Baltic J. Modern Computing, Vol. 12 (2024), No. 2, 125-149
https://doi.org/10.22364/bjmc.2024.12.2.01

Model Basis for Cybersecurity of Socio-Cyberphysical Systems

Renata DANIELIENE[1], Sergiy BRONIN[2], Oleksandr MILOV[3], Serhii YEVSEIEV[3]

[1]Vilnius University, Kaunas faculty, Muitines str. 8, Kaunas, 44280
[2]Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska Street, City of Kyiv, Ukraine, 01601
[3]National Technical University "Kharkiv Polytechnic Institute", Building 2, str. Kirpychova, Kharkiv, Ukraine, 61002

renata.danieliene@knf.vu.lt, sergiy.bronin@knu.ua, oleksandr.milov@khpi.edu.ua, serhii.yevseiev@gmail.com

ORCID 0000-0003-3308-0919, ORCID 0000-0003-3094-0450, ORCID 0000-0001-6135-2120, ORCID 0000-0003-1647-6444

Abstract. Designing systems based on high-computational technologies of the Internet of Things, smart and mobile technologies require integrating security across all stages of the lifecycle. Those systems often evolve cyber-physical, socio-cyber-physical systems, which require consideration of their structure as multi-platform, and requires the formation of multi-circuit security systems. At the same time, in each platform (social, cloud and physical) it is necessary to form both internal and external security contours. This approach ensures not only objectivity, but also timely preventive measures to protect information. The article discusses the main approaches to modelling multi-circuit security systems taking into account the physical infrastructure. The proposed approaches provide not only a taxonomy of cybersecurity system models, but also allow to assess the advantages and disadvantages of each class and to ensure the necessary level of objectivity in modelling the security of social-cyberphysical systems.

Keywords: information representation models, socio-cyberphysical systems, computational models, expert models, logical-mathematical models, taxonomy of security models

1. Introduction

The analysis (Carielli et al., 2018, 2020; Hryshchuk, 2016; Shmatko et al., 2020; Yevseiev, Pohasii et al., 2021; Yevseiev, Ponomarenko et al., 2021; Yevseiev, Ryabukha et al., 2021; Yevseiev, Murr et al., 2023; Hryshchuk and Yevseiev, 2016) showed that the construction and operation of a multi-loop security system provides a significant increase in the level of security of the socio-cyberphysical system as a whole, and leads

to a reduction in the risks of disruption of the continuity of business processes in particular. The construction of multi-circuit security systems is based on the need to take into account cyber threats in both internal and external circuits, and also implies the need to integrate methods for ensuring the security of the physical and cybernetic levels of the protected system with methods of social engineering.

The ability to ensure accuracy, consistency, and reliability in the development of security requirements both at different stages of security system development and for various levels of a socio-cyberphysical system provides motivation for the use of formal methods and models of various types (agent-oriented, system dynamics models, discrete-event, analytical, etc.). The subject of formal models and specifications can be illustrated by the different types of security attributes and requirements that appear in published security models.

In the context of information and cyber security, formal methods involve the use of both specialized language and reasoning methods. Informal methods, on the other hand, are written in natural language and rely on common sense. The use of formal notations, especially those with well-understood semantics, can improve the accuracy of security policy formulation. The use of formal proofs can provide additional confidence that certain policy implementation methods satisfy the formal definition of security.

In the proposed work, an attempt is made to classify the models used for the design of security systems largely by areas of application (computational, expert, interactive and set-theoretic). This approach allows us to jointly consider models of cybersecurity systems of various levels of formalization, which is necessary in the process of designing cybersecurity systems.

After considering such a taxonomy of models, it is natural to attempt to analyze various languages for representing the knowledge of experts, users, analysts and other categories of users of cybersecurity systems.

As a result of applying this approach, taking into account various platforms for representing a socio-cyberphysical system, a classification and integration of security models can be proposed, which together form the model basis of the security system of a socio-cyberphysical system.

2. Multi-loop nature of the cybersecurity system of socio-cyberphysical systems

Figure 1 shows a structural and logical diagram of a socio-cyberphysical system (SCS), which clearly represents the multi-loop nature of the cybersecurity system of socio-cyberphysical systems.

In the formal presentation of the relationships reflecting the functioning of both the external and internal circuits of the security system, as well as the models and methods of each of the platforms, the following notations are used.

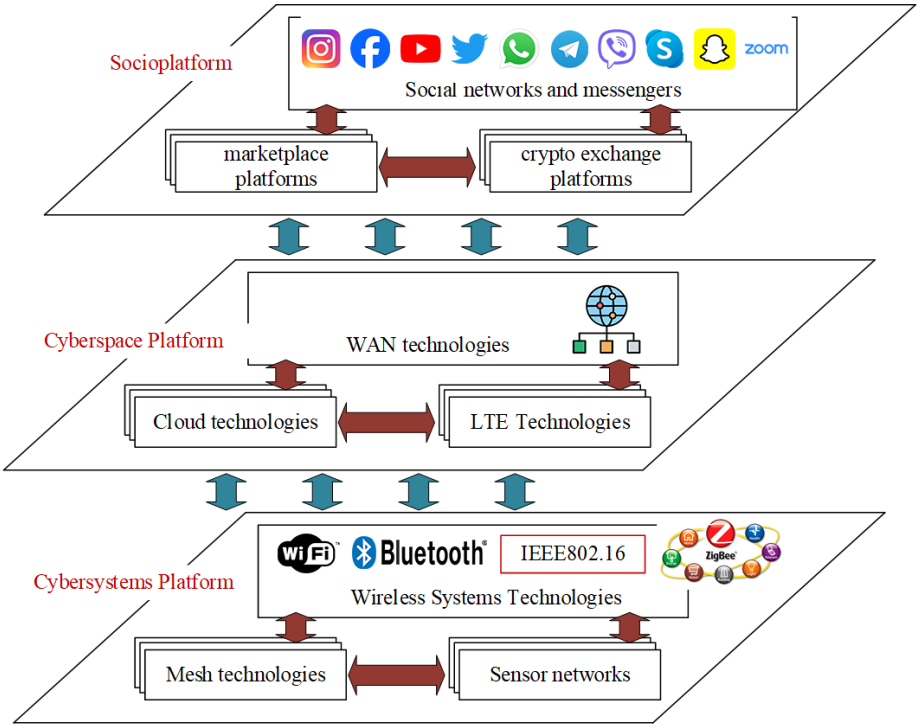

Figure 1. Structural and logical diagram of the socio-cyberphysical system

Variable indices correspond to basic security services: C – confidentiality; I – integrity; A – availability; Au – authenticity, Aff – affiliation; $N_{1_i}^{C}$ – the number of objects providing security service, such as confidentiality; for other security services – the same; Q is the total number of known cyber threats; Tr_R^A is the set of potential threats, the implementation of which is effective for the attacker; Tr_i – threat to the i-th information resource; P_i^A – estimation of the cost of the successful implementation of an attack on the i-th resource from the side of the attacker; C_i^A – the cost of carrying out an attack on the i-th resource by the attacker; Tr_C^D is the set of threats against which it is economically expedient to build protection; P_i^D – estimation of the cost of the loss of the i-th information resource for the defense side; P_i^D – the cost of protecting the i-th information resource for the side of the defense; K_i^A is the rating coefficient (importance) of the implementation of the threat to the i-th information resource; M is the cardinality of the set of selected potentially effective threats for the attacking side; K_j^D is the rating coefficient (importance) of building the protection of the j-th information resource; Q is the total number of known cyber threats;

$\left|\left\{W_{hybrid\ C,I,A,Au,Af}\right\}\right|_{synerg}$ is the power of the set of hybrid threats (i.e., their number), and $\left\{W_{hybrid\ C,I,A,Au,Af}\right\}_{synerg}$ is the set of hybrid threats, which, according to the accepted assumption, are defined as a set of threats simultaneously for all security services; M is the number of threats that are selected by an expert from a set $\left\{i\right\}_i^M$, that is a subset of the entire set of threats of the classifier, that is, $M \le Q$. w_{SCSi}^C, w_{SCSi}^I, w_{SCSi}^A, w_{SCSi}^{Au}, w_{SCSi}^{Aff} – the expert weights of security services: confidentiality, integrity, availability, authenticity and involvement; χ_i^{SCS} is the weighting factor of security services: confidentiality, integrity, availability, authenticity and authenticity of the manifestation of the attack of the *i*-th threat; p_{rj} – probability of realization of at least one threat to the *j*-th asset, *i* – a threat, $\forall i \in n$, n – the number of threats, *j* – information resource (asset), $\forall j \in m$, m – the number of assets; r_{motiv} – the probability of the attacker's motivation to implement the threat; W_{cp}^{SCS} – the computational resources of the attacker; W_{cash}^{SCS} – the attacker's financial resources; μ_{kg}^j – the weighting coefficient of the *g*-th metric of the *j*-th security service for the *k*-th expert. Normalization of weight coefficients: $\sum_{k=1}^{K}\sum_{g=1}^{B}\mu_{kg}^j = 1$, w_{kg}^j – the value of the assessment of the *g*-th characteristic of the information security tool mechanism by the *k*-th expert for the *j*-th security service in the case when the degree of security of the system and the destructive actions of attackers are independent. Wherein *B*={*cryptographic resistance* (C_r), *Key data amount*, S_c, *encryption/decryption of data complexity*, O_E}. Thus, we have so many characteristics of technical means of information security: $\mu^j = \left|\left\{C_r^j, S_c^j, O_E^j\right\}\right|$ $\mu^j = \left\{C_r^j, S_c, O_E\right\}$, which corresponds to the level of security of cryptographic means of information security. To describe the set of characteristics, we use the index *g*: μ_g, where $\left(\left\{g\right\}_1^B\right)$. the coefficients ε, ζ > 0, and describe the damage inflicted on themselves by the "prey" and "predator", respectively.

To design *a cyber systems platform security model*, it is necessary to take into account:

– internal loop:

$$Q_{ISL_1}^{SCS} = A_1^{SCS} \bigcup A_3^{SCS} \bigcap A_2^{SCS} \bigcap A_4^{SCS},$$

– external loop:

$$Q_{ESL_1}^{SCS} = A_1^{SCS} \bigcup A_3^{SCS} \bigcap A_2^{SCS} \bigcap A_4^{SCS};$$

To design *a cyberspace platform security model*, it is necessary to take into account:

– internal loop:

$$Q_{ISL_2}^{SCS} = A_1^{SCS} \bigcup A_2^{SCS} \bigcup A_3^{SCS},$$

– external loop:

$$Q_{ESL_2}^{SCS} = A_1^{SCS} \bigcup A_2^{SCS} \bigcup A_3^{SCS} \bigcap A_4^{SCS};$$

To design *a social media platform security model*, you need to consider:

– internal loop:

$$Q_{ISL_3}^{SCS} = A_1^{SCS} \bigcup A_2^{SCS} \bigcup A_3^{SCS},$$

– external loop:

$$Q_{ESL_3}^{SCS} = A_1^{SCS} \bigcup A_2^{SCS} \bigcup A_3^{SCS} \bigcap A_4^{SCS}.$$

where

$$A_1^{SCS} = \begin{cases} \dfrac{dN_1}{dt} = \left(\arg \max\limits_{\forall Tr_l \in Tr_C^D} K_l^D \times K_l^A \right) \times \\ \times \left(\sum\limits_{i=1}^{Q} \begin{pmatrix} N_{1_i}^C \times A_i^C + N_{1_i}^I \times A_i^I + N_{1_i}^A \times A_i^A + \\ + N_{1_i}^{Au} \times A_i^{Au} + N_{1_i}^{Aff} \times A_i^{Aff} \end{pmatrix} \right) - \\ - \left(\sum\limits_{i=1}^{M} \left(w_{SCSi}{}^C \bigcap w_{SCSi}{}^I \bigcap w_{SCSi}{}^A \bigcap w_{SCSi}{}^{Au} \bigcap w_{SCSi}{}^{Aff} \right) \chi_i^{SCS} \right) \tilde{N}_1 \times \\ \times \left(N_2 \times \left| W_{\text{hybrid } C,I,A,Au,Af\ synerg} \right| \right); \\ \dfrac{dN_2}{dt} = -\left(\dfrac{1}{M} \sum\limits_{i=1}^{M} \nu_i \times p_{rj} \times r_{motiv} \right) \tilde{N}_2 + \\ + \left(\dfrac{1}{KB} \sum\limits_{k=1}^{K} \sum\limits_{g=1}^{B} \left(\mu_{kg}^l \times w_{kg}^l \right) \right) \tilde{N}_2 \tilde{N}_1. \end{cases},$$

and $\tilde{N}_1 = \sum\limits_{i=1}^{Q} \begin{pmatrix} N_{1_i}^C \times A_i^C + N_{1_i}^I \times A_i^I + N_{1_i}^A \times A_i^A + \\ + N_{1_i}^{Au} \times A_i^{Au} + N_{1_i}^{Aff} \times A_i^{Aff} \end{pmatrix}$ – number of objects representing attack targets, taking into account their hybridity;

$\tilde{N}_2 = N_2 \times \left| W_{\text{hybrid } C,I,A,Au,Af\ synerg} \right|$ – equation for changes in the number of modern SCS threats, taking into account the possibility of their signs of synergy and hybridity;

where

$\left| W_{\text{hybrid } C,I,A,Au,Af\ synerg} \right|$ – the power of multiple hybrid threats (that is, their number)

and

$$W_{\text{hybrid } C,I,A,Au,Af\ synerg} = W_{synerg}^C \bigcap W_{synerg}^I \bigcap W_{synerg}^A \bigcap W_{synerg}^{Au} \bigcap W_{synerg}^{Aff},$$

where there are many hybrid threats, which, according to the accepted assumption, are defined as many threats simultaneously for all security services. The calculation of individual components is given in (Shmatko et al., 2020).

Threat dynamics are determined using the Lotka-Volterra model:

$$A_2^{SCS} = \begin{cases} \dfrac{dN_1}{dt} = \left(\arg \max\limits_{\forall Tr_l \in Tr_C^D} K_l^D \times K_l^A \right) \times \\ \times \left(\sum\limits_{i=1}^{Q} \begin{pmatrix} N_{1_i}^C \times A_i^C + N_{1_i}^I \times A_i^I + N_{1_i}^A \times A_i^A + \\ + N_{1_i}^{Au} \times A_i^{Au} + N_{1_i}^{Aff} \times A_i^{Aff} \end{pmatrix} \right) - \\ - \left(\sum\limits_{i=1}^{M} \left(w_{SCSi}{}^C \bigcap w_{SCSi}{}^I \bigcap w_{SCSi}{}^A \bigcap w_{SCSi}{}^{Au} \bigcap w_{SCSi}{}^{Aff} \right) \chi_i^{SCS} \right) \times \\ \times \tilde{N}_1 \left(\tilde{N}_2^1 \bigcap \tilde{N}_2^2 \bigcap \ldots \bigcap \tilde{N}_2^w \right); \\ \dfrac{dN_2}{dt} = - \left(\dfrac{1}{M} \sum\limits_{i=1}^{M} \nu_i \times p_{rj} \times r_{motiv} \right) \times \left(\tilde{N}_2^1 \bigcap \tilde{N}_2^2 \bigcap \ldots \bigcap \tilde{N}_2^w \right) + \\ + \left(\dfrac{1}{KB} \sum\limits_{k=1}^{K} \sum\limits_{g=1}^{B} \left(\mu_{kg}^l \times w_{kg}^l \right) \right) \times \left(\tilde{N}_2^1 \bigcap \tilde{N}_2^2 \bigcap \ldots \bigcap \tilde{N}_2^w \right) \tilde{N}_1, \end{cases}$$

where the number of "predators" (attackers) belongs to the set $\overline{\left\{ \tilde{N}_2^j \right\}}, j \in 1, \ldots Q.$

$$A_3^{SCS} = \begin{cases} \dfrac{dN_1}{dt} = \left(\arg \max\limits_{\forall Tr_l \in Tr_C^D} K_l^D \times K_l^A \right) \times \\ \times \left(\sum\limits_{i=1}^{Q} \begin{pmatrix} N_{1_i}^C \times A_i^C + N_{1_i}^I \times A_i^I + N_{1_i}^A \times A_i^A + \\ + N_{1_i}^{Au} \times A_i^{Au} + N_{1_i}^{Aff} \times A_i^{Aff} \end{pmatrix} \right) - \\ - \left(\sum\limits_{i=1}^{M} \begin{pmatrix} \left(w_{SCSi}{}^C \bigcap w_{SCSi}{}^I \bigcap w_{SCSi}{}^A \bigcap w_{SCSi}{}^{Au} \bigcap w_{SCSi}{}^{Aff} \right) \times \\ \times \chi_i^{SCS} \end{pmatrix} \right) \times \\ \times \tilde{N}_1 \left(\sum\limits_{j=1}^{w} \tilde{N}_2^w \right); \\ \dfrac{dN_2}{dt} = - \left(\dfrac{1}{M} \sum\limits_{i=1}^{M} \nu_i \times p_{rj} \times r_{motiv} \right) \left(\sum\limits_{j=1}^{w} \tilde{N}_2^w \right) + \\ + \left(\dfrac{1}{KB} \sum\limits_{k=1}^{K} \sum\limits_{g=1}^{B} \left(\mu_{kg}^l \times w_{kg}^l \right) \right) \times \left(\sum\limits_{j=1}^{w} \tilde{N}_2^w \right) \tilde{N}_1, \end{cases}.$$

$$A_4^{SCS} = \begin{cases} \dfrac{dN_1}{dt} = \left(\arg \max\limits_{\forall Tr_l \in Tr_C^D} K_l^D \times K_l^A \right) \times \\ \times \left(\sum\limits_{i=1}^{Q} \left(\begin{array}{l} N_{1_i}^C \times A_i^C + N_{1_i}^I \times A_i^I + N_{1_i}^A \times A_i^A + \\ + N_{1_i}^{Au} \times A_i^{Au} + N_{1_i}^{Aff} \times A_i^{Aff} \end{array} \right) \right) - \\ - \left(\begin{array}{l} \sum\limits_{i=1}^{M} \left(w_{SCS\,i}{}^{C} \cap w_{SCS\,i}{}^{I} \cap w_{SCS\,i}{}^{A} \cap w_{SCS\,i}{}^{Au} \cap w_{SCS\,i}{}^{Aff} \right) \times \\ \times \chi_i^{SCS} \end{array} \right) \times \\ \times \tilde{N}_1 \left(\sum\limits_{j=1}^{w} \tilde{N}_2^w \right) - \varepsilon \tilde{N}_1^2; \\ \dfrac{dN_2}{dt} = -\left(\dfrac{1}{M} \sum\limits_{i=1}^{M} \nu_i \times p_{rj} \times r_{motiv} \right) \left(\sum\limits_{j=1}^{w} \tilde{N}_2^w \right) + \\ + \left(\dfrac{1}{KB} \sum\limits_{k=1}^{K} \sum\limits_{g=1}^{B} \left(\mu_{kg}^l \times w_{kg}^l \right) \right) \times \left(\sum\limits_{j=1}^{w} \tilde{N}_2^w \right) \tilde{N}_1 - \xi \tilde{N}_2^2, \end{cases}$$

Thus, taking into account the proposed approach, the general model of a multi-loop security system of a socio-cyberphysical system can be presented as:

$$Q_{general}^{SCS} = \left(Q_{ISL_1}^{SCS} \cup Q_{ESL_1}^{SCS} \right) \cup \left(Q_{ISL_2}^{SCS} \cup Q_{ESL_2}^{SCS} \right) \cup \left(Q_{ISL_3}^{SCS} \cup Q_{ESL_3}^{SCS} \right).$$

The proposed models practically allow us to take into account the modern financial and computing capabilities of attackers. In addition, the proposed approach takes into account the possibility of not only influencing the victim by combining targeted (mixed) attacks with social engineering methods, but also taking into account the socio-political-economic state of the society in which the victim is located.

One of the directions for increasing the efficiency of operational management of cybersecurity systems is the creation of software and hardware decision-making systems that automate the processes of recognizing situations and finding control solutions in the context of cyber attacks (Milov, 2019). As a search area for solutions in such systems, a knowledge model is used that integrates all aspects that characterize the operation of both protected critical infrastructure systems and cybersecurity systems, methods, techniques and algorithms for managing systems. The construction of such a search area requires the development of means for generating appropriate models of the cyber environment and the control object, recognizing and classifying situations, searching and developing solutions, and more, i.e., all information inherent in the area of cybersecurity management of critical infrastructure systems.

At the same time, the search area must be constructive, i.e., allow the automation of human-machine decision-making processes, taking into account real-time security management of the protected system.

The considered area of application of a cybersecurity system is characterized by the presence of a large number of algorithms, the targeted combination of which into computational and interactive models makes it possible to find control solutions, the

implementation of which makes it possible to ensure the appropriate level of cybersecurity of the protected system. At the same time, this area is also characterized by descriptions of the processes of searching for solutions, presented in the form of certain sets of statements (expert and set-theoretic models). All information (a model of management activity in operational decision making), or rather its representation, is based on the concept of a formal system, this allows us to build a unified representation of different types of information, agreed upon within a single formalism (Hofman, 2017).

Considering that information characterizing the control area can be processed by four types of models, we will determine the coverage of this area by each type of model (see Table 1).

Table 1. Types of Decision Support System (DSS) models and areas of their application

Model type	Application area
Computational models	operational dispatch control problems described by methods of computational mathematics
Expert models	tasks of developing and making management decisions
Conversational Models	tasks of human-machine interaction in management, training and design
Set-theoretic models	description of data elements characterizing the control area

Analysis of the use of different types of models showed that their joint use is not only possible, but also necessary. Therefore, in the system for ensuring the security of critical infrastructure objects there must be a mechanism that allows one to represent multiple models in some related composition and have a mechanism for accessing the models and any of their elements.

The design of a mechanism for representing multiple models in computer memory has a real basis, because the formal system serves to create means of integral description and presentation of information about all types of system models.

A unified access mechanism should be considered in two aspects - access for the purpose of automated or automatic search and decision-making, and access for the purpose of determining the model basis.

The purpose of the article is to describe the proposed taxonomy of cybersecurity system models, which includes four sets of models: computational models, expert models, conversational models and set-theoretic models. A method is proposed for formalizing each of these classes of models and a method for representing a cybersecurity system in memory.

3. Petri nets as a universal means of representing models in cybersecurity systems

Petri nets have gained wide recognition as a convenient and visual tool for describing models and processes of information transformation (Baez and Master, 2020; Cassandras and Lafortune, 2021; Zhou and Wu, 2018). Unlike traditional automata, Petri nets make it possible to describe many different types of models and the processes occurring in

them by establishing local relationships between components and tracking local changes in the states of the entire system of models (Cantrell, 2021; Grobelna and Karatkevich, 2021; Petty et al., 2022; Shahriar et al., 2020; Zhu et al., 2020).

Definition 1. A Petri net is a set $N=<P, T, F, H, M_0>$ where P is a finite non-empty set of symbols called positions; T is a finite non-empty set of symbols called transitions; incidence function

$$\left.\begin{aligned} F : P \times T \to \{0,1\} \\ H : T \times P \to \{0,1\} \end{aligned}\right\} \quad (1)$$

$M_0 : P \to \{0, 1, 2, 3, \ldots\}$ — initial marking.

A Petri net is represented by a labelled directed graph with a set of vertices $P \cup T$. Position vertices are represented by circles, transition vertices are represented by rectangles. An arc leads from a vertex-position p to a transition vertex t if and only if $F(t, p) = 1$. An arc leads from a transition vertex t to a vertex-position p if and only if $H(t, p) = 1$ The vertices-positions are marked with non-negative integers (position marking), which in the graphical representation of the network are placed inside the circle position or are depicted by the corresponding number of points (chips) in the circle position.

This definition applies to simple Petri nets. In the general case, a Petri net is a multigraph, and under triggering conditions it is required that the number of tokens in each input position p of transition t be greater than or equal to the number of arcs connecting p and t. Transition t then adds, after triggering, as many tokens to each of its input positions as there are arcs from t going into it.

More complete information about the structure and functioning of a Petri net, as well as about the applications of Petri nets for modelling the functioning of a wide range of systems, can be obtained in (David and Alla, 2010; Jensen, 2013; Reisig, 2016).

Let us note some basic facts for the subsequent presentation related to the properties of Petri nets (Balbo, 2007; Best and Wimmel, 2013; Cabasino et al., 2013; Desel and Reisig, 2015; Eshuis, 2013; Giua and Silva, 2018; Liu et al., 2012; Popova-Zeugmann, 2013; Reisig, 2013; Wang, 2007). There are algorithms that allow: for any Petri net to determine whether it is bounded; for any network transition, determine whether it is reachable in it; for any marking in the network, determine whether it is reachable in N, that is, whether it belongs to the set $R(N)$; for any network, establish whether the network is alive, and also prove that the equivalence problem for Petri nets is solvable.

The graph representation of Petri nets allows you to clearly depict their structural features and functioning dynamics. But to create means of presenting information, a form is needed that would allow automatic or automated transformation of networks, their construction from other networks. At the same time, the rules for transforming and constructing networks, as well as the networks themselves, must be "*well structured*." Next, we use the normal representation of a subclass of Petri nets (*regular nets*) and their generalization (*structured nets*), which is based on the algebra of regular nets. Regular and structured networks are used as models of real-life logical structures, and the algebra of these networks serves as the basis for developing a mechanism for defining and constructing a knowledge base.

4. Computational models of security system management

Operational management algorithms help in solving problems related to managing the security of a critical infrastructure facility. As a result of the operation of these algorithms, the security system must receive information about the consequences of existing or expected changes and deviations of the operating mode parameters of a critical infrastructure object beyond acceptable limits, as well as instructions on measures to prevent threats or ensure the required level of security of the protected object. Depending on the level of threats to a critical infrastructure facility, the security system may have different time to respond to a particular cyber incident. To do this, algorithms must be constructed in such a way that they can be simplified depending on the required reaction time. As such a means of describing algorithms that are customized depending on the problem situation, their model description is proposed (Balusamy et al., n.d.; Krishnan et al., 2013; Miehling et al., 2019), i.e., a description in the form of a set of parameters and their transformations.

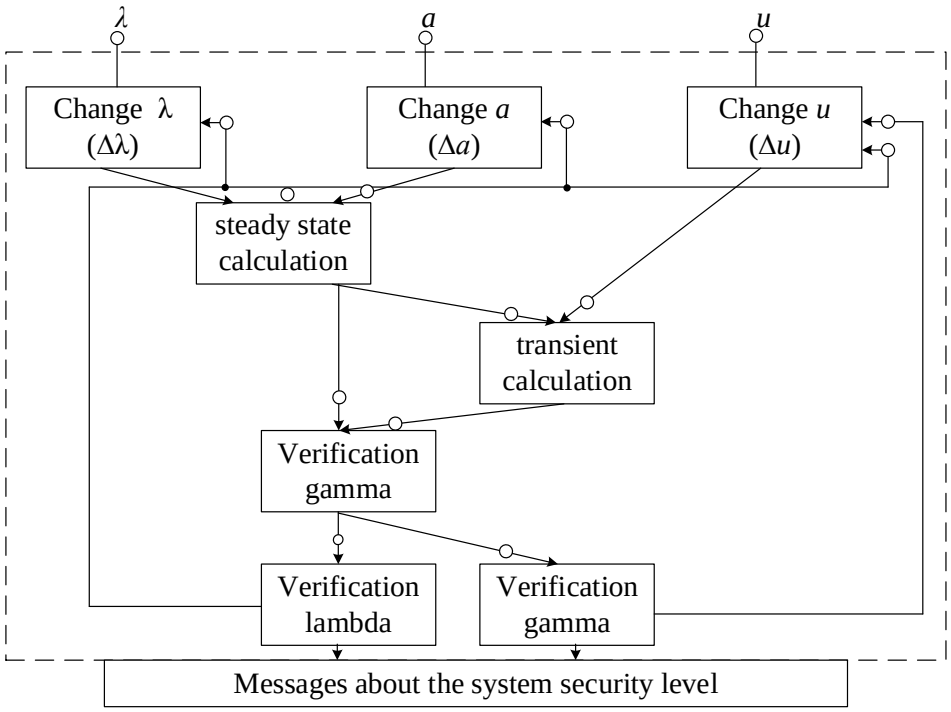

Figure 2. A graphical representation of a simplified model for quickly solving problems of ensuring the safety of critical infrastructure facilities

A graphical representation of a simplified model for quickly solving problems of ensuring the safety of critical infrastructure facilities is shown in Fig. 2. Assessing the level of safety of critical infrastructure objects can be considered as a search for an answer to the question of whether a given vector of mode parameters and the scheme of a critical infrastructure object or a vector of emergency situations γ(λ, a, u) belongs to

the area of the set of normal functioning in the space of parameters λ — Y_λ or γ — Y_γ. Here *a* is the disturbance vector; *u* is the vector of control decisions; Y_λ, Y_γ are vectors of boundary parameter values for normal and emergency situations.

To calculate the level of cybersecurity, it is necessary to create a set of models, using which additional information for the cybersecurity system is calculated.

The tasks of operational management of cybersecurity can be represented in the form of models on which one can organize calculations leading to obtaining a solution. Such models are called computational (Lockwood and Klein-Flügge, 2021; Mao et al., 2023; Reza Shaebani et al., 2020; Tump et al., 2024).

Definition 2. We will call *a simple computational model* a pair <*x*, *A*>, where *x* is a set of variables; *A* is a finite set of relations connecting these variables. Relations contain information about the mutual dependence of variables and are used to calculate the values of the latter. The concept of "relations" is specified in such a way that partial relations are defined by a set of variables and a set of operators whose inputs and outputs are variables from this set.

A graph whose vertices are the variables and relations of the model and whose edges connect the relations with their associated variables will be called *the logical diagram of the model*. A model is a structure of information whose elements may have uncertain values.

From the operators generated by the relations of the model, *computational processes* are created that change its state after each completion of the operator. Let's call the *state of the model* a pair $\langle W, \bar{W} \rangle$, where *W* is an arbitrary set of model elements, called a state diagram; $\bar{W}$ is the *value* of the set *W*. To determine the set of admissible computational processes on the model, we define *a control strategy* or simply control, which is a set of rules. Control can be specified by: predicates on the set of all computational processes of the model, grammars over alphabets consisting of operators, and an operator scheme of the program.

Computational models with control are seen as a means of representing solution-finding processes. Unlike algorithms, computational models with control are non-deterministic descriptions of processes (in the general case, the process may not be uniquely defined on the model by the initial data). But with a suitable choice of control, it is possible to limit the set of admissible computational processes so that the model, together with the control, represents the processes as uniquely as some algorithm.

Problem on a computational model we'll call it three $\langle U, \bar{U}, V \rangle$, where U and V are sets of model variables (*U*, *V* are the input and output of the model); $\bar{U}$ — value of the set U (initial data); <*U*, *V*> - task diagram. Computational process that transfers a model from its initial state $\left(W_0, \bar{W}_0\right)$ to the target $\left(W_g, \bar{W}_g\right)$, solves the problem if $W_0=U_0$, $V \subset W_g$. Meaning $\bar{V} \subset \bar{W}_g$ of the output of the problem will be called the *answer*.

A problem is *solvable on a model* under some control if there is an *effective composition* of procedures that solves it.

Each computational model allows one to define the tasks of operational management of system security in terms of variables and their values. All these problems can be divided into equivalence classes, consisting of problems with the same scheme (*U*, *V*). For each such class of problems, an algorithm with input *U* and output *V* is sought that solves problems of this class. If such an algorithm exists on the model, then it solves the

problem if and only if it is applicable to the initial data *U*. Problems on the model are solved in two stages: solution planning, i.e., compiling an algorithm that solves the problem; interpretation of the algorithm and obtaining the result.

We will represent many different types of cybersecurity management problems by a variety of models on which these problems can be solved. However, many of the tasks, and therefore the models, are interconnected, i.e., there is synchrony and asynchrony in their implementation.

To describe such a case, we introduce into the computational model and into any set of computational models the concept of control on model C. In this case, we write the computational model in the form P(G, C), where G = (x, A) determines its information structure; C is the control structure or all possible chains of operators that can make up the computational process of solving the problem. To represent G we will use simple computational models.

We consider the computational model in the form of a bipartite graph consisting of variables and operators connecting them. When constructing a multi-model representation for cybersecurity management problems, it is first necessary to construct the information structure of each of the models, and then only the control structure.

Let's represent the control structure in the form of a Petri net. To do this, we introduce a number of definitions.

Definition 3. An *S*-net is a Petri net in which each event can appear only once and the launch of a transition does not entail a decrease in the markings in its input positions.

Definition 4. The control structure *C* for the computational model is the S-net, which is obtained from the model by unambiguously replacing all operators with transitions and variables with positions.

To fully define the model, it is necessary to select the initial marking, i.e., set the initial state of the control structure. To do this, points are assigned to some positions, i.e., they are placed in all positions corresponding to the input variables of the model. The problem can be solved on the model when all positions corresponding to the output variables of the problem are marked.

Let us introduce definitions characterizing the control structure of models.

Definition 5. The state diagram of an *S*-Petri net is the bipartite graph $G(Q, \Phi)$, where the vertices $q_i \in Q$ there will be S-net states and arcs $\varphi_i \in \Phi$ — events (operators). The state of the *S*-network will be uniquely determined by the logical marking vector $v = (v_1, v_2, ..., v_n)$, where n is the number of positions;

$$v_i = \begin{cases} 1, & \text{if } m_i \geq 1; \\ 0, & \text{if } m_i = 0 \end{cases} \tag{2}$$

t_i —number of markers at position p_i.

Arc $\varphi_i \in \Phi$ comes from state $q_k \in Q$ and comes to state $q_l \in Q$, if event φ_l may appear in the q_k state, as a result of which the system goes into q_l.

Among the states there is an initial one — $q_0 \in Q$, uniquely determined by the initial marking, and the final one — q_F, in which all positions are numbered.

Definition 6. A computational process on a model with control will be called a path: on a state diagram, starting with state q_0 and ending with q_F.

Each control structure C generates a certain set of computational processes that are uniquely determined by the state diagram. Considering many computational models of operational dispatch control problems, the control structure of each of which is represented by a Petri net, we noticed that for their representation they require the possibility of information exchanges, i.e., they require the organization of an

asynchronous computing process.

Let us highlight a sequential computational process in which a certain chain of operators (transitions) on a Petri net is executed only sequentially. To organize asynchronous processes, we divide all positions of Petri nets into two sets: synchronization positions, which are common to several models, with the help of which the interaction of processes on the models is synchronized; positions of internal states of models, each having one input and one output arc. Each such position belongs to only one process, one of the state positions of which is marked, thereby expressing the current state of the process.

If we now transform each of the Petri nets so that only unidirectional arcs emanate from the positions of internal states, and bidirectional arcs from the synchronization positions, then we can obtain a composition of models on which asynchronous processes are solvable. In Fig. 3, to clarify the above, three asynchronous processes are presented, each implemented by its own Petri net.

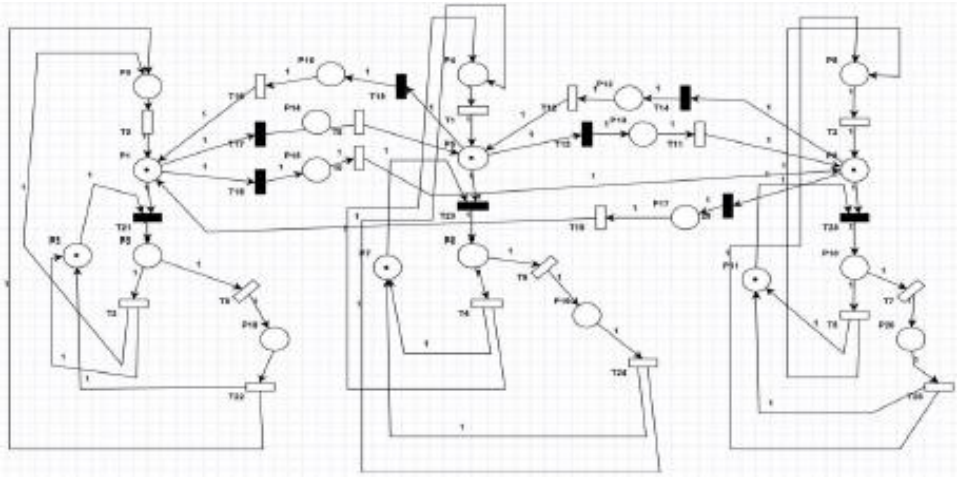

Figure 3. Parallel topology modelling using Petri Nets
[Source: Ameen A. Evaluation of the computer networks security level based on petri nets & a set of parameters //Agricultural Sciences. – C. 82.]

5. Presentation of expert models for managing the security of a socio-cyberphysical system

Analysis of the processes of developing control decisions in the cybersecurity system showed that all reasoning is based on the knowledge accumulated in the cognitive model. It turned out to be possible to represent this knowledge by a variety of cause-and-effect relationships that can be described in the concepts and relationships of the information-model basis. The more experienced a person is, the richer his set of relationships and the wider and more complete his information-model basis. Structuring relationships depending on the goals and objectives of management allows a person to solve the problem of enumeration when searching for solutions using classification methods. The reasoning used by humans when searching for solutions in the class of

security systems corresponds to the scheme "if *A* and from *A* follows *B*, then *B*," which corresponds to the "modus ponens" rule in logic (Akama and Nantajeewarawat, 2022).

All this makes it possible to create tools that make it possible to describe expert mental models of the functioning of a security system in the language of first-order predicate logic (Console et al., 2021; Delgrande and Rantsoudis, 2020; Fitting and Mendelsohn, 2023).

An expert model is a set of statements that reflects both the subjective knowledge of experts about the environment, object and management processes, and the objective laws of the subject area of management. An example of one of the statements of the model is the statement that the shutdown of some network nodes entails an increase in load in neighbouring network nodes.

In the security system, to describe such expert models, logical-algebraic models (LAM) are used (Emelyanov, 2018; Levin, 218 C.E.; Menshikh, 2015; Pathuddin et al., 2022), which make it possible to describe the entire set of expert statements. The sequence of actions of an expert - system analyst with this description is shown in Fig. 4.

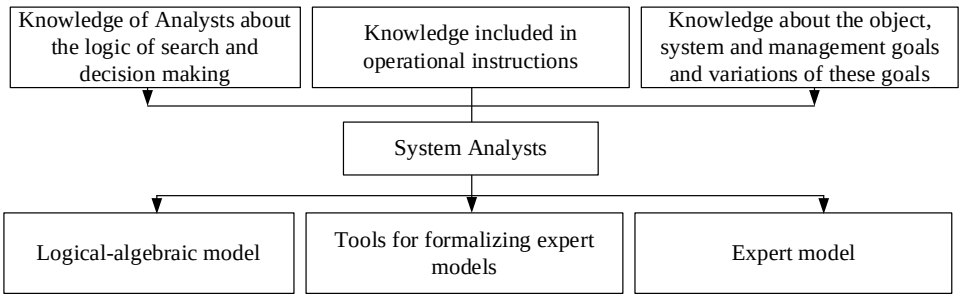

Figure 4. The sequence of actions of an expert - system analyst

Definition 7. The logical-algebraic model is defined by the expression

$$LAM = \left\langle T_L, H_L, \Psi_L, \Theta_L, \rho\left(\Psi_L, \Theta_L\right)\right\rangle \tag{3}$$

where T_L are model terms; N_L - rules for constructing correct expressions in LAM (LAM syntax); Ψ_L — axioms describing the laws of the control domain (truly interpreted, correctly constructed expressions); Θ_L— inference rules; $\rho(\Psi_L, \Theta_L)$ is the area of interpretation.

Given the initial interpretation in the form of axioms, the rules allow us to derive all true expressions in LAM (Ψ_L and Θ_L define the semantics of LAM). These models, unlike formal systems, allow contractual interpretation, that is, they allow changes in the semantics of LAM. For this, two possibilities are used: changing the system of axioms — $\rho(\Psi_L)$ and changing the inference rules — $\rho(\Theta_L)$.

The truth of some expressions λ_i in LAM is established in two ways: either λ_i is found as a result of logical inference in LAM from the existing system of axioms according to the rules of inference, or the truth of λ_i is obtained by interpreting it.

An axiomatic component of the LAM model or any subset of its elements determined by partitioning Ψ_L into equivalence classes, corresponds to the concept of "an expert model of decision-making in a given class of problem situations." In other words, expert

models are sets of statements about the methods and techniques of searching for control solutions in a certain class of situations and are described in the language of first-order predicate logic. The choice of such a description language is not accidental, but is determined by the presence of a general procedure for finding solutions called the principle of resolutions (Heil, 2021).

Logical-algebraic models with control can be considered as a means of representing the processes of searching for control recommendations to security officials. These models, unlike algorithms, are non-deterministic descriptions of logical processes: in the general case, the process is not uniquely defined in the model. But a suitable choice of control can limit the set of admissible logical processes.

Definition 8. We will call *a problem on a logical-algebraic model* the expression $<\Psi_L,\ \Psi,\ t,\ \rho(\Psi_L,\Theta_L)>$, where Ψ_L — all model statements; Ψ — a statement whose deducibility must be proven; t is the set of model terms; $\rho(\Psi_L,\Theta_L)$ — interpretation of the model statements.

The problem scheme is determined by the expression $<\Psi_L,\ \Psi>$. Logical process proving derivability Ψ from Ψ_L, solves the problem if there is an interpretation area $\rho(\Psi_L,\ \Theta_L)$, in which it brings the process to nil. The value of the term, the substitution of which brings the problem to nil, is called *the answer*. A problem is *solvable on a model* under some control if there is *a domain of interpretation* of the model and a *logical process* admissible in it that allows one to find the answer.

Expert models of processes for managing operating modes of critical infrastructure objects define decision-making tasks in operational dispatch control in terms of logical expressions and terms. All these problems are divided into equivalence classes, consisting of problems with the same scheme, each of which is associated with a certain logical-algebraic model. For any class of problems, you can search for an algorithm with inputs $\Psi_L,\ \Psi$ and outputs nil, solving problems of this class. If such an algorithm exists, then it solves the problem, if and only if it is applicable to the terms of some logical-algebraic model.

From the above it follows that all problems on expert models are solved in two stages - planning (searching for candidates for applying the inference rule) and interpretation (application of the inference rule). If there is no truth interpretation, new candidates for applying the rules of inference are sought and the process is repeated.

Analysing the process of logical inference in an expert model, one can see that two sets are involved in this process: statements of the expert model and inference rules. These sets do not intersect, which makes it possible to describe the model and logical inference processes on it using the apparatus of bipartite graphs (Jebali et al., 2020; Yang et al., 2007). At the same time, logical inference control is built on such a model.

The set of inference rules in the system, in turn, is divided into two subsets: the first is characterized by the absence of conditions for the application of each of the rules; for the second, each rule is assigned a condition for its application. The examples considered were related to the rules of the second subset. In fact, the conditions for their use contained information about the search pattern, for which there was a computational interpretation procedure. Typically, such a set of rules arises in problems for which the inference processes are well studied and structured.

However, tasks arise in the system for which the interpretation can be specified by a computational process, which is a sequence of computational procedures. The essential thing is that this sequence must be determined in the process of logical inference. For such problems, the inference rules do not contain application (interpretation) conditions.

They work by pattern searching, not in search procedures, but in a set of model statements.

The presence of such rules in the system makes it possible to build a flexible computational process for searching of control recommendations and control commands.

6. Presentation of dialogue models to describe interaction with a cybersecurity system

One of the requirements for cybersecurity decision-making systems is to provide users with means of interactive communication. At the same time, since the system has several classes of users at the input, a description apparatus is developed, built on the basis of dialogue models. The basis of such a device is the assumption that, when interacting with the system, the user can perform a very specific set of actions and expects a very specific reaction to them. Operations performed by the user are not performed arbitrarily, but in a certain relationship with each other, depending on the current state of the system, on what stage of interaction is currently being performed. The description of the structure of the dialogue, its script (human-machine conversations) should clearly express the relationship between the interaction of man and machine. In this case, one can abstract from the specific content of these actions and focus on the structural aspects of this interaction. The explicitly defined relationship between the actions of the user and the machine, described by natural language phrases, will be called the structure of the *dialogue script* (Chen et al., 2021).

Such structures are not of a "prescriptive" nature, characteristic of algorithms, but of a "limiting" nature. This is determined by the fact that the dialogue subsystem also includes a significant user component with its inherent non-algorithmizable behaviour. The "restrictive" nature lies in the fact that it does not set strict sequences of operations, but only establishes certain rules for their implementation, limiting the user from committing unacceptable actions in a particular context of his interaction with the cybersecurity system. On the other hand, this does not prevent the machine structure from being given instructions of an algorithmic nature, since many restrictions can narrow the set of acceptable alternatives strictly to one at each step of the dialogue process.

The user, in his communication with the cybersecurity system, can use operations of a different nature - from trivial ("Type the word TRAFFIC") to informal ("What potential attacks are possible within 24 hours?"). At the same time, obtaining an answer to the last question is the result of solving the complex problem of developing an answer. From the point of view of describing the structure of the dialogue, what is important is not the method of their execution, but the place of such operations between the user and the cybersecurity system, and the paths of information exchange between them.

As studies have shown (Kishita et al., 2020; Ranganathan et al., 2023), the structure of a dialogue script can be represented as a bipartite graph containing vertices of two types: conditions and operators (Ouyang et al., 2020). Conditions are introduced into the structure in order to use them to determine the possibility of executing certain operators at different times. An arc going from a condition to a statement means that the statement can only be executed if the given condition is met. An arc going from an operator to a condition means that after each execution of this operator, the corresponding condition can be satisfied. With each condition we associate a variable non-negative integer characterizing the multiplicity of its execution. When an operator is triggered, all

conditions emanating from it are considered not just to have occurred, but to have occurred again. This is marked by an increase in their multiplicity by one. In addition, we will assume that for all conditions that are predecessors of the triggered operator, the multiplicity decreases by one. A mandatory condition for the operator to be triggered is that all predecessor conditions have a multiplicity other than zero. The considered structure can be formally described using a Petri net.

Definition 9. We define the structure of the dialogue script by the expression

$$D=<E, S, F, B, M_0>, \quad (4)$$

where E, S is a finite set of conditions and operators; F, B — set of arcs that define the incidence of conditions;

$$F: E \times S \to \{0, 1\};$$
$$B: S \times E \to \{0, 1\}, \quad (5)$$

and operators; M_0: $E \to \{0, 1, 2, 3, 4, \ldots\}$ — initial marking of event multiplicities.

The operators that make up the set S are some transformers. Operator $s_i \in S$ is ready to operate if the trigger condition is met: $(\forall e)M(e)\text{-}F(e,s_i)\geq 0$, where M is the current multiplicity marking. In other words, the s_i operator can work if all input events have a multiplicity other than zero. Operator $s_i \in S$ is triggered when performing the associated transformation of the current state, as well as when changing the current marking of multiplicities of events: $(\forall e)M'(e) = M(e) - F(e, s_i) + B(s_i, e)$.

The functioning of such a structure consists in the sequential operation of operators, entailing a corresponding sequential change in the marking of events and the current state of the process.

Networks of the described type can serve as formal models of the activities that are carried out by the user and the computer system in their joint dialogue. Dialogue is a single network in which two subnetworks are distinguished, intersecting only according to a variety of conditions. One of the subnets is a model of the security system's activity, the other is a model of the user's activity in dialogue with the cybersecurity system. The activation of an operator in a network during its operation symbolizes the performance of one or another action by one of the interacting parties. The functioning of these two subnetworks is naturally linked to each other through common conditions. The alternation of information exchange operators between the operator and the cybersecurity system, which is prescribed by the complete network, defines a formal dialogue model described by the expression $D(G_D, C_D)$. Here G_D, C_D define information and control structures or all possible chains of operators that can make up a computational process.

Definition 10. The state diagram of a D-Petri net (dialogue Petri net) is a bipartite graph $G(E, S)$, where the vertices $e_i \in E$ will correspond to the state of the D-network, arcs $s_i \in S$ — events (operators). The state of the D-network will be uniquely determined by a logical vector of markings of the form $v = (v_1, v_2, v_3, \ldots, v_n)$, where n is the number of positions;

$$v_i = \begin{cases} 1, \text{ if } m_i \geq 1 \\ 0, \text{ if } m_i = 0 \end{cases}, \quad (6)$$

m_i — number of markers in position s_i. Arc $s_i \in S$ comes from state $e_k \in E$ to the state $e_l \in E$, if the event s_i can appear in the state e_k, then the system goes to e_i.

Among the states we will distinguish the initial $e_0 \in E$, uniquely determined by the initial marking, and the final $e_F \in E$, in which all positions are numbered.

Definition 11. A computational (search) process on the model of dialogue with control will be called a path on the state diagram, starting with state e_0 and ending with e_F.

Each control structure of the dialogue generates a certain set of processes for searching for an answer, uniquely determined by the state diagram. To use dialogue models in a security system, the state diagram is specified in advance and is determined by the scenario and class of dialogue situations.

7. Presentation of problem situations of socio-cyberphysical systems

One of the main tasks that arises when creating cybersecurity systems is the task of forming classes of states of critical infrastructure objects, for which it is necessary to develop control decisions to ensure cyber protection. For critical infrastructure objects, the number of possible states significantly exceeds the number of possible control actions (control decisions).

The task of identifying problem situations and their classification is formulated as follows. There is a certain set of parameters $\{X\}$ that characterize the state of the critical infrastructure, and it is known that the classes of problem situations $K_1, K_2, \ldots, K_n$ are subsets of the set $\{X\}$. Given a description of the set $\{X\}$, information about classes of problem situations and a description of a certain set of parameters $\{x^*\} \subset \{X\}$. It is required to establish to which of the classes $K_1, K_2, \ldots, K_n$ the set of parameters $\{x^*\}$ belongs.

This general task in the system is considered as a combination of two tasks:

1. Presentation of a set of parameters characterizing the state of the critical infrastructure, and the formation of a decision rule designed to calculate the membership of all possible sets of parameters to each of the K_j classes. This task is called *the problem representation task*. It is solved by experts — system analysts.
2. Automatic determination of the membership of all possible sets of parameters specified on $\{X\}$ to classes of problem situations using a decision rule.

The first problem is solved when designing a system or when adapting it to the control domain and is posed as a generalization problem on the set $\{X\}$. The result of this generalization is generalized representations of G classes of problem situations K_i. The second problem is solved at the stage of system operation in automatic mode.

To present problem situations, it is necessary to: 1) describe the set $\{X\}$; 2) on the set, define (describe) equivalence classes corresponding to the classes of problem situations M_S; 3) describe the rules that allow you to determine whether a certain set of parameters belongs to the class K_i.

All procedures for presenting problem situations in the system are expert in the sense that they are built by system analysts in an interactive mode of training and system design.

8. Presentation of set-theoretic models to describe the states of a socio-cyberphysical system

Any state of critical infrastructure is always characterized by many parameters, the

values of which can vary within certain limits. This set of parameters is named, and on it you can build any relationships that are useful from the point of view of solving problems of managing critical infrastructure objects, in particular, relationships that describe the network topology, characterize the states of nodes, gateways, data transmission channels, etc. All relationships in the system are specified in a table and are determined by the name of the relationship and the list of attributes (named parameters) on which they are defined.

Such relationships exist in the database for all concepts characterizing the field of critical infrastructure management and are of a conditionally constant nature, that is, they do not change descriptively over a long period of time. At the same time, relationships arise and disappear that characterize the processes of recognizing situations, searching and making decisions. For example, in the process of searching for a solution, a need arises to find out from the network topology which set of network nodes $\{Y\}$ is connected to a specific node B. This need is realized by specifying a dynamically (situationally) emerging relationship (connection) of node B. The appearance of dynamic relationships is associated with the presence in security system for presenting information not only at the data level, but also at the knowledge level (Catal et al., 2023; Zwilling et al., 2022). It is at the level of knowledge in the process of searching for solutions that dynamic relationships arise, for which it is necessary to establish their current truth by interpreting them on the basis of data.

All relationships are built on attributes defined in the database. The existence of conditionally permanent relationships in the database from the point of view of their adequacy to the state of the power system is supported by Database Management System (DBMS) tools represented by the system administrator. Dynamic relationships exist when there is a procedure that allows you to determine the presence or absence of the possibility of establishing such a relationship on the attributes of the database.

The relationships considered are based on a set-theoretic relational data model and are supported by standard DBMSs (Gao et al., 2021).

9. Language for defining knowledge as a means of constructing a semiotic model of information representation

Analysing the considered set of models, one can notice that the language of a formal system can serve both to describe the theory of management activity in operational decision making (knowledge base) and to describe the information on which this theory is based (database).

Let us now consider the possibility of expanding the formal system from the point of view of using it to build automated design tools for models and knowledge bases and databases. Let's try to find the answer to the question "What needs to be added to the formal system in order to be able not only to describe its components, but also to modify it in order to adjust it to the management area?" To do this, it is necessary to introduce rules into the formal model: D, allowing you to change the alphabet of the model T; Ω, allowing you to change the rules of H, i.e., the syntax of the model; χ, allowing you to change the set Ψ by modification, i.e. introducing or removing laws, facts, changed consequences, relations in the subject area of management (the introduction of this rule expands Ψ not only by applying rules to it Θ (obtaining consequences from the initial premises), but also allows, in the design, training and adaptation modes, both the

addition of new axioms and the exclusion of outdated ones. Rules Θ in this case, they can be treated as currently contained in Ψ expressions and missing ones); Ξ, defining change rules Θ. The problem solved using these rules is called the *adaptation problem*.

As a result of such additions, the expression of the formal system takes the form

$$A = < T, H, \Psi, \Theta, D, \Omega, \chi, \Xi > \quad (7)$$

and is the formal definition of the semiotic model (Smith, 2023; Yevseiev, Tolkachov, et al., 2023).

The semiotic model has two components: inductive and deductive. The main thing from the point of view of creating means of presenting and describing information is the inductive component. With its help, system users (experts) can identify problem situations, describe the logic for solving operational decision-making problems, and create algorithms.

As a unified means of describing and presenting information about the theory of control activity in security systems, a knowledge definition language (KDL) is used, the main purpose of which is to describe all attributes of the control area, assign conditionally constant and dynamic relationships to them and construct axioms of the theory of control activity during operational decision making.

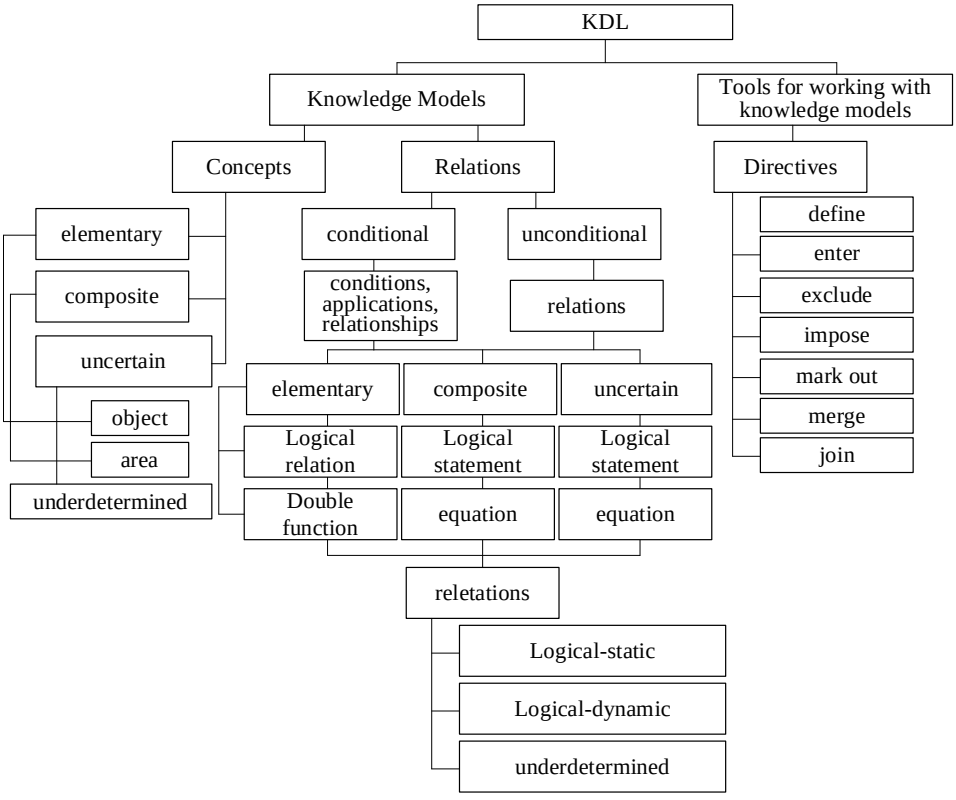

Figure 5. The structure of language elements

Formally, a language is defined by an expression of the form

$$L_{KDL} = <L_T, L_N, L_P> \quad (8)$$

where L_T, L_N — many terminal and non-terminal symbols; L_P —a set of rules that include language directives. All terminal and non-terminal elements of the language have their own specific types.

The structure of language elements is shown in Fig. 5. Concepts and relationships allow you to describe any of the models included in the knowledge base. Directives are means of user interaction with the system.

Concepts are defined in a wide range - from elementary (for example, *channel* (*x*; *y*)), composite (local organization network) to undefined. divided, i.e. not making sense in a specific area of application. The type of concepts determines the scope of their interpretation, therefore, for any undefined concept, the security system will require the expert to redefine or exclude it.

Relationships are divided into conditional and unconditional. The first ones are always of a specific nature, valid only for a given field of application. The specificity is determined by the condition of application of this relation. Unconditional relations have the nature of laws that are always valid in theory, in other words, they have a universally valid character.

All relations and conditions of application (also considered as relations with their own area of interpretation) can be elementary (for example, the connection relation on the network), composite and indefinite. In this case, the conditions of application can only have logical-static or logical -dynamic type. Other relations may also have a computational type. The difference between these types (excluding the underdetermined one) is that the computational type gives, after interpretation, only the numeric values of the parameters. Other types give the logical values of the truth or falsity of the interpretation in combination with the numeric values parameters on which this truth or falsity is established. With relations of an underdetermined type, the biohazard system acts as with concepts, i.e., it requires either their further definition (setting the area of interpretation) or exclusion.

The logical-static type of relationship implies the presence in the database of a table corresponding to each of the defined relationships of this type. The logical-dynamic type of relationship implies the presence in the knowledge base of a procedure with a specific area of interpretation in the database for it. Each relation of this type in the knowledge base must have its own procedure, which can at any time determine many parameters that ensure the truth or falsity of this type of relation.

The expert describes each of the newly introduced models or any change in terms of concepts and relationships in nuclear weapons. According to the relevant directives, these descriptions enter the input of the KDL processor, which processes the task descriptions, selects a set of elements and operators, and builds a graph of the information model. The resulting graph is the input information for the static scheduler; it builds a control graph of the knowledge base, called the logical model of the knowledge base. Thus, having built a knowledge base in the form of sets of models and data and defined the KDL as a means of description, it is possible to develop means of finding a solution using the considered models.

10. Conclusion

1. A classification system for security system design models is proposed depending on the areas of application, namely computational, expert, interactive and set-theoretic models. This approach allows us to jointly consider models of cybersecurity systems of various levels of formalization, which is necessary in the process of designing cybersecurity systems. As a result of applying this approach, taking into account various platforms for representing a socio-cyberphysical system, it becomes possible to form a unified model basis for the security system of a socio-cyberphysical system.

2. A structural and logical diagram of a socio-cyberphysical system is proposed, which clearly reflects the multi-circuit nature of the cybersecurity system of socio-cyberphysical systems and includes a model of a cybersystem platform, a model of a cyberspace platform and a model of a social network platform. The dynamics of threats to the contours and platforms of the socio-cyberphysical system are determined using the Lotka-Volterra model.

3. A method is proposed for formalizing each of the specified classes of models and a method for representing a cybersecurity system in memory (computational models, expert models, interactive models and set-theoretic models). It is proposed to use the Petri net apparatus as a universal means of representing models in cybersecurity systems, on the one hand, and as a means of integrating the considered models into a single whole, on the other hand.

4. A requirement has been formed for the need to form a set of computational models to calculate the level of cybersecurity. It is shown that the problems of operational management of cybersecurity can be represented in the form of computational models on which it is possible to organize calculations leading to obtaining a solution.

5. It is proposed to use a class of expert models based on the language of first-order predicate logic to describe mental models of the functioning of a security system, which makes them a universal tool for reflecting both the subjective knowledge of experts about the environment, object and management processes, and the objective laws of the subject area of management. In the cybersecurity system, to describe expert models, it is proposed to use logical-algebraic models that make it possible to describe the entire set of expert statements.

6. It is recommended to describe the structure of the dialogue, its scenario expressing the relationship between the interaction of the user and the cybersecurity system, to abstract from the specific content of these actions and focus on the structural aspects of this interaction. It is proposed to implement these aspects in the form of dialogue models. To do this, it is necessary to preset the state diagram of the system, which is determined by the scenario and class of dialogue situations.

7. The classification of knowledge representation languages is considered, in terms of concepts and relationships of which each of the newly introduced models or any change to them can be represented.

References

Akama, K., Nantajeewarawat, E. (2022). A foundation of logical problem solving . *International Journal of Innovative Computing, Information and Control*, ***18***(5), 1559–1570.

Baez, J. C., Master, J. (2020). Open Petri nets. *Mathematical Structures in Computer Science*, ***30***(3), 314–341. https://doi.org/10.1017/S0960129520000043

Balbo, G. (2007). Introduction to Generalized Stochastic Petri Nets. In *Formal Methods for Performance Evaluation* (pp. 83–131). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72522-0_3

Balusamy, S., Dudin, A. N., Graña, M., Mohideen, A. K., Sreelaja, N. K., Malar, B. (n.d.). *Cyber Security and Computational Models*.

Best, E., Wimmel, H. (2013). *Structure Theory of Petri Nets* (pp. 162–224). https://doi.org/10.1007/978-3-642-38143-0_5

Cabasino, M. P., Giua, A., Seatzu, C. (2013). *Introduction to Petri Nets* (pp. 191–211). https://doi.org/10.1007/978-1-4471-4276-8_10

Cantrell, W. A. (2021). *Verification and validation methods for extended Petri nets modeling cyberattacks : a dissertation* [University of Alabama in Huntsville]. https://scholar.google.com/citations?view_op=view_citation&hl=en&user=tjfW1IkAAAAJ&citation_for_view=tjfW1IkAAAAJ:WF5omc3nYNoC

Carielli, S., Eble, M., Hirsch, F., Rudina, E., Zahavi, R. (2020). *IoT Security Maturity Model: Practitioner's Guide*. https://www.iiconsortium.org/pdf/IoT_SMM_Practitioner_Guide_2020-05-05.pdf

Carielli, S., Rudina, E., Soroush, H., Zahavi, R. (2018). *IoT Security Maturity Model: Description and Intended Use*. http://www.iiconsortium.org/pdf/SMM_Description_and_Intended_Use_2018-04-09.pdf

Cassandras C.G., Lafortune, S. (2021). Petri Nets. In *Introduction to Discrete Event Systems* (pp. 259–302). Springer International Publishing. https://doi.org/10.1007/978-3-030-72274-6_4

Catal, C., Ozcan, A., Donmez, E., Kasif, A. (2023). Analysis of cyber security knowledge gaps based on cyber security body of knowledge. *Education and Information Technologies*, ***28***(2), 1809–1831. https://doi.org/10.1007/s10639-022-11261-8

Chen, Y., Liu, Y., Chen, L., Zhang, Y. (2021). *DialSumm: A Real-Life Scenario Dialogue Summarization Dataset*.

Console, M., Guagliardo, P., Libkin, L. (2021). Propositional and Predicate Logics of Incomplete Information. *International Conference on Principles of Knowledge Representation and Reasoning*. https://api.semanticscholar.org/CorpusID:52839075

David, R., Alla, H. (2010). *Discrete, Continuous, and Hybrid Petri Nets*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-10669-9

Delgrande, J., Rantsoudis, C. (2020). A preference-based approach for representing defaults in first-order logic. *Proceedings of the 18th International Workshop on Non-Monotonic Reasoning, NMR*.

Desel, J., Reisig, W. (2015). The concepts of Petri nets. *Software & Systems Modeling*, *14*(2), 669–683. https://doi.org/10.1007/s10270-014-0423-3

Emelyanov, A. S. (2018). Theory of decision-making: the logical-mathematical aspects. *International Journal of Advanced Studies*, ***8***(2), 22–39. https://doi.org/10.12731/2227-930X-2018-2-22-39

Eshuis, R. (2013). Statechartable Petri nets. *Formal Aspects of Computing*, ***25***(5), 659–681. https://doi.org/10.1007/s00165-011-0204-5

Fitting, M., Mendelsohn, R. L. (2023). *First-order modal logic* (Vol. 480).

Gao, C., Zhang, X., Liu, H. (2021). Data and knowledge-driven named entity recognition for cyber security. *Cybersecurity*, ***4***(1), 9. https://doi.org/10.1186/s42400-021-00072-y

Giua, A., Silva, M. (2018). Petri nets and Automatic Control: A historical perspective. *Annual Reviews in Control*, ***45***, 223–239. https://doi.org/10.1016/j.arcontrol.2018.04.006

Grobelna, I., Karatkevich, A. (2021). Challenges in Application of Petri Nets in Manufacturing Systems. *Electronics*, ***10***(18), 2305. https://doi.org/10.3390/electronics10182305

Heil, J. (2021). *First-order logic: A concise introduction*. Hackett Publishing Company, Inc.

Hofman, J. (2017). *A decision support framework for cybersecurity management*. https://api.semanticscholar.org/CorpusID:198991426

Hryshchuk, R. (2016). *Fundamentals of cyber security: Monograph*.

Hryshchuk R., Yevseiev S. (2016). The synergetic approach for providing bank information security: the problem formulation. *Ukrainian Scientific Journal of Information Security*, ***22***(1). https://doi.org/10.18372/2225-5036.22.10456

Jebali, A., Sassi, S., Jemai, A. (2020). Inference Control in Distributed Environment: A Comparison Study. In *Risks and Security of Internet and Systems: 14th International Conference* (pp. 69–83). Springer International Publishing. https://doi.org/10.1007/978-3-030-41568-6_5

Jensen, K. (2013). *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Volume 1*. Springer Science & Business Media.

Kishita, Y., Mizuno, Y., Fukushige, S., Umeda, Y. (2020). Scenario structuring methodology for computer-aided scenario design: An application to envisioning sustainable futures. *Technological Forecasting and Social Change*, *160*, 120207. https://doi.org/10.1016/j.techfore.2020.120207

Krishnan, G. S. S., Anitha, R., Lekshmi, R. S., Kumar, M. S., Bonato, A., Graña, M. (2013). *Computational Intelligence, Cyber Security and Computational Models*. Springer Science & Business Media.

Levin, V. (218 C.E.). Logico-mathematical methods and their applications . *Control, Communication and Security Systems*, ***2***, 213–244.

Liu, G., Jiang, C., Zhou, M., Xiong, P. (2012). Interactive petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, ***43***(2), 291–302.

Lockwood, P. L., Klein-Flügge, M. C. (2021). Computational modelling of social cognition and behaviour—a reinforcement learning primer. *Social Cognitive and Affective Neuroscience*, 761–771. https://doi.org/10.1093/scan/nsaa040

Mao, R., Li, X., He, K., Ge, M., Cambria, E. (2023). MetaPro Online: A Computational Metaphor Processing Online System. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, 127–135. https://doi.org/10.18653/v1/2023.acl-demo.12

Menshikh, A. (2015). Logico-mathematical models of the choice of security measures . *Actual Problems of the Activity of the Correctional Centre Units*, 283–285.

Miehling, E., Rasouli, M., Teneketzis, D. (2019). *Control-Theoretic Approaches to Cyber-Security* (pp. 12–28). https://doi.org/10.1007/978-3-030-30719-6_2

Milov, O. (2019). Adaptive decision support systems for cyber security. *Advanced Information Systems*, ***3***(1). https://doi.org/10.20998/2522-9052.2019.1.22

Ouyang, S., Zhang, Z., Zhao, H. (2020). *Dialogue Graph Modeling for Conversational Machine Reading*. arXiv:2012.14827

Pathuddin, P.,Linawati, L., Mubarik, M., Fadlun, F., Anggraini. (2022). High logical-mathematical intelligence learner's problem-solving performance on integer operation problem. *AIP Conference Proceedings*, 020048. https://doi.org/10.1063/5.0096068

Petty, M. D., Whitaker, T. S., Bearss, E. M., Bland, J. A., Cantrell, W. A., Colvett, C. D., Maxwell, K. P. (2022). Modeling cyberattacks with extended Petri nets. *Proceedings of the ACM Southeast Conference*, 67–73. https://doi.org/10.1145/3476883.3520209

Popova-Zeugmann, L. (2013). Time Petri Nets. In *Time and Petri Nets* (pp. 31–137). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-41115-1_3

Ranganathan, A., Tamminaina, S. G., Raina, G. (2023). A Study Of Dialog Summarization Across Datasets And Domains. *Proceedings of the 2023 7th International Conference on Natural Language Processing and Information Retrieval*, 196–202. https://doi.org/10.1145/3639233.3639245

Reisig, W. (2013). *Elements of Distributed Algorithms: Modeling and Analysis with Petri Nets*. Springer Science & Business Media.

Reisig, W. (2016). *Understanding petri nets*. Springer.

Reza Shaebani, M., Wysocki, A., Winkler, R., Gompper, G., Rieger, H. (2020). *Computational models for active matter*.

Shahriar, Md. A., Bappy, F. H., Fakhrul Hossain, A. K. M., Saikat, D. D., Ferdous, M. S., Chowdhury, M. J. M., Bhuiyan, M. Z. A. (2020). Modelling Attacks in Blockchain Systems using Petri Nets. *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 1069–1078. https://doi.org/10.1109/TrustCom50675.2020.00142

Shmatko, O., Balakireva, S., Vlasov, A., Zagorodna, N., Korol, O., Milov, O., Petrov, O., Pohasii, S., Rzayev, K., Khvostenko, V. (2020). Development of methodological foundations for designing a classifier of threats to cyberphysical systems. *Eastern-European Journal of Enterprise Technologies*, *3*(9–105). https://doi.org/10.15587/1729-4061.2020.205702

Smith, C. B. (2023). The Semantic Attack Surface: A Systems-Dynamic Model of Narrative in Cyberspace. *IEEE Transactions on Technology and Society*, *4*(2), 146–157. https://doi.org/10.1109/TTS.2022.3210782

Tump, A. N., Deffner, D., Pleskac, T. J., Romanczuk, P., Kurvers, R. H. J. M. (2024). A Cognitive Computational Approach to Social and Collective Decision-Making. *Perspectives on Psychological Science*, ***19***(2), 538–551. https://doi.org/10.1177/17456916231186964

Wang, J. (2007). Petri Nets for Dynamic Event-Driven System Modeling. In P. A. Fishwick (Ed.), *Handbook of Dynamic System Modeling*. Chapman and Hall/CRC. https://doi.org/10.1201/9781420010855.CH24

Yang, Y., Li, Y., Deng, R. H. (2007). New Paradigm of Inference Control with Trusted Computing. *Data and Applications Security* , 243–258.

Yevseiev, S., Murr, P., Milevskyi, S., Korol, O., Melnyk, M. (2023). Development of a Sociocyberphysical Systems Cyber Threats Classifier. *2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 1–7. https://doi.org/10.1109/ISMSIT58785.2023.10304895

Yevseiev, S., Pohasii, S., Milevskyi, S., Milov, O., Melenti, Y., Grod, I., Berestov, D., Fedorenko, R., Kurchenko, O. (2021). Development Of A Method For Assessing The Security Of Cyber-Physical Systems Based On The Lotkavolterra Model. *Eastern-European Journal of Enterprise Technologies*, *5*(9–113). https://doi.org/10.15587/1729-4061.2021.241638

Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O., Korol, O., Milevskyi, S., Pohasii, S., Tkachov, A., Shmatko, O., Melenti, Y., Sievierinov, O., Ostapov, S., Gavrilova, A., Tsyhanenko, O., Herasimov, S., Nyemkova, E., Tomashevsky, B., Grod, I., Opirskyy, I., … Florov, S. (2021). Synergy of building cybersecurity systems. In *Synergy of building cybersecurity systems*. https://doi.org/10.15587/978-617-7319-31-2

Yevseiev, S., Ryabukha, Y., Milov, O., Milevskyi, S., Pohasii, S., Ivanchenko, Y., Ivanchenko, I., Melenti, Y., Opirskyy, I., Pasko, I. (2021). Development of a method for assessing forecast of social impact in regional communities. *Eastern-European Journal of Enterprise Technologies*, *6*(2–114). https://doi.org/10.15587/1729-4061.2021.249313

Yevseiev, S., Tolkachov, M., Shetty, D., Khvostenko, V., Strelnikova, A., Milevskyi, S., Golovashych, S. (2023). The concept of building security of the network with elements of the semiotic approach. *ScienceRise*, *1*, 24–34. https://doi.org/10.21303/2313-8416.2023.002828

Zhou, M. C., Wu, N. (2018). *System modeling and control with resource-oriented Petri nets*. Crc Press.

Zhu, Q., Qin, Y., Zhao, Y., Chunjie, Z. (2020). A hierarchical colored Petri net–based cyberattacks response strategy making approach for critical infrastructures. *International Journal of Distributed Sensor Networks*, ***16***(1), 155014771988980. https://doi.org/10.1177/1550147719889808

Zwilling, M., Klien, G., Lesjak, D., Wiechetek, Ł., Cetin, F., Basim, H. N. (2022). Cyber Security Awareness, Knowledge and Behavior: A Comparative Study. *Journal of Computer Information Systems*, *62*(1), 82–97. https://doi.org/10.1080/08874417.2020.1712269

Received March 14, 2024, revised April 2, 2024, accepted April 5, 2024