stages and improving the accuracy of results.

The developed LLM for selecting analytical descriptions through similarity analysis and using fuzzy relations simplifies and speeds up the selection of optimal solutions for modeling. The mechanism of implication using one-parameter G-norms (Greenberg norms), which are aimed at improving the decision-making process in complex and uncertain conditions, allows automation of the process of adapting procedural models, which reduces the time spent on model revision and enhances the quality of results. The basis of such a system is the ability to efficiently process and analyze a set of input parameters that affect the efficiency and accuracy of the tasks performed. Since such systems often face large amounts of data and the need to adapt to dynamic changes in conditions, using G-norms allows solving these problems with the help of fuzzy logic and mathematical models.

The use of fuzzy logic in the system allows for effective taking into account uncertainty and ambiguity, reducing the uncertainty of results and increasing the system's stability in the face of changing input parameters. The model adapts to evolving conditions and optimizes parameters, which are vital for analyzing complex technical systems and making real-time decisions.

Thus, an intelligent system for simulation-based modeling of information entities significantly improves the efficiency of scientific research, reduces the time for calculations and modeling, and increases the accuracy and reliability of the results, making it an essential tool for many information systems.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] H. Liu, Z. Shi, R. Han, Y. Zhu, Intelligent Decision-making Modeling based on Object-Oriented Bayesian Network, in: 3rd Int. Conf. on Information and Computing, Wuxi, China, 2010, 300–303. doi:10.1109/ICIC.2010.347

[2] L. Lellis Rossi, et al., A Procedural Constructive Learning Mechanism with Deep Reinforcement Learning for Cognitive Agents, J Intell Robot Syst 110(38) (2024). doi:10.1007/s10846-024-02064-9

[3] S. Singh, S. Kumar, B. K. Tripathi, A Comprehensive Analysis of Quaternion Deep Neural Networks: Architectures, Applications, Challenges, and Future Scope. Arch Computat. Methods Eng. (2024). doi:10.1007/s11831-024-10216-1

[4] L. Chen, H. Ma, J. Wen, UniSim: An Autonomous Multi-Agent Simulation Method with Intelligent Perception, in: 4th International Conference on Information Systems Engineering, ICISE, Shanghai, China, 2019, 48–52, doi: 10.1109/ICISE.2019.00017.

[5] Y. Smitiukh, et al., Development of a Prototype of an Intelligent System for Predicting the Quality of Dairy Production, IEEE Intelligent Systems (2022). doi:10.1109/IS57118.2022.10019699

[6] D. M. Case, C. D. Stylios, Fuzzy Cognitive Map to Model Project Management Problems, in: Annual Conference of the North American Fuzzy Information Processing Society, NAFIPS, 2016, 1–6. doi:10.1109/NAFIPS.2016.7851612

[7] V. Sobchuk, I. Zamrii, Y. Olimpiyeva, S. Laptiev, Functional Stability of Technological Processes based on Non-linear Dynamics with the Application of Neural Networks, Advanced Information Systems 5(2) (2021) 49–57. doi:10.20998/2522-9052.2021.2.08

[8] Y. Kostiuk, et al., Information and Intelligent Forecasting Systems based on the Methods of Neural Network Theory, in: Smart Information Systems and Technologies, SIST, 2023, 168–173.

[9] Y. Kostiuk, Development of Intelligent Components of Information Systems, in: Challenges and Problems of Modern Science: Collection of Scientific Papers, vol. 1, 2023, 337–342. doi:10.6084/m9.figshare.22886720
[10] O. Kryvoruchko, et al., Analysis of Technical Indicators of Efficiency and Quality of Intelligent Systems, Journal of Theoretical and Applied Information Technology 101(24) (2023) 127–139.
[11] Y. Kostiuk, et al., Research of Methods of Control and Management of the Quality of Butter on the Basis of the Neural Network, in: Smart Information Systems and Technologies, SIST, 2022, 106–110. doi:10.1109/SIST54437.2022.9945764
[12] J. Brasse, et al., Explainable Artificial Intelligence in Information Systems: A Review of the Status Quo and Future Research Directions, Electron Markets 33(26) (2023). doi:10.1007/s12525-023-00644-5
[13] S. Oh, J. Byun, Bayesian Uncertainty Estimation for Deep Learning Inversion of Electromagnetic Data, IEEE Geoscience and Remote Sensing Letters 19 (2022) 1–5. doi:10.1109/LGRS.2021.3072123
[14] H. Rong, Z. Yang, Neural Networks, Sequential Intelligent Dynamic System Modeling and Control, Springer, Singapore, 2024. doi:10.1007/978-981-97-1541-1_2
[15] S. Gnatyuk, et al., Method for Managing IT Incidents in Critical Information Infrastructure Facilities, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3826 (2024) 326−333.
[16] P. Anakhov, et al., Protecting Objects of Critical Information Infrastructure from Wartime Cyber Attacks by Decentralizing the Telecommunications Network, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3550 (2023) 240−245.
[17] Y. F. Wang, M. Xie, K. M. Ng, Y. F. Meng, Quantitative Risk Analysis Model of Integrating Fuzzy Fault Tree with Bayesian Network, in: 2011 IEEE Int. Conf. on Intelligence and Security Informatics, 2011, 267–271. doi:10.1109/ISI.2011.5984095
[18] C. Liu, J. Tan, Z. Shi, Research on 3D Intelligent System based on Network Information Technology, in: IEEE 2nd Int. Conf. on Data Science and Computer Application, ICDSCA, Dalian, China, 2022, 512–516. doi:10.1109/ICDSCA56264.2022.9988328
[19] S. Guarino, et al., Holistic Risk Assessment in Industrial Control Systems: Combining Multiple Bayesian Networks with Multi-Criteria Decision Making, in: 32nd Mediterranean Conference on Control and Automation, MED, 2024, 37–42. doi:10.1109/MED61351.2024.10566260.
[20] V. Astapenya, et al., Analysis of Ways and Methods of Increasing the Availability of Information in Distributed Information Systems, in: IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (2021) 174−178. doi: 10.1109/picst54195.2021.9772161.
[21] B. Sahoh, A. Choksuriwong, The Role of Explainable Artificial Intelligence in High-Stakes Decision-Making Systems: a Systematic Review, J Ambient. Intell. Human Comput. 14 (2023) 7827–7843. doi:10.1007/s12652-023-04594-w
[22] L. Jiawei, Risk Assessment of Accounting Information System based on AHP and Fuzzy Comprehensive Evaluation Method, in: 6th Int. Conf. on Computer Sciences and Convergence Information Technology, ICCIT, (2011) 905–908.
[23] J. Schneider, et al., Building Robust Risk Management as a Method of Situational Awareness at the Local Level, in: IEEE Int. Symposium on Technologies for Homeland Security, HST, 2018, 1–7. doi:10.1109/THS.2018.8574167