Production rule ontology of automatized smart emergency dispatching support of the power system

Vladimir S. Morkun[1]
ORCID: http://orcid.org/0000-0003-1506-9759; morkunv@gmail.com
Ihor A. Kotov[1]
ORCID: http://orcid.org/0000-0003-2445-6259; rioexito@gmail.com
Oleksandra Y. Serdiuk[1]
ORCID: https://orcid.org/0000-0003-1244-7689; o.serdiuk@i.ua
Iryna A. Haponenko[1]
ORCID: http://orcid.org/0000-0002-0339-4581; irinagaponenko44@gmail.com
[1] Kryvyi Rih National University, Vitaliy Matusevych Str. 11. Kryvyi Rih, 50027, Ukraine

ABSTRACT

The research deals with improving methods and systems of control over power systems based on intellectualization of dispatch decision support. There are results of developing a principal trigger scheme of the decision support system algorithm. The proposed model of algorithm visualization in the form of a trigger state network of the computer system provides interaction with power objects of mining and metallurgical complexes and regions. A new interpretation of components of the network trigger model is introduced. The model is interactively related to both user-operator actions and states of power system components. With that, the state of the automata model is associated with fulfillment a set of metarules to control the logical inference. There are new forms of presenting algorithms controlling knowledgebases that interact with the external environment and aggregate primitives of states, triggers and transactions of operations and generalize standard visualization languages of algorithms are proposed. This allows unification of smart systems interacting with the external environment. The authors develop models for representing knowledgebase processing algorithms interacting with power objects that combine states, triggers and transaction operations and generalize standard visualization languages of algorithms. This enables description of functioning database algorithms and their event model, which provides a reliable unification of smart systems interacting with control objects of mining and metallurgical power systems. The research solves the problem of building a knowledgebase and a software complex of the dispatch decision support system based on the data of computational experiments on the power system scheme. The research results indicate practical effectiveness of the proposed approaches and designed models.

Keywords: Ontology; Intelligence; Trigger; Algorithm; Production; Context; Power System; Dispatcher; Automation

INTRODUCTION

Today, aspects of crisis management under time pressure including emergencies associated with power plants, electric grids and mining and metallurgical complex consumers are of particular importance. These situations are emergencies due to great damage occurring in an extremely limited period of time. In such a situation, a decision maker is unable to respond adequately at the required pace. There are two main reasons for that – big data requiring error-free evaluation in a short time and psychological pressure on a decision maker due to enhanced responsibility. Estimates of the human factor impacting in dispatch control practices in power systems of the mining and metallurgical complex indicate that after a year of continuous work reliability of a dispatcher shift reduces and makes up to 46 % of the initial one.

Application of smart systems supporting managerial activity is an efficient trend of control and decision-making automation, decision support systems (DSSs) being the main component here.

Smart hardware and software complexes – the dispatcher's advisers – can be considered the most balanced and effective solution to the problem of automation of emergency control systems (ECSs) within the automated dispatch control system (ADCS) of the power system. They are noted for relatively low initial costs, no extra requirements to the existing data infrastructure during implementation, high technical and economic efficiency. Dispatch advisers include software and hardware decision support complexes, which are implemented by means of artificial intelligence technologies. To date, considerable experience has been accumulated in building DSSs for power systems control [1].

It is necessary to state that to make control over power system modes smart, unification of production of decision support systems is required. Such a model will ensure dynamic transitions between different

forms of knowledge representation within a single expert system when factors, conditions or requirements to process control change dynamically in case of emergencies. Consequently, automation of emergency control of power systems in the mining and metallurgical complex by means of the smart decision support system based on models, methods and tools of evolution of incorporation of professional ontologies is an important scientific problem.

The control over such a complicated and hierarchical system as a power system is chosen to be an object of our research. The automated system of dispatch control is a complicated man-machine complex based on collection, transmission and processing of data of various representation forms. Therefore, efficiency of the ADCS should be mainly assessed in terms of evaluating interaction between the operating dispatch personnel (ODP) and information technologies used.

LITERATURE REVIEW

Preliminary analysis reveals that in current socio-economic, technological, managerial, regulatory and legal conditions, it is efficient to develop and implement a smart DSS for automation of the dispatch emergency control in order to improve efficiency of crisis management in the power system. DSS building requires application of the best structural-logical and software solutions accompanied by a detailed analysis of the existing world experience in theoretical and practical developments in this field.

In [2] analyzes the essence of an emergency and methodology of building control decision support systems. It is noted that application of classical methods of controlling complex dynamic systems with inconsistent decision-making criteria does not provide a desired result. Efficiency, minimum resources used and the highest controlling impact should be the criteria [3, 4]. Given that the DSS is a man-machine system, there is always a problem of control lag and a task of reducing the time of solution generation. Consideration of uncertainty, incompleteness of conditions and criteria for evaluating a situation is of particular attention in the DSS. In [5] formulates basic criteria and requirements of the system approach to building the DSS. The main ideological principle stated is that "the idea ... of methodology is accumulation of knowledge in the computer form of knowledgebases and its subsequent use for decision-making". It is proposed to divide knowledgebases into the following categories: universal, problematic and specific [6]. The ideology of expert systems should be used as the basic software platform for the DSS [5, 7].

There are over 200 expert systems analyzed and applied to various areas of science, engineering and industry.

These systems realize up-to-date principles of the theory, development, implementation and improvement of artificial intelligence systems including expert ones.

Separate mention should be made of decision support systems in power engineering and power system dispatch control.

Computer modeling of the control processes of the power equipment of the mining and metallurgical complex was considered [8].

The paper looks into and analyzes both simulator systems for operational personnel of power facilities and specialized DSSs developed for the professional area of power system management including dispatch emergency control of power system modes [5, 9].

The analysis enables concluding that the automation object should be a man-machine complex as a whole which makes and implements managerial decisions. Automation of control dispatch decision-making is implemented through DSSs based on effective use of professional knowledge formalized in a system of ontologies. Thus, relevance of the problem of automatizing power system mode control by developing and implementing DSS tools into the ADCS environment is confirmed. The solution of this problem is formulated as an integral task of developing and implementing methods of incorporating professional ontologies for automation of power system operation modes in standard and emergency conditions.

Numerous research works are devoted to formalization and systematization of ontologies as tools of representing professional knowledge.

In [10] interprets an ontology model as general formalism for scientific knowledge systematization. Problems and tasks solved in ontological information systems are focused upon. Languages and environments for working with ontologies are analyzed. In [11], current approaches to representation and use of knowledge in smart systems are reviewed and systematized. Special attention is paid to ontological models of knowledge representation. In addition, knowledge description languages and tools for working with knowledge are described in detail.
In [12, 13] are devoted to methods of automatic building of ontologies by automata programming. Methods of building an ontological model of a subject area based on situational modeling are considered. In [14] studies the most general approaches to objectivization and formalization of knowledge in smart systems. Formal logical and probabilistic models for knowledge representation were considered. In [15] is fundamental and generalizing in rela-

tion to the field of building knowledge bases of intelligent systems. In [16] deals with formation of ontologies taking into account specifics of applied professional areas providing one of the possible classifications of ontologies and principles of ontology engineering. In [17] considers problems of compliance of ontology models with peculiarities of professional areas. In [18] studies possibilities of applying ontology models to training and advising smart systems. In [19] systematizes methods of applying ontologies and semantic technologies to knowledge control systems. In particular, application of ontology models and semantic networks to knowledge control systems is proposed. In [20] describes new technologies of ontology synthesis and knowledge control. In [21] considers ontology as a basic tool of big data representation, storage and control. In [22] provides ontology tools for managerial decision support.

There are a large number of definitions of the concept "Ontology" depending on subject areas, tasks to be solved, formalisms used, and the degree of representation detail. One of the most common definitions of the ontology is given by Gruber T. In [21] "Ontology" is a formal, explicit, precise definition (specification) of a shared conceptualization". In general, we can state that ontology is a system of formalisms used to describe a subject area. Ontologies are presented by different structural and linguistic models depending on the tasks to be solved.

THE PURPOSE OF THE ARTICLE

The research aims at automating smart emergency dispatching support of power systems by standardizing mathematical and structural-logical models of production forms of professional ontologies, procedures of generating elementary rules of productions and their networks, developing syntax of productions and their formal-linguistic basis.

According to the problem and the research aim, the objectives of the article are formulated as follows:

– development of a unified formal ontology model of the production form of professional knowledge of dispatch emergency control over power systems based on the theoretical-set model of semantic network clusters;

– development of formal grammar of generating production rule syntax of the structural-linguistic ontology model based on the transaction scheme of the smart system's state;

– development and testing of the DSS software complex for smart support of practical emergency dispatching of the power system based on a new structure of the DSS kernel as an interpreter of the metarules program.

MAIN PART. THE DECISION SUPPORT SYSTEM OF EMERGENCY DISPATCHING OF THE POWER SYSTEM

Let us develop a unified structural and logical model of the production form ontology of professional knowledge representation. We take a model of the semantic network of the ontologies hierarchy as a basis. Let us introduce a formal rule of product representation in terms of semantic network clusters: the production rule R is a set of semantic network clusters N_S, such that

$$N_S^{Clust} = \{N_{Si}^{Clust}, i = 1, n^{Clust}\}, \tag{1}$$

$$\forall(N_{Si}^{Clust})((N_{Si}^{Clust} \in N_S^{Clust} \wedge N_S^{Clust} \subseteq N_S), \tag{2}$$

are interrelated by the following ratio:

$$R: \underset{i=1}{\overset{n-1}{L_{op}}} N_{Si}^{Clust} \rightarrow N_{Sn}^{Clust}, \tag{3}$$

where L_{op} is a logical operation, AND ($\wedge$) or OR ($\vee$) are connectives combining the first n-1 clusters of the semantic network. The operation NOT ($\neg$) is applied with n=1 and usually implemented in knowledgebases into the semantic conditional component (antecedent) of the production; $\rightarrow$ is a logical implication operation [23, 24].

Introducing connectives of conjunction ($\wedge$) and disjunction ($\vee$), we can write for the production of the type "AND":

$$R: N_{S1}^{Clust} \wedge N_{S2}^{Clust} \wedge ... \wedge N_{Sn-1}^{Clust} \rightarrow N_{Sn}^{Clust},$$
$$R: \underset{i=1}{\overset{n-1}{\wedge}} N_{Si}^{Clust} \rightarrow N_{Sn}^{Clust}; \tag{4}$$

for the production of the type "OR":

$$R: N_{S1}^{Clust} \vee N_{S2}^{Clust} \vee ... \vee N_{Sn-1}^{Clust} \rightarrow N_{Sn}^{Clust},$$
$$R: \underset{i=1}{\overset{n-1}{\vee}} N_{Si}^{Clust} \rightarrow N_{Sn}^{Clust}. \tag{5}$$

We should indicate that logical operations realized in relation to semantic network clusters refer also to the marker of activation (actualization) of these clusters: active is 1 (true), inactive is 0 (false).

Activation or actualization of the semantic network (its cluster/segment) will be interpreted as implementation of the semantic network (or its identifier) into a controlled area of memory ("a message board") for its participation in operations of the smart system [25, 26], [27, 28].

Let us assume that when interpreting semantic network clusters included in expressions (1), (2) and (3), each cluster is treated as a single linguistic constant or a meaning of a linguistic variable in any op-

erations within the current formal system. For practical implementation, the amount of semantic network clusters can be very significant, so it is advisable to use cluster identifiers or cluster indices resulted from their dynamic separation taking into account relevant contexts.

Production activation (logical realization of implication) occurs on the basis of calculation (antecedent) of logical expressions

$$\bigwedge_{i=1}^{n-1} N_{Si}^{Clust} \text{ or } \bigvee_{i=1}^{n-1} N_{Si}^{Clust}. \qquad (6)$$

If the result of the expression is true, the production consequent N_{Sn}^{Clast} is written into the controlled memory area ("the message board").

As is known, in a general case a production model is defined by a tuple of the following form [15], [17], [29, 30]

$$R = <S, L, A \rightarrow B, Q>, \qquad (7)$$

where: is the class of situations for which the production R is adequate; L is a production trigger condition; $A \rightarrow B$ is the kernel (structure) of the production; Q is an informal substantiation of the production.

Let us formalize the main part of the production – its kernel, taking the concept of building productions by semantic network clusters proposed in the work as a basis proposed in the work.

The basic terminal alphabet of the production is a set of clusters of the semantic network of the knowledgebase:

$$\Sigma = A_t = \{\varepsilon\} \bigcup N_S^{Clust},$$
$$N_{Si}^{cj} \in N_S^{cj}, \qquad (8)$$
$$N_S^{cj} \subseteq N_S.$$

When building productions, only clusters of a given alphabet should be used. For example, for the c_i-th context:

$$N_S^{Cj} = \{N_{Sk}^{c\,j} \mid k = 1, n_N\}, \qquad (9)$$

where n_N is the number of clusters related to the c_i-th context.

We formalize the production kernel. Let us introduce interpretation of the semantic network of the production Ns as a network of clusters based on the formalism of semantic networks and taking into account evolution of ontologies in the hierarchical system of knowledge representation

$$N_s = <N_S^{Clust}, I^{Clust}, G^{Clust}>, \qquad (10)$$

where: $N_S^{Clust} = \{N_{Si}^{Clust}, i = 1, n^{Clust}\}$ is a set of clusters (concepts) that form a semantic network of the production; $I^{Clust} = \{i_l, l = 1, n_l\}$ is a set of connectives among clusters Ns_i^{Clast} (concepts); $G^{Clust} = \{g_q, q = 1, n_q\}$ is a set of incidence reflections of connectives onto clusters (concepts).

In (10), the model components have the following content. Connectives between semantic subnetwork clusters $I^{Clust} = \{i_l, l = 1, n_l\}$ mean connection of completely identical statements from facts of different clusters, i.e. i^{Clust} can be defined as

$$\forall(s_i, s_k \mid s_i \in f_i \in N_{Si}^{Clust}, s_k \in f_k \in N_{Sk}^{Clust}), \qquad (11)$$
$$(s_i = s_k)i^{Clust} : (s_i, s_k) \rightarrow s,$$

where the incidence reflection looks like

$$g_q : c_l = (s), s \in f_i \wedge s \in f_k. \qquad (12)$$

Thus, production synthesis within the developed theoretical –set interpretation is described as a procedure of concatenation of clusters of semantic network subsets of the knowledgebase meeting the following condition:

$$\exists(N_{S1}^{Clust}, ..., N_{Sn-1}^{Clust}, N_{Sn}^{Clust}) :$$
$$(\bigcap_{k=1}^{n-1} N_{Sk}^{Clust} = s_{sn}) \vee (\bigcap_{k=1}^{n-1} N_{Sk}^{Clust} = s_{an}) \rightarrow \qquad (13)$$
$$(N_{S1}^{Clust}, ..., N_{Sn-1}^{Clust}, N_{Sn}^{Clust} \in R).$$

According to (13), the minimum production can be expressed as follows (Fig. 1).
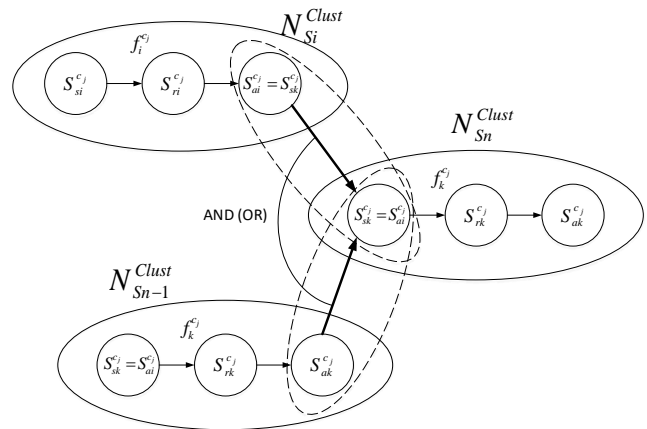
Fig. 1. Combination of clusters of semantic networks making minimum production
Source: compiled by the authors

After combining clusters and making the production, the latter is a directed graph in the following form

$$R = G_{Clust},$$ (14)

$$G_{Clust} = \{V(G_{Clust}), A(G_{Clust})\},$$

where: is a set of clusters of the semantic network, $V(G_{Clust}) = N_S^{Clust} = \{N_{Sk}^{Clust}, k = 1, n_{Clust}\}$; $A(G_{Clust})$ is a set of connectives between clusters.

Let us introduce a separate interpretation of vertex transit for combining clusters G_{Clast} (as for the production directed graph), which depends on the problem conditions and, in general, can be arbitrary:

$$I^{Clust} = (P_s^{Clust}, P_e^{Clust}),$$ (15)

where: P_S^R is an incidentor of primary nodes (clusters) of production links, $P_E^R(N_{Sk}^{Clust}, N_{Sk+1}^{Clust}) = s_i^{c_j}, s_i^{c_j} \in N_{Sn}^{Clust}$. is an incident or of finite nodes (clusters) of production links, $P_E^R(N_{Sk}^{Clust}, N_{Sk+1}^{Clust}) = s_i^{c_j}, s_i^{c_j} \in N_{Sn}^{Clust}$.

Now the general formal model of the product graph R looks like

$$G^R\{V(G_{Clust}), A(G_{Clust}), I^R)\},$$
$$G^R = \{V(G)_{Clust}, A(G)_{Clust}, P_S^R, P_E^R\}.$$ (16)

We develop a formal production language. Since production interpretation depends on the context, it is necessary to consider groups (classes) of contexts when building a formal language.

We define a set of contexts for interpreting the production

$$C^R = \{c_i \mid i = 1, n_c\},$$ (17)

where n_c is the number of contexts (subject areas).

The set of context classes for production is

$$G^{Rc_i} = \{g_m^{Rc_i} \mid m = 1, n_g\},$$ (18)

where n_g is the number of context classes.

Then, the cluster of the semantic network of the production associated with the c_i–th context is:

$$N_R^{ClustC_i} = G_R^{c_i} = \{V(G_{Clust}), A(G_{Clust})\},$$
$$N_R^{c_i} \subset N.$$ (19)

Taking into account the possibility to classify segments (clusters) of the semantic production network by contexts $g_m^{sc_i}$, we can write

$$N_R^{ClustC_i} = \{\{R_{1g_m^{c_i}}^{c_i}\}, \{R_{2g_m^{c_i}}^{c_i}\}, ..., \{R_{n_g g_m^{c_i}}^{c_i}\}\},$$ (20)

where $\{R_{mg_m^{c_i}}^{c_i}\}$ is a production rule making a segment (cluster) of the semantic network and corresponds to the classification feature $g_m^{c_i}$.

We assume that the knowledgebase is standardized and the singled out clusters of the network are not doubled, while properties of knowledgebase productions should be the following:

$$N_R^{C_i} = R_{1g_m^{c_i}}^{c_i} \bigcup R_{2g_m^{c_i}}^{c_i} ... \bigcup R_{n_g g_m^{c_i}}^{c_i} = \bigcup_{m=1}^{n_g} R_{mg_m^{c_i}}^{c_i},$$ (21)

$$R_{1g_m^{c_i}}^{c_i} \bigcap R_{2g_m^{c_i}}^{c_i} ... \bigcap R_{n_g g_m^{c_i}}^{c_i} = \bigcap_{m=1}^{n_g} R_{mg_m^{c_i}}^{c_i} = \varnothing,$$ (22)

$$\forall R_{mg_m^{c_i}}^{c_i} (R_{mg_m^{c_i}}^{c_i} \subseteq N_R^{C_i}),$$ (23)

$$\forall R_{pg_m^{c_i}}^{c_i} \forall R_{qg_m^{c_i}}^{c_i} (R_{pg_m^{c_i}}^{c_i} \neq R_{qg_m^{c_i}}^{c_i}).$$ (24)

In (21) essentially describes the $N_R^{C_i}$ production network of the C_i-th context with respect to which inductive or deductive inferences are possible.

Let us introduce production syntax and the production network:

<Production network> :: = <production> [<production>];

< Production> :: = < production type> <cluster> [<cluster>] < cluster>;

< Production type> :: = AND | OR;

<Cluster> :: = <semantic network>.

It should be noted that to expand a production network in RAM, it is necessary to specify incidence of the network graph.

We develop a formal language of the model representation of the production network of the professional area of the ontologies hierarchy.

Let us assume that the formal language of the production belongs to a certain class of semantic networks $G^{C_i}s$ in the current context –

$$\forall R_k^{c_i}, k = 1, n_s (\bigvee_{k=1}^{n_g} R_k^{c_i} \in N_{g_m^{c_i}}^{c_i}).$$ (25)

We define the production language for the subject area $c_i \in C$ and the m-th network class as follows

$$L(G)_{g_m^{c_i}}^{Rc_i} = < \Sigma_{g_m^{c_i}}^R, N_{g_m^{c_i}}^R, P_{g_m^{c_i}}^R, S_{g_m^{c_i}}^R >,$$ (26)

where: G is the formal grammar of the production network; $\sum^R$ is the main terminal alphabet of the production network; N^R is an auxiliary finite nonterminal alphabet; P^R is rules of substitution (production) of the formal network grammar $\exists a, \exists b, (a, b) \in P^R : a \to b$; S is a starting nontermi-

nal symbol of the grammar G; $N^R \bigcap \Sigma^R = \varnothing$, $P^R \subset ((N^R \bigcup \Sigma^R)^+ \times (N^R \bigcup \Sigma^R)^*)$.

Now let us set the rules of formal grammar P on the production network for the language $L(G)^{Rc_i}_{g_m^{c_i}}$

$$N^P \rightarrow P^R_{g_m^{c_i}} \mid P^R_{g_m^{c_i}} [P^R_{g_m^{c_i}}], \qquad (27)$$

$$P^R \rightarrow N^{Clust}_{Ra} N^{Clust}_{Rc}, \qquad (28)$$

$$N^{Clust}_{Ra} \rightarrow N_{sa} \mid [N_{sa}], \qquad (29)$$

$$N^{Clust}_{Rc} \rightarrow N_{sc}, \qquad (30)$$

where: $P^R_{g_m^{c_i}}$ is the network production. If the segmentation feature is the contextual one $g_m^{c_i}$, $N^{Pc_i} = N^{Pc_i}_{mg_m^{c_i}}$; N^{Clust}_{Ra}, N^{Clust}_{Rc} are corresponding clusters of semantic networks specified as a production antecedent and consequent; N_{sa} is a semantic network referred to the cluster of the antecedent network; N_{sc} is a semantic network referred to the cluster of the consequent network.

Let us generalize the language of context production subnetworks for all the classes of knowledgebase contexts on production networks

$$L(G)^R = < \Sigma^R, N^R, P^R, S^R >, \qquad (31)$$

where: $\Sigma^R = N^R = \{n_{Rk} \mid k=1, n_R\}$ is all the context networks of production knowledgebase networks; $N^R = \{\{N_{s_{1g_m^{c_0}}}^R\}, \{N_{s_{2g_m^{c_0}}}^R\} ... \{N_{s_{n_g g_m^{c_0}}}^R\}\}$;

$N_{s_{mg_m^c}}^R$ = {subnetwork, subnetwork_class, network_layer}.

Let us summarize formal grammar rules - P^R for the whole level of the production network taking into account that there are four non-terminals in the formal language of the production network $L(G)^R$

$Ns \rightarrow$ <network layer>;
<network layer> $\rightarrow$ <subnetwork class><subnetwork>;

<network_class> $\rightarrow \forall g_m (\{N_{smg_m^{c_o}}\} \subseteq N_S)$;

<subnetwork> $\rightarrow \forall N_{Sk} (N_{Sk} \in N_S)$;

<production> $\rightarrow P^R_{g_m^{c_i}}$.

Let us develop a structural and linguistic model of the ontology for the production network

knowledgebase – KB_P. We use the generalized formula

$$O_{KB_P} = < X^P, R^P, F^P >. \qquad (32)$$

$P^R_{g_m^{c_i}}$ productions are the concepts of the production network ontology, which in general case form multiple concatenations among themselves. For a network of any context, we have

$$X^P = N^P = \{P_m \mid m=1, n_P\} = \\ = \{\{N_{Rka}^{Clust} N_{Rc}^{Clust} \mid k=1, n_{Clust}\} \mid m=1, n_P\} \qquad (33)$$

or

$$X^P = \{N_{Rka}^{Clust} N_{Rc}^{Clust} \mid k=1, n_{Clust}\} = \\ = \{\{N_m^S \mid m=1, n_s\}\}.$$

Since for a given knowledgebase (KB_P), local context productions have connectives, the set of relations between them is not empty– $R^P \neq \varnothing$.

To define a set of F^P interpretation functions, we will assume that part of the productions (or semantic subnetworks forming these productions) can be used to interpret other productions of the current KB_P level. In this case, the production subnetworks can be divided into situational groups - classes - subsets of interpretive productions (with index 1) and subsets of interpretive products (with index 2), as shown in the expression

$$N^R = \{\{N_1^R\}, \{N_2^R\}\}, \qquad (34)$$

where $N_1^R \bigcup N_2^R = N^R$ is the whole set of productions of the level, $N_1^R \bigcap N_2^R = \varnothing$.

Then the interpretation function fj in general will be

$$f_j^R : Op(\{(R_{21}^{c_0}, R_{22}^{c_0}, ..., R_{2k}^{c_0}, I_j)\}) \rightarrow (R_{1j}, I_j), \qquad (35)$$

$$\exists(R_{1i}^s \in N_1^R), \exists(R_{21}^s, R_{22}^s, ..., R_{2k}^s \in R_2^s)$$

$$(R_{1i}^{c_0} = f(R_{21}^{c_0}, R_{22}^{c_0}, ..., R_{2k}^{c_0})).$$

where Op is the product aggregation operation. For production networks such an operation is usually a logical production sequence, the content of which is combination of output and input clusters.

On the basis of the developed structural-set models, the formal model of the unified ontology of production networks for professional ontologies hierarchy of the "network structure with possibility of expansion" type is obtained, which is expressed as follows

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)   Information technologies in energy
engineering and manufacturing   173

$$O^R = < \bigcup_{j=1}^{N} [N_{1j}^R \cup N_{2j}^R], R^R, F^R > . \qquad (36)$$

Here is an example of practical use of the developed mathematical models of production network representation and a model of professional ontology O^R of production networks of the knowledgebase KB^R. The developed mathematical models are invariant with respect to the subject areas. As an example of the professional area, let us choose a logical description of dispatcher personnel's actions for operational switching during emergency responses as shown in [31].

Let us introduce a set of atomic statements referring to the same context c_0, where c_0 = "Elimination of process disturbances when disconnecting buses of 150-330-750kV":

S_1^{c0} is "short circuit", S_2^{c0} is "on", S_3^{c0} is "buses or equipment in the thermal power station area", S_4^{c0} is "connected to", S_5^{c0} is "differential bus protection», S_6^{c} is «connection", S_7^{c0} is "breakdown (disturbance)", S_8^{c0} is "connection switcher", S_9^{c0} is "false performance", S_{10}^{c0} is "protection of autotransformer", S_{11}^{c0} is "or", S_{12}^{c0} is "switching off", S_{13}^{c0} is "power station", S_{14}^{c0} is "belongs", S_{15}^{c0} is "substation", S_{16}^{c0} is "state – without voltage", S_{17}^{c0} is "power supply of consumer", S_{18}^{c0} is "MV of power station", S_{19}^{c0} is "state – out of service" , S_{20}^{c0} is "ACR", S_{21}^{c0} is "ATS", S_{22}^{c0} is "duty personnel", S_{23}^{c0} is "must (has to)", S_{24}^{c0} is "provide voltage to", S_{25}^{c0} is "consumer", S_{26}^{c0} is "quantity", S_{27}^{c0} is "two bus systems", S_{28}^{c0} is "bus sectionalizing switch (BSS)", S_{29}^{c0} is "conduct testing", S_{30}^{c0} is "switch off", S_{31}^{c0} is "conduct simultaneously", S_{32}^{c0} is "type of equipment".

A set of facts of semantic network clusters that form productions:

$$f_1 = (s_1^{c_0}, s_2^{c_0}, s_3^{c_0}), f_2 = (s_5^{c_0}, s_4^{c_0}, s_3^{c_0}),$$
$$f_3 = (s_1^{c_0}, s_2^{c_0}, s_6^{c_0}), f_4 = (s_7^{c_0}, s_{32}^{c_0}, s_5^{c_0}),$$
$$f_5 = (s_7^{c_0}, s_{32}^{c_0}, s_8^{c_0}), f_6 = (s_9^{c_0}, s_{32}^{c_0}, s_5^{c_0}),$$
$$f_7 = (s_9^{c_0}, s_{11}^{c_0}, s_{10}^{c_0}), f_8 = (s_{12}^{c_0}, s_6^{c_0}, s_4^{c_0}),$$
$$f_9 = (s_6^{c_0}, s_4^{c_0}, s_3^{c_0}), f_{10} = (s_3^{c_0}, s_{14}^{c_0}, s_{13}^{c_0}),$$
$$f_{11} = (s_3^{c_0}, s_{14}^{c_0}, s_{15}^{c_0}), f_{12} = (s_3^{c_0}, s_7^{c_0}, s_{16}^{c_0}),$$
$$f_{13} = (s_7^{c_0}, s_{32}^{c_0}, s_{17}^{c_0}), f_{14} = (s_7^{c_0}, s_{32}^{c_0}, s_{18}^{c_0}),$$
$$f_{15} = (s_7^{c_0}, s_{11}^{c_0}, s_{19}^{c_0}), f_{16} = (s_7^{c_0}, s_{32}^{c_0}, s_{20}^{c_0}),$$
$$f_{17} = (s_7^{c_0}, s_{32}^{c_0}, s_{21}^{c_0}), f_{18} = (s_{19}^{c_0}, s_{32}^{c_0}, s_{20}^{c_0}),$$

$$f_{19} = (s_{19}^{c_0}, s_{32}^{c_0}, s_{21}^{c_0}), f_{20} = (s_3^{c_0}, s_{26}^{c_0}, s_{27}^{c_0}),$$
$$f_{21} = (s_3^{c_0}, s_7^{c_0}, s_{16}^{c_0}), f_{22} = (s_{30}^{c_0}, s_{32}^{c_0}, s_{28}^{c_0}),$$
$$f_{23} = (s_{31}^{c_0}, s_{29}^{c_0}, s_{27}^{c_0}), f_{24} = (s_{22}^{c_0}, s_{23}^{c_0}, s_{24}^{c_0}),$$
$$f_{25} = (s_{23}^{c_0}, s_{24}^{c_0}, s_3^{c_0}), f_{26} = (s_{23}^{c_0}, s_{24}^{c_0}, s_{25}^{c_0}).$$

We create subnetworks with a relevance index

$$N_1^{RC_0} = \{f_{10}, f_{11}, f_{12}, f_{20}, f_{22}, f_{23}\}, 1;$$
$$N_{21}^{RC_0} = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}\}, 1;$$
$$N_{22}^{RC_0} = \{f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{24}, f_{25}, f_{26}\}, 1;$$

The interpretation function for the production network level will look like this

$$f_1^R : Op(\{(N_{21}^{Rc_0}, 1), (N_{22}^{Rc_0}, 1)\}) \rightarrow (N_1^{Rc_0}, 1).$$

Thus, the structural-linguistic model of the unified professional ontology of production networks is built.

Let us specify the operation within the product transaction model

$$o_{ik} = < ID_{ik}, l_{ik}, C_{ik} >, \qquad (37)$$
$$C_{ik} = < N_{oik}, A_{oik} >,$$

where: ID_{ik} where ID_{ik} is an identifier (number) of the current operation; l_{ik} is the parameter of the operation start; C_{ik} is a command (operation block); N_o is the name of a command (operation); $A_o = \{a_{ok} \mid k = 1, n_a\}$ are operands (parameters) of a command (operation).

Now let us detail graphical interpretation of one state of performing the transaction algorithm of the program system considering the introduced models and show it in Fig. 2.

Fig. 2 shows all the components of the above formal models – a state, a trigger, a transaction and operations. They are important for program implementation. When visualizing the algorithm of transactions, we can limit ourselves to displaying the operations and their parameters.

Implementation of the proposed visualization model requires developing a formal language to branch a scheme of the functioning algorithm of the trigger state program machine.
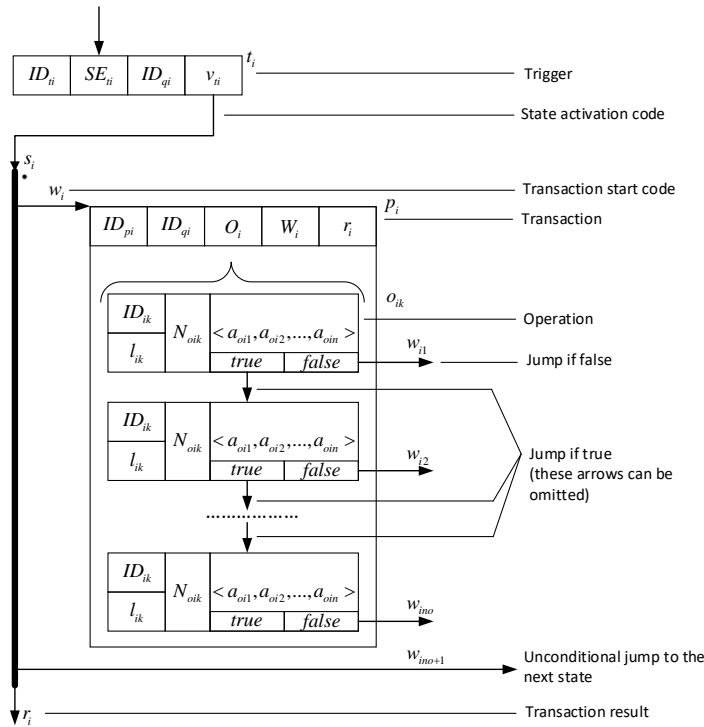
Fig. 2. **Visualization of one state of the program system**
Source: compiled by the authors

Let us consider that the program is a subset of the set of graph paths of the state network

$$\operatorname{Pr}og = B_{prog},$$
$$B_{prog} = \{b_i \mid i = 1, n_b\},$$
$$b_i = \{s_{i,k}, s_{i,k+1}, \dots, s_{i,n_{bi}}\},$$
$$\forall s_{ik} \forall s_{ik+1} \forall k (s_{ik} \in S \wedge s_{ik+1} \in S_i \wedge$$
$$\wedge k = 1, 2, \dots, n_o) : s_{ik} \prec s_{ik+1}, \quad (38)$$
$$B_{prog} \subseteq N_s,$$

where: $\operatorname{Pr}og = B_{prog}$ is a program (an algorithm of performing a series of transactions); $B_{prog} = \{b_i \mid i = 1, n_b\}$ is a set of branches (paths) of the program algorithm; b_i is the i-th branch of the program algorithm specified as a rigid sequence of the trigger network's states; N_S is a complete network of states.

The resulting algorithm visualization model in the form of a trigger state network of the program system interacts with the external environment. The operations in the model can be considered at different levels. If operations are considered as elementary commands of the format

　　　　<operation code> <operand codes>,
the model enables reflecting the operation of an automata processor or an interpreter at a low machine instruction level.

If operations are considered as complete expressions (or their sequential sets), the model ena-

bles visualizing structural algorithms in high-level programming languages.

If operations are considered predicates with respect to a set of parameters, the model allows implementing the functional approach at a higher level of structural functional programming.

Besides, if operations are considered as queries in relation to a database (SQL queries) or a knowledgebase (packages of metarules), the model allows describing the process of data and knowledge management by forming query packages – transactions. The model can represent parallel computational processes by running several states from a single trigger.

The developed formal apparatus of the trigger model representation of the software complex operation allows us to proceed to the synthesis of the structure and description of functioning methods of the DSS. Taking into account the specifics of the professional subject area, the generalized structural scheme of including the DSS in the emergency control cycle of power systems is given in Fig. 3.

Fig. 3 shows directions of the input (input arc I) and output (output arc II) information paths. A power plant or its component (power object) is considered as a control object (CO) – (block 2). The control subject (CS) (block 3) is represented by the ODP of the power plant as instantiation of the decision maker. The DSS (block 1) acts as an information interface between the CS and the CO. In its turn, the role of hardware-software interface between the
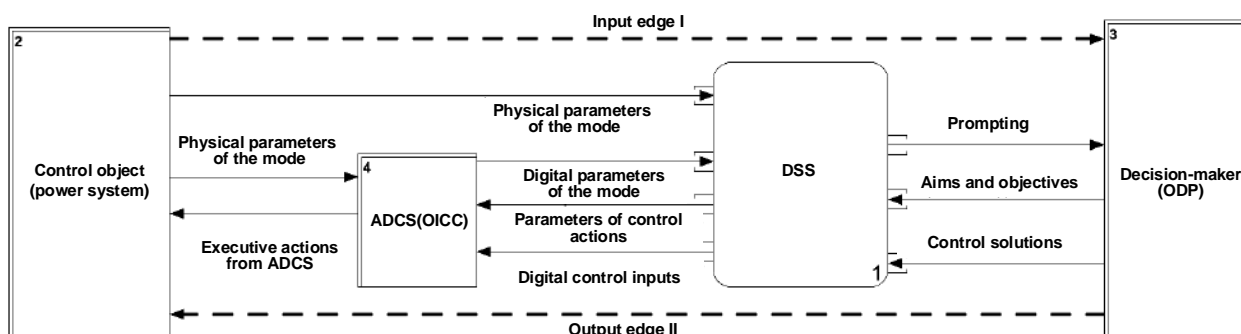
Fig. 3. **Context diagram of the generalized structural scheme of the DSS inclusion into the emergency control cycle of power systems**
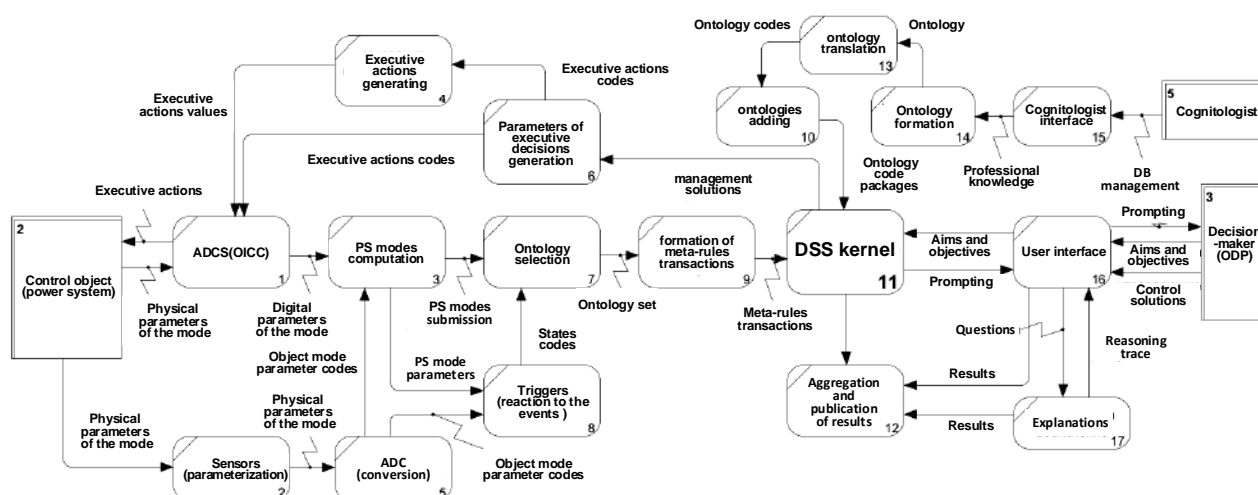*Source:* **compiled by the authors**

DSS and CO is performed by the operating information controlling complex of ADCS tools (block 4) [32, 33]. The tools of the operating information and controlling complex (OICC) convert physical indicators of the power system's mode into digital codes which are fed to the DSS triggers. The DSS receives aims and objectives from the ODP, as well as managerial decisions to be justified, and provides formed recommendations for final decision-making. Codes of managerial decisions (actions) and generated numerical control parameters are sent to the ADCS. The ADCS tools direct and realize controlling physical actions in relation to power objects' equipment.

Guided by the general contextual scheme, let us carry out detailization of the DSS structure corresponding to requirements of the tasks to be solved. The DSS should process knowledge based on metarule transactions. The developed structural diagram of DSS decomposition is shown in Fig. 4.

Fig. 4 shows not only the structure of the DSS, but also its connection with the ADCS tools of the power system. This circumstance enables two operational modes of the smart system in parallel – the asynchronous interactive mode with user participation (a decision maker – the ODP of the power plant) and practically autonomous (automatic) real-time mode of technological processes of the control object.

Based on the DSS diagram and the developed formal trigger model of visualization of the software system functioning, a generalized algorithm of DSS operation is developed and shown in Fig. 5. For simplicity, the diagram does not show all attributes of states, transactions, and operations. Algorithm stages are represented by states to which consecutive sections are linked. Algorithm branching is performed on completion of a section, or on the result of a particular operation. The *s7* state is terminal and contains no operations

Fig. 4. **Diagram of DSS decomposition within the framework of controlling industrial object (power system)**
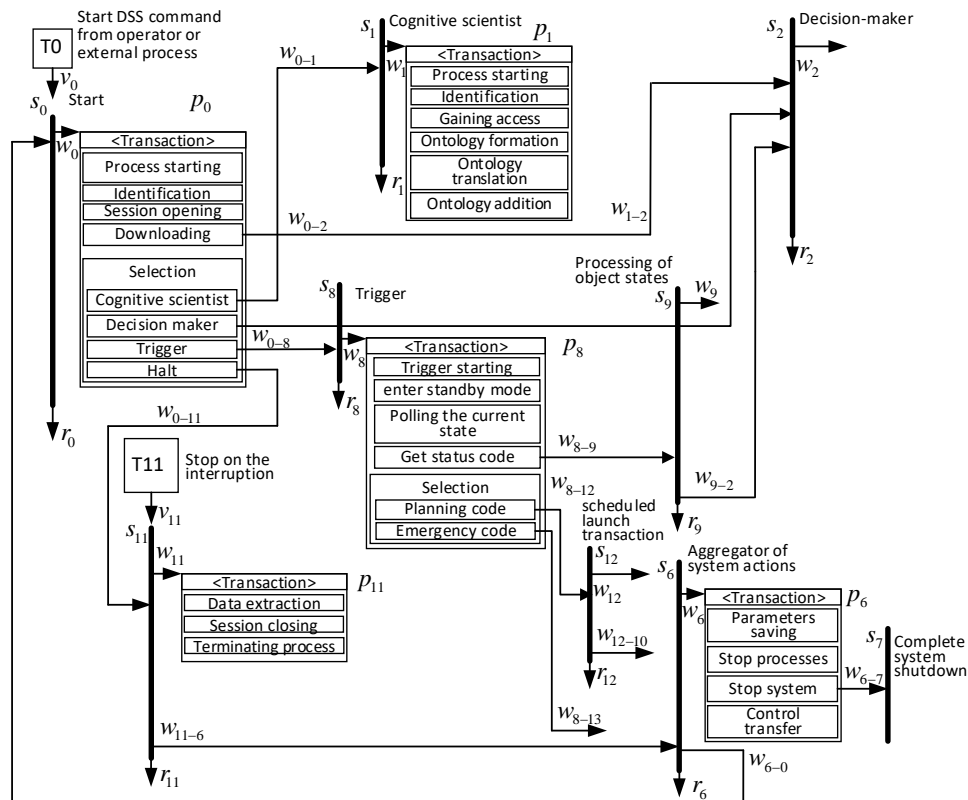*Source:* **compiled by the authors**

Fig. 5. **Fragment of the expanded triggering scheme of the DSS functioning algorithm**
Source: **compiled by the authors**

The diagram (Fig. 5) reflects all the basic components of algorithms – linear sections of work execution, branching and returning (cycles). This particular scheme is extended and contains entire program blocks as operations, which can also be represented by a similar scheme. One transaction in the body of a transaction can be another transaction. One transaction in the body of a transaction can be another transaction this nesting is implemented by a mechanism returning each operation the TRUE or FALSE logical value. The suggested mechanism of presenting algorithm schemes uses minimal tools of visualization, allows compact and adequate presentation of functioning, work scenarios and algorithms of software complexes execution.

The developed functional scheme of the DSS is an event-driven complex. There are 13 main states (stages) of the system in the generalized model of DSS functioning. According to the developed functioning model, the activity of each state is indicated by its labeling (e. g. #($s0$) = 1). All parameters of formal models of states that characterize cycles of the DSS operation are shown in Fig. 6.

Algorithm execution starts from the state s_0. The state can be activated by the user (operator) or externally by means of the ADCS OICC via the T_0 trigger. The trigger throws the flag v_0, and activates the state s_0. The state, in turn, generates the w_0 character which is passed to the P_0 transaction. If the input character is accepted, the transaction is started and executed in the transaction order. The "Choice" operation implements four branching options for the following stages of the DSS algorithm execution. When the operator selects the "Stop" operation, transition to the stop state with the corresponding stop operations of the current process takes place. In addition, the stop condition can be generated by the external environment via the trigger T_{11}.

When using the interrupt mechanism, self-diagnostic procedures for the smart system can be implemented. The DSS architecture is designed in such a way that its algorithm has multiple entry points and is activated by event triggers, which also involves processing its own failures. Such events can be the system's own failures. In this case, interruption and processing of the emergency occur.

Software implementation of the DSS algorithm requires storage and transmission of multiple signals, identifiers, flags, and state results. Besides, transmitting actual parameters to operations and returning the results of intermediate calculations is required. A stack is used for this purpose. The stack can be provided by the compiler of the selected platform of a development and a programming language.

.

Start
$t_0 \rightarrow v_0$
$\#(s_0) = 1$
$q_0 \rightarrow w_0 \rightarrow p_0$
$p_0 \rightarrow r_0$
$p_0 \rightarrow (w_{0-1}, w_{0-2}, w_{0-8}, w_{0-11})$
$\delta(s_0, w_{0-1}) \rightarrow s_1$
$\mu(s_0, w_{0-1}) \rightarrow r_1$
$\delta(s_0, w_{0-2}) \rightarrow s_2$
$\mu(s_0, w_{0-2}) \rightarrow r_2$
$\delta(s_0, w_{0-8}) \rightarrow s_8$
$\mu(s_0, w_{0-8}) \rightarrow r_8$
$\delta(s_0, w_{0-11}) \rightarrow s_{11}$
$\mu(s_0, w_{0-11}) \rightarrow r_{11}$

Cognitologist
$\#(s_1) = 1$
$q_1 \rightarrow w_1 \rightarrow p_1$
$p_1 \rightarrow r_1$
$p_1 \rightarrow (w_{1-2})$
$\delta(s_1, w_{1-2}) \rightarrow s_2$
$\mu(s_1, w_{1-2}) \rightarrow r_2$

Decision-maker
$\#(s_2) = 1$
$q_2 \rightarrow w_2 \rightarrow p_2$
$p_2 \rightarrow r_2$
$p_2 \rightarrow (w_{2-3}, w_{2-5}, w_{2-10})$
$\delta(s_2, w_{2-3}) \rightarrow s_3$
$\mu(s_2, w_{2-3}) \rightarrow r_3$
$\delta(s_2, w_{2-5}) \rightarrow s_5$
$\mu(s_2, w_{2-5}) \rightarrow r_5$
$\delta(s_2, w_{2-10}) \rightarrow s_{10}$
$\mu(s_2, w_{2-10}) \rightarrow r_{10}$

Problem Statement
$\#(s_3) = 1$
$q_3 \rightarrow w_3 \rightarrow p_3$
$p_3 \rightarrow r_3$
$p_3 \rightarrow (w_{3-4})$
$\delta(s_3, w_{3-4}) \rightarrow s_4$
$\mu(s_2, w_{3-4}) \rightarrow r_4$

Logical conclusion
$\#(s_4) = 1$
$q_4 \rightarrow w_4 \rightarrow p_4$
$p_4 \rightarrow r_4$
$p_4 \rightarrow (w_{4-5})$
$\delta(s_4, w_{4-5}) \rightarrow s_5$
$\mu(s_4, w_{4-5}) \rightarrow r_5$

Output results
$\#(s_5) = 1$
$q_5 \rightarrow w_5 \rightarrow p_5$
$p_5 \rightarrow r_5$
$p_5 \rightarrow (w_{5-6})$
$\delta(s_5, w_{5-6}) \rightarrow s_6$
$\mu(s_5, w_{5-6}) \rightarrow r_6$

Aggregator system actions
$\#(s_6) = 1$
$q_6 \rightarrow w_6 \rightarrow p_6$
$p_6 \rightarrow r_6$
$p_6 \rightarrow (w_{6-7}, w_{6-0})$
$\delta(s_6, w_{6-7}) \rightarrow s_7$
$\delta(s_6, w_{6-0}) \rightarrow s_0$
$\mu(s_6, w_{6-0}) \rightarrow r_0$

Trigger
$\#(s_8) = 1$
$q_8 \rightarrow w_8 \rightarrow p_8$
$p_8 \rightarrow r_8$
$p_8 \rightarrow (w_{8-9}, w_{8-12}, w_{8-13})$
$\delta(s_8, w_{8-9}) \rightarrow s_9$
$\mu(s_8, w_{8-9}) \rightarrow r_9$
$\delta(s_8, w_{8-12}) \rightarrow s_{12}$
$\mu(s_8, w_{8-12}) \rightarrow r_{12}$
$\delta(s_8, w_{8-13}) \rightarrow s_{13}$
$\mu(s_8, w_{8-13}) \rightarrow r_{13}$

Object state handling
$\#(s_9) = 1$
$q_9 \rightarrow w_9 \rightarrow p_9$
$p_9 \rightarrow r_9$
$p_9 \rightarrow (w_{9-2})$
$\delta(s_9, w_{9-2}) \rightarrow s_2$
$\mu(s_9, w_{9-2}) \rightarrow r_2$

Tracing
$\#(s_{10}) = 1$
$q_{10} \rightarrow w_{10} \rightarrow p_{10}$
$p_{10} \rightarrow r_{10}$
$p_{10} \rightarrow (w_{10-5})$
$\delta(s_{10}, w_{10-5}) \rightarrow s_5$
$\mu(s_{10}, w_{10-5}) \rightarrow r_5$

Stop on interruption
$t_{11} \rightarrow v_{11}$
$\#(s_{11}) = 1$
$q_{11} \rightarrow w_{11} \rightarrow p_{11}$
$p_{11} \rightarrow r_{11}$
$p_{11} \rightarrow (w_{11-6})$
$\delta(s_{11}, w_{11-6}) \rightarrow s_6$
$\mu(s_{11}, w_{11-6}) \rightarrow r_6$

Scheduled start of a transaction
$\#(s_{12}) = 1$
$q_{12} \rightarrow w_{12} \rightarrow p_{12}$
$p_{12} \rightarrow r_{12}$
$p_{12} \rightarrow (w_{12-10})$
$\delta(s_{12}, w_{12-10}) \rightarrow s_{10}$
$\mu(s_{12}, w_{12-10}) \rightarrow r_{10}$

Start interruption
$\#(s_{13}) = 1$
$q_{13} \rightarrow w_{13} \rightarrow p_{13}$
$p_{13} \rightarrow r_{13}$
$p_{13} \rightarrow (w_{13-10})$
$\delta(s_{13}, w_{13-10}) \rightarrow s_{10}$
$\mu(s_{13}, w_{13-10}) \rightarrow r_{10}$

Fig. 6. **Formal models of state steps of the DSS functioning algorithm**
Source: compiled by the authors

Using the mechanism of stack saving and parameter tracing, the error-free movement of the system in the state space with the possible branching and cyclic returns is provided. Based on the developed above structural decomposition scheme of the DSS included in the control framework of an industrial object (power system), we can synthesize a structural and logical trigger model and determine functioning methods of the decision support system kernel. According to the established classical architecture of smart systems, the DSS kernel includes its two main modules – the logical inference block (LIB) and the knowledgebase [34, 35]. The LIB receives a problem in a formalized form and solves it by manipulating knowledgebase concepts [36, 37]. In this case, the mechanism of knowledge processing is usually defined in advance and is rigidly bound to the forms of knowledge representation in a particular system and subject area. Changes in the professional area, structure or way of formalizing the knowledgebase entail the need to process the LIB. Besides, the classical approach clearly specifies a kernel: the LIB is a software module (usually a machine code) that processes knowledge representation structures known to it in the knowledgebase, while knowledgebases are rules of logical inference formalized to a certain degree within a particular subject area.

However, in complex structured, hierarchically organized professional fields, such as emergency control of power system modes, the classical scheme of the DSS kernel does not allow using all forms of knowledge representation of the subject area in a single smart system. Therefore, the context of the study addresses the problem of simultaneous use of different forms of knowledge presented in the form

of ontologies and evolutionarily generalizing each other. This approach cannot be implemented within the framework of the classical structure of the DSS.

In order to overcome this problem, a new structural-logical architecture of the DSS kernel is proposed and methods of its functioning are determined. The proposed architecture is shown in Fig. 7. Fig. 7a shows the DSS kernel in the form of a single generalized block operating in the control environment of an industrial object – a power system. As can be seen, the DSS kernel is connected with the external object, receives from it parameters of functioning and ontologies of knowledge processing metarules formed on their basis. Recommendations and managerial decisions, as well as data sets for visualization and publication are generated at the output of the kernel

Fig. 7b shows the classical two-module architecture of the DSS kernel. However, as noted above, it cannot be implemented within the framework of the research tasks solved and in an integrated subject environment under consideration.

It is suggested to replace the LIB with the software processor - an interpreter which executes the byte-code of knowledgebase processing programs. Translated sets (packages, series, sequences) of metarules ontologies act as such programs. This makes it possible to build a LIB as a universal processor

(interpreter) which does not depend on the problem area and task structure. Metarule packages (programs) implement ration information interaction between the interpreter and the knowledgebase. So, it is logical to call them "transactions".

Metarules receive actual parameters for processing from an external control object (in this case - the OICC of the ADCS of the power system). As can be seen from the new scheme, when comparing it with the classical architecture, the transaction (the metarule program) is both part of the knowledgebase and part of the LIB. The main advantage of such DSS kernel scheme is separation of the LIB architecture and methods of functioning from specifics of the knowledgebase. The logical inference is treated as execution of a metarule program translated into the low-level byte-code. This ensures versatility of the interpreter and increases its efficiency.

On the basis of the received formal and structural-logical models, the software complex of emergency control support system of the power system is developed.

The main setting panel of the software complex for a subset of emergencies with selection of triggers and real-time monitoring of sensor point parameters is shown in Fig. 8.
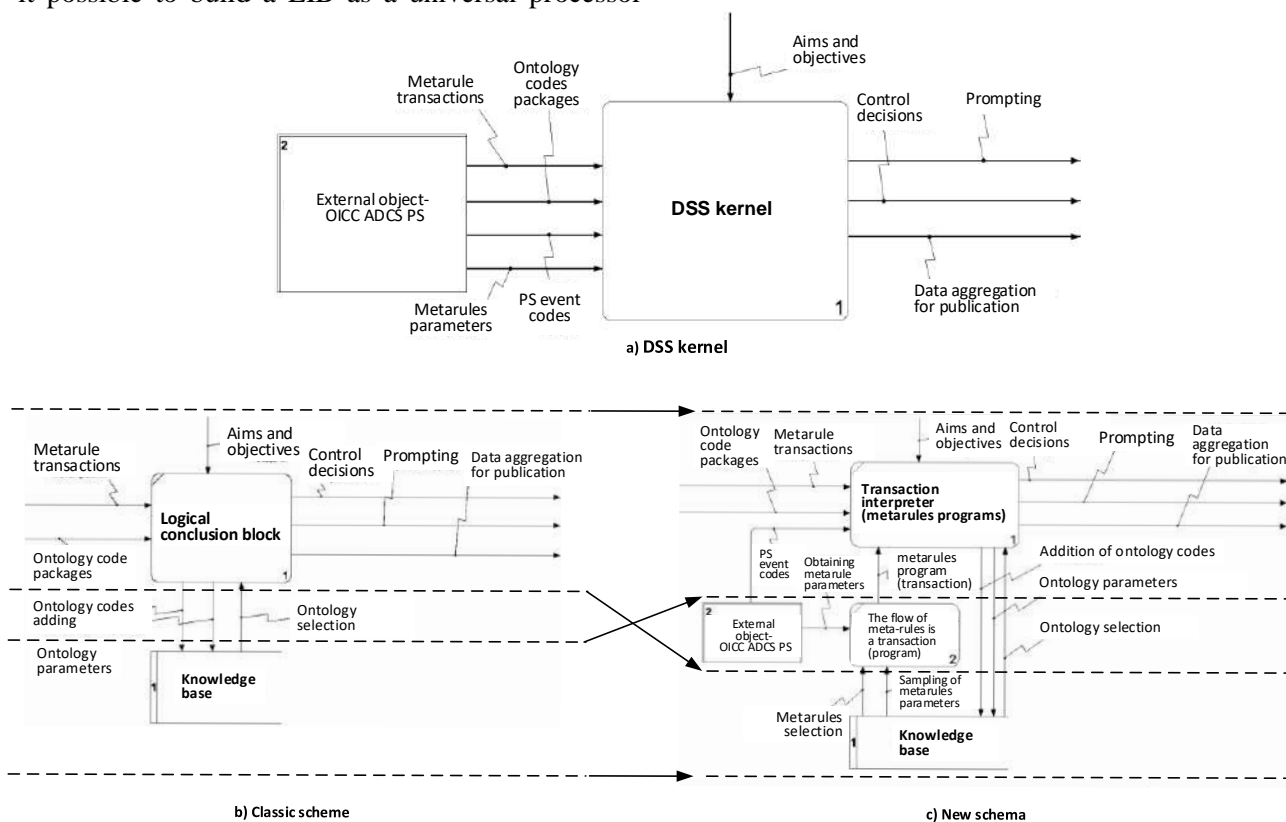

Fig. 7. **New structure of the DSS kernel as an interpreter of the metarule program**
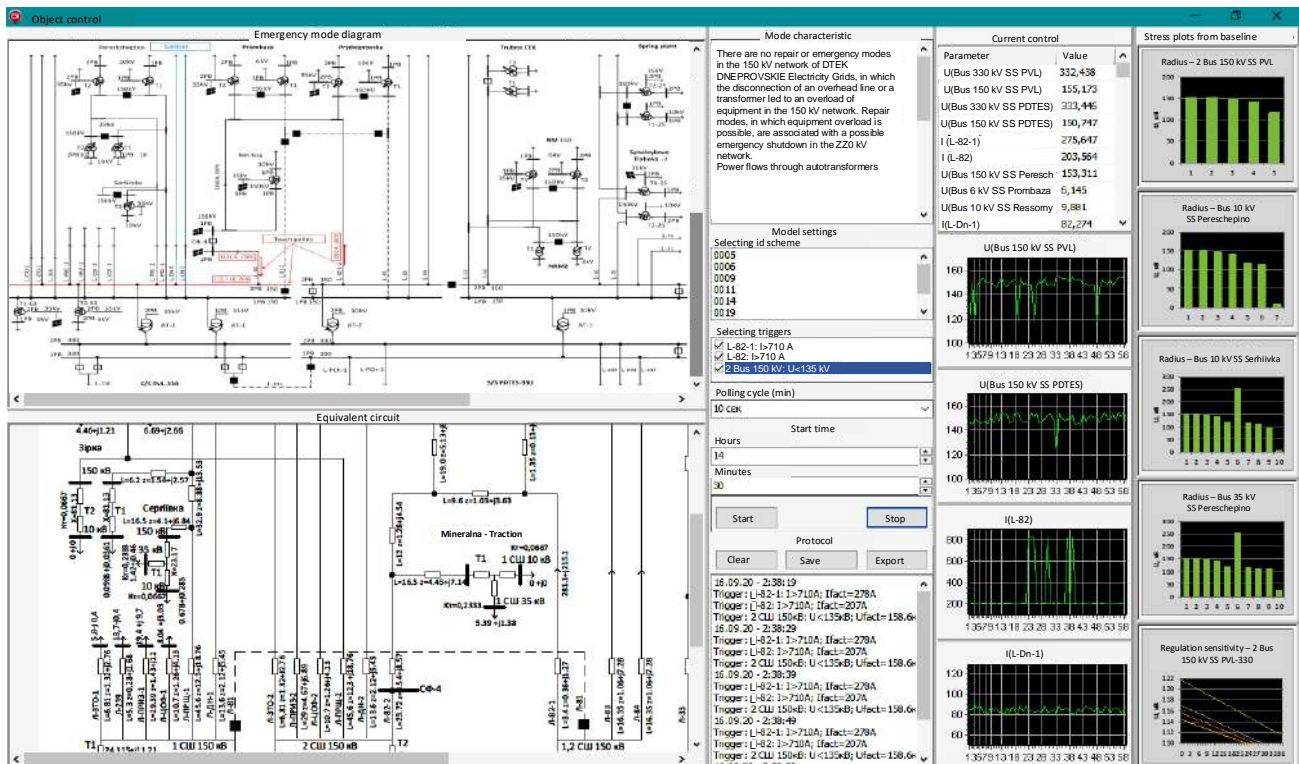Source: compiled by the authors

Fig. 8. **Setting emergency control parameters**
Source: compiled by the authors

To illustrate and confirm practical operability of the DSS software package, a severe emergency is selected with confirmed calculation of the emergency mode of the 330 kV grid of the JSC *DTEK Dnipro Power Grid*. It should be noted that modes of cascade emergencies being especially serious are of particular interest. However, the developed mathematical models and practical implementation of the DSS software package are invariant in relation to the types of emergencies and do not require additional adjustment. All that is needed is the layout of an appropriate knowledgebase. When the dispatcher's operational materials are worked out, the mentioned problem can be solved.

Operation of the DSS when controlling a specific emergency is under consideration – shutdown of the OHL 150kV L-81 (Pavlogradskaya-330 Substation - PdTES-330) for repair; emergency shutdown of OHL 330 kV L-234 overhead power line (Pavlogradskaya-330 Substation - PdTES-330). The emergency mode is characterized by reduction of voltage at the 150kV BS of Pavlogradskaya-330 and the current overload of the OHL 150kV L-82 (Pavlogradskaya-330 Substation - PdTES-330). Power flows through autotransformers of the mentioned substations exceed permissible load levels in emergency modes caused by shutdown of the OHL 330kV L-234 (Pavlogradskaya-330 substation - PdTES-330).

Basic settings of the software package are as follows: the diagram of the 330kV power network of DTEK with summer 2020 parameters; Sensor points and monitored mode parameters are indicated for the network. The bottom panel of Fig. 8 shows the corresponding design substitution diagram. The following triggers are selected: 1) L-82-1 I> 710 A, 2) L-82-2 I> 710 A, 3) 2 BS 150kV U < 135kV. The interrogation cycle is 10s. The time of triggers interrogation start and the DSS logging mode are set.

The controlled type of emergencies leads to a voltage drop at 2 150kV BS of Pavlogradskaya-330 Subsection by 113.7kV (76% of U_H) and a current increase on the OHL 150kV L-82 up to 1339 (162% of I_H). The corresponding parameter releases are shown in the panel graphs.

When the software system is running, accidental releases of controlled parameters are registered, corresponding triggers are activated, and transactions are started. The result of the logic output is displayed on the panel shown in Fig. 9.

The DSS cycle results in regulatory-instructional materials of dispatch control obtained from professional ontologies of the knowledgebase as well as recommendations of control corrective actions, their values and directions of change based on calculated sensitivity matrices. In addition, there is a possibility to use the knowledgebase as a database with sampling of necessary data for analysis.

Fig. 9. **Panel of decisions made and knowledgebase sampling**
Source: compiled by the authors

CONCLUSIONS

The paper presents the results of development of mathematical and structural-logical ontology models of products for the automatized emergency dispatching support system of the power system. The developed models are generalized for the classical output machine model. A new interpretation of the trigger model of the DSS components is introduced.

The research results enable the following conclusions:

–there are developed mathematical and structural-logical models of unification of the production ontology of professional knowledge of dispatch emergency control in the power system based on clusters of semantic networks to form inductive logical inference models;

– the formal syntax grammar of production rules, the functional scheme of transaction of one state of the smart system to form the emergency dispatch control knowledgebase is developed;

– on the basis of the automata model of the trigger scheme of the DSS functioning algorithm with multiple entry points into the computational process and a new structure of the DSS kernel in the form of a metarule interpreter, the DSS software package to form the emergency dispatching control of the power system mode is developed and tested.

Further research includes developing diagrams of knowledgebase interpretation and the inductive inference block that functionally overlap, unlike the classical architecture of smart systems. It is suggested to include the ontology selector in the structure of the DSS to select the form of knowledge and data representation online, as well as modules of dynamic formation, validation and replenishment of ontologies.

REFERENCES

1. Grishanov, S. A. "Expert System for Diagnosis of Generators Unit of Thermal Power Station" (in Russian). *Scientific papers DonNTU. Series: "Electrical and power engineering"*. Ukraine. 2013; Vol. 1(14): 83–90. DOI: https://doi.org/10.31474/2074-2630.

2. Osak, A. B., Panasetsky, D. A. & Buzina, E. Ya. "Consumer Operation Mode-Wise Intelligent Emergency Control" (in Russian). *Irkutsk State Technical University Bulletin*. Irkutsk: Russian Federation. 2017; Vol. 21. No. 9: 173–184. DOI: https://doi.org/10.21285/1814-3520-2017-9-173-184.

3. Negnevitsky, M., Voropai, N., Kurbatsky, V., Tomin, N. & Panasetsky, D. "Development of an Intelligent System for Preventing Large-Scale Emergencies in Power Systems". *IEEE Power & Energy Society General Meeting*. 2013. p. 1–5. DOI: https://doi.org/10.1109/PESMG.2013.6672099.

4. Sinchuk, O. N., Kozakevych, I. A. & Vornikov, D. N. "Control System of Wind Generator Based on Switched Reluctance Motor". *Applied Aspects of Information Technology. Publ. Science and Technical*. Odessa: Ukraine. 2019; Vol.2 No.3. 230–242. DOI: https://doi.org/10.15276/aait.03.2019.5.

5. Massel, L. & Kuzmin, V. "Typal Intelligent DSS for Making Strategic Decisions in the Energy Sector and Examples of Application Based on Agent-Service Approach". *Proceedings of the 21st International Workshop on Computer Science and Information Technologies (CSIT 2019). Atlantis Highlights in Computer* Sciences. 2019; Vol. 3: 273–278. DOI: https://doi.org/10.2991/csit-19.2019.48.

6. Nalepa, G. J. "Modeling with Rules Using Semantic Knowledge Engineering". *Intelligent Systems Reference Library. Publ. Springer, Cham. Springer International Publishing AG*. 2018; Vol. 130: 435 p. DOI: https://doi.org/10.1007/978-3-319-66655-6.

7. Wrzalik, A. & Jereb, B. "Use of Expert Systems in Crisis Management". *Proceedings of the conference System Safety: Human - Technical Facility - Environment, CzOTO(2019)*. 2019; Vol.1 Issue 1: 406–411. DOI: https://doi.org/10.2478/czoto-2019-0052.

8. Sinchuk, O. N., Fedotoff, V. A., Somochkyn, Al. B., Kozakevych, I. A. & Somochkyna S. V. "Computer Simulation of Movement and Accurate Positioning of Mining Electric Locomotives Trains when Unloading Cars". *Applied Aspects of Information Technology. Publ. Nauka i Tekhnika*. Odesa: Ukraine. 2020; Vol. 3 No. 3. 165–178. DOI: https://doi.org/10.15276/aait.03.2020.5.

9. Morkun, V. S. & Kotov, I. A. "Intellectualization of Emergency Control of Power Systems on the Basis of Incorporated Ontologies of Knowledge-Bases". *Acta Mechanica ET Automatica*. 2019; Vol. 13 No. 2: 86–94. DOI: https://doi.org/10.2478/ama-2019-0012.

10. Wang, R., Wang, G., Yan, Y., Sabeghi, M., Ming, Z., Allen, J. K. & Mistree, F. "Ontology-Based Representation of Meta-Design in Designing Decision Workflows". *ASME. J. Comput. Inf. Sci. Eng*. 2019; Vol. 19(1): 19 p. DOI: https://doi.org/10.1115/1.4041474.

11. Zhang, J., Zhao, Wu, Xie, G. & Chen, H. "Ontology- Based Knowledge Management System and Application, Procedia Engineering". *Procedia Engineering*. 2011; Vol.15: 1021–1029. DOI: https://doi.org/ 10.1016/j.proeng.2011.08.189.

12. Naykhanova, L. V. "The Technology of Creating Methods for the Automatic Construction of Ontologies Using Genetic and Automatic Programming" (in Russian). *Publ. Buryat Scientific Center. Siberian Branch of the RAS*. Ulan-Ude: 2008. 244 p. ISBN 978-5-7925-0287-1.

13. Antonov, I. V. & Voronov, M. V. "Method of Automated Construction of Domain Ontology" (in Russian). *Data Modeling and Analysis*. 2011; Vol 1 No 1: 116–130. DOI: https://doi.org/10.17759/mda.

14. Hien D. Nguyen, Nhon V. Do, Nha P. Tran, Xuan Hau Pham & Vuong T. Pham. "Some Criteria of the Knowledge Representation Method for an Intelligent Problem Solver in STEM Education". *Applied Computational Intelligence and Soft Computing*. 2020; Vol. 2020: 14 p. DOI: https://doi.org/10.1155/ 2020/9834218.

15. Gavrilova, T. A. & Khoroshevsky, V. F. "Knowledge Bases of Intelligent Systems" (in Russian). *Publ. Piter*. SPb.: Russian Federation. 2000. 384 p. ISBN 5-272-00071-4.

16. Golitsyna, O. L. & Zaitseva, A.V. "Applied Ontologies Formation Based on Subject Area Texts". *KnE Engineering*. 2018; Vol. 3(6): 6–17. DOI: https://doi.org/10.18502/keg.v3i6.2965.

17. Euzenat, J. & Shvaiko, P. "Ontology Matching". *Springer-Verlag Berlin Heidelberg*. 2013. 511 p. DOI: https://doi.org/10.1007/978-3-642-38721-0.

18. Tarus, J. K., Niu, Z. & Mustafa. G. "Knowledge-Based Recommendation: A Review of Ontology-Based Recommender Systems for E-Learning". *Artif Intell Rev*. 2018; Vol. 50: 21–48. DOI: https://doi.org/ 10.1007/s10462-017-9539-5.

19. Kudryavtsev, D. V. "Knowledge Management Systems and Application of Ontologies" (in Russian). *Publ. Polytechnic University*. SPb.: Russian Federation. 2010. 344 p. ISBN 978-5-7422-2982-7.

20. Mika, P. & Akkermans, H. "Towards a New Synthesis of Ontology Technology and Knowledge Management". *Knowledge Engineering Review, Cambridge University Press*. 2004; Vol. 19 No 4: 317–345. DOI: https://doi.org/10.1017/S0269888905000305.

21. Gao, W., Guirao, J.L., Basavanagoud, B. & Wu, J. "Partial Multi-Dividing Ontology Learning Algorithm". *Computer Science. Inf. Sci.* 2018; Vol. 467: 35–58. DOI: https://doi.org/10.1016/j.ins.2018.07.049.

22. Chaplinsky, Yu.P. "Ontological Components of Manager Decision-Making Support" (in Ukrainian). *Scientific papers. NUFT Bulletin.* 2013; No. 48: 65–68.

23. MirHassani, S. A. "Hooshmand F. Models and Mathematical Logic". *Methods and Models in Mathematical Programming. Publ. Springer, Cham.* 2019. 67–113. DOI: https://doi.org/10.1007/978-3-030-27045-2_3.

24. Aristidou, M. "Is Mathematical Logic Really Necessary in Teaching Mathematical Proofs". *American Journal of Education.* 2020; Vol.7 Issue 1: 99–122. DOI: https://doi.org/10.30958/aje.7-1-5.

25. Spitsyn, V. G. & Tsoi, Yu. R. "Intelligent systems" (in Russian). *Tomsk Polytechnic University Publ.* Tomsk: Russian Federation 2012. 176 p. ISBN: 5-98298-354-3.

26. Morkun, V. S. & Kotov, I.A. "Information Technologies for Power Supply Dispatch Control Based On Linguistic Corpus Ontologies". *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu.* Dnipropetrovsk: Ukraine. 2019; Vol. 6(174): 130–136. DOI: https://doi.org/10.29202/nvngu/2019-6/19.

27. Pollak, G. A. "Intelligent Information Systems" (in Russian). *Publ. Center of the South Ural State University.* Chelyabinsk: Russian Federation. 2011. 141 p.

28. Corkill, Dalvi Prathamesh. "Collaborating Software: Blackboard and Multi-Agent Systems & the Future". *In Proceedings of the International Lisp Conference.* New York: 2003. 12 p.

29. Matuszny, M. "Building Decision Trees Based on Production Knowledge as Support in Decision-Making Process". *Production Engineering Archives.* 2020; Vol. 26(2): 36–40. DOI: https://doi.org/10.30657/ pea.2020.26.08.

30. Kang, J. A. "Shortening Matching Time in OPS5 Production Systems". *IEEE Transactions on Software Engineering.* 2004; Vol.30. Issue 7: 448–457. DOI: https://doi.org/10.1109/TSE.2004.32.

31."DS-8 Instruction for the Prevention and Elimination of Technological Disruptions in the Electrical Part of Power Plants and Electrical Grids in the Dnieper ES Region" (in Russian). *Ministry of Fuel and Energy of Ukraine. State Enterprise National Energy Company "Ukrenergo". Dnieper Electric Power System.* Zaporizhzhia: Ukraine. 2008. 67 p.

32. Andreev, E. B., Kutsevich, N. A. & Sinenko, O. V. "SCADA Systems: An Inside View" (in Russian). *Publ. "PTSoft".* Moscow: Russian Federation. 2004. 176 p. ISBN 5-9900271-1-7.

33. Domyshev, A. V., Osak, A. B. & Buzina, E. Ya. "Automation of Control and Management Systems for Electric Power Facilities Based on SCADA-ANARES" (in Russian). *Conference Proceedings: System Research in Energy. ISEM SO RAS.* Irkutsk: Russian Federation. 2005; Issue 35: 230–237.

34. Jackson, P. "Introduction to Expert Systems" (in Russian). *Williams Publ.* 2001. 624 p.

35. Thakar, S. & Nagori V. "Analysis of Rule Based Expert Systems Developed and Implemented for Career Selection". In: Modi, N., Verma, P., Trivedi, B. (eds). *Proceedings of International Conference on Communication and Networks. Advances in Intelligent Systems and Computing. Publ. Springer.* 2017; Vol. 508: 723–731. DOI: https://doi.org/10.1007/978-981-10-2750-5_73.

36. Maylawati, D., Darmalaksana, W. & Ramdhani, M. A. "Systematic Design of Expert System Using Unified Modelling Language". *IOP Conference Series: Materials Science and Engineering. The 2nd Annual Applied Science and Engineering Conference (AASEC 2017) 24 August 2017*. Bandung: Indonesia. 2018; Vol. 288: 7 p. DOI: https://doi.org/10.1088/1757-899X/288/1/012047.

37. Mohammed, A. A., Ambak, K., Mosa, A. & Syamsunur, D. "Expert System in Engineering Transportation: A Review". *Journal of Engineering Science and Technology.* 2019; Vol. 14(1): 229–252.

Онтологія продукційних правил автоматизованої інтелектуальної підтримки протиаварійної диспетчеризації енергосистеми

Володимир Станіславович Моркун[1)]
ORCID: http://orcid.org/0000-0003-1506-9759; morkunv@gmail.com
Ігор Анатолійович Котов[1)]
ORCID: http://orcid.org/0000-0003-2445-6259; rioexito@gmail.com
Олександра Юріївна Сердюк[1)]
ORCID: https://orcid.org/0000-0003-1244-7689; o.serdiuk@i.ua
Ірина Анатоліївна Гапоненко[1)]
ORCID: http://orcid.org/0000-0002-0339-4581; irinagaponenko44@gmail.com
[1)] Криворізький національний університет, вул. Віталія Матусевича. 11. Кривий Ріг, 50027, Україна

АНОТАЦІЯ

Дослідження присвячене вдосконаленню методів і систем керування енергосистемами на основі інтелектуалізації підтримки прийняття диспетчерських рішень. Викладено результати розробки принципової тригерної схеми алгоритму функціонування системи підтримки прийняття рішень. Запропонована модель візуалізації алгоритмів у вигляді тригерної мережі станів комп'ютерної системи, яка забезпечує взаємодію з енергооб'єктами гірничодобувних, металургійних комплексів і регіонів. Введена нова інтерпретація компонентів мережевої тригерної моделі. Модель інтерактивно пов'язана як з діями користувача-оператора, так і з станами компонентів енергосистеми. При цьому стан автоматної моделі пов'язується з виконанням пакета метаправил для управління логічним висновком. Запропоновані нові форми подання алгоритмів управління базами знань, які взаємодіють із зовнішнім середовищем і об'єднують примітиви станів, тригерів і транзакцій операцій і узагальнюють стандартні мови візуалізації алгоритмів. Це дозволяє уніфікувати інтелектуальні системи, які взаємодіють із зовнішнім середовищем. Розроблені моделі подання алгоритмів обробки баз знань, які взаємодіють з енергооб'єктами і уніфікують стани, тригери і транзакції операцій в узагальнені стандартні мови візуалізації алгоритмів. Це дозволяє описувати алгоритми роботи баз даних і їх подієву модель, що забезпечує надійну уніфікацію інтелектуальної системи, яка взаємодіє з об'єктами керування енергосистем гірничодобувних і металургійних комплексів. Розв'язана задача побудови бази знань і програмного комплексу диспетчерської системи підтримки прийняття рішень на основі даних, отриманих в результаті обчислювальних експериментів на схемі енергосистеми. Результати дослідження показали практичну ефективність запропонованих підходів і розроблених моделей.

Ключові слова: Онтологія; інтелект; тригер; алгоритм; продукція; контекст; енергосистема; диспетчер; автоматизація

ABOUT THE AUTHORS

Vladimir S. Morkun – Dr. Sci. (Eng), Prof. Vice-Rector for Research. Kryvyi Rih National University.
11, Vitaliy Matusevych Str. Kryvyi Rih, 50027,Ukraine
ORCID: http://orcid.org/0000-0003-1506-9759; morkunv@gmail.com
Research field: Technological Processes under Uncertainty; Distributed Control of ore Beneficiation Interrelated Processes; Synthesis of Nonlinear Nonstationary Dynamical Systems

Володимир Станіславович Моркун – д-р техніч. наук, професор. Проректор з наукової роботи. Криворізький національний університет, вул. Віталія Матусевича, 11. Кривий Ріг, 50027,Україна

Ihor A. Kotov – PhD. (Eng), Associate Prof. of the Modeling and Software Department.
Kryvyi Rih National University. 11, Vitaliy Matusevych Str. Kryvyi Rih, 50027, Ukraine
ORCID: http://orcid.org/0000-0003-2445-6259; rioexito@gmail.com
Research field: Artificial Intelligence Systems; Knowledge Representation; Analysis and Management of Power System Modes; Databases and Knowledge Bases

Ігор Анатолійович Котов– канд. техніч. наук, доцент каф. Моделювання та програмного забезпечення. Криворізький національний університет, вул. Віталія Матусевича, 11. Кривий Ріг, 50027,Україна

Oleksandra Y. Serdiuk – Assistant of the Automation, Computer Science and Technology Department.
Kryvyi Rih National University. 11, Vitaliy Matusevych Str. Kryvyi Rih, 50027, Ukraine
ORCID: https://orcid.org/0000-0003-1244-7689; o.serdiuk@i.ua
Research field: Energy Efficiency of Traction Electric Drives

Олександра Юріївна Сердюк – асистент кафедри Автоматизації, комп'ютерних наук і технологій. Криворізький національний університет, вул. Віталія Матусевича, 11. Кривий Ріг, 50027,Україна

Iryna A. Haponenko – PhD (Eng.), Senior Researcher of the Science and Research Section.
Kryvyi Rih National University. 11, Vitaliy Matusevych Str. Kryvyi Rih, 50027, Ukraine
ORCID: http://orcid.org/0000-0002-0339-4581; irinagaponenko44@gmail.com
Research field: Effective Technologies of Dust Suppression during Drilling and Blasting Operations

Ірина Анатоліївна Гапоненко – канд. техніч. наук, старший науковий співробітник Відділу науки і досліджень. Криворізький національний університет, вул. Віталія Матусевича, 11. Кривий Ріг, 50027,Україна

184   Information technologies in energy
engineering and manufacturing   ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)