

25. Muller, M. Geoinformationen fur die intelligente stadt – Gute entscheidungen leicht gemacht [Text] / M. Muller, T. Neder // Zeitschrift fur Geodasie, Geoinformation und Landmanagement. – 2015. – Vol. 140, Issue 4. – P. 249–254.
26. Roche, S. Geographic information science II: Less space, more places in smart cities [Text] / S. Roche // Progress in Human Geography. – 2015. – Vol. 40, Issue 4. – P. 565–573. doi: 10.1177/0309132515586296
27. Karpinskii, Yu. O. Formuvannia natsionalnoi infrastruktury prostorovykh danykh – priorytetnyi napriam topografo-heodezychnoi ta kartohrafichnoi diialnosti [Текст] / Yu. O. Karpinskii, A. A. Liashchenko // Visnyk geodezii ta kartografii. – 2001. – Issue 3. – P. 65–74.
28. Duce, S. Microtheories for Spatial Data Infrastructures – Accounting for Diversity of Local Conceptualizations at a Global Level [Text] / S. Duce, K. Janowicz // Lecture Notes in Computer Science. – 2010. – P. 27–41. doi: 10.1007/978-3-642-15300-6\_3
29. Gould, N. From taxonomies to ontologies: formalizing generalization knowledge for on-demand mapping [Text] / N. Gould, W. Mackaness // Cartography and Geographic Information Science. – 2015. – Vol. 43, Issue 3. – P. 208–222. doi: 10.1080/15230406.2015.1072737
30. Kohan, S. S. Development of knowledge base structure of geoinformation monitoring system for evaluation of quality status of agricultural lands [Text] / S. S. Kohan, A. A. Moskalenko // Eastern-European Journal of Enterprise Technologies. – 2015. – Vol. 5, Issue 2 (77). – P. 32–37. doi: 10.15587/1729-4061.2015.51050
31. Oreshchenko, A. Mentalne modeliuвання vyrobnychykh protsesiv: znachennia dlia kartografii [Text] / A. Oreshchenko // Chasopys kartografii. – 2016. – Issue 15 (1). – P. 23–37.
32. Melnyk, L. Transformation processes perception by urban residents of the town of Shpola [Text] / L. Melnyk, S. Batychenko // Chasopys sotsialno-ekonomichnoi heohrafii. – 2016. – Vol. 2, Issue 21. – P. 122–125.
33. Gnatiuk, O. Creation of capable territorial communities in Ukraine: advantages, risks, dangers [Text] / O. Gnatiuk, P. Ostapenko // Ekonomichna ta sotsialna heohrafiia. – 2016. – Issue 76. – P. 73–83.

*Для автоматизації аналізу станів об'єктів використані правила для представлення знань, а для їх відображення, інтерактивної обробки та перевірки вдосконалені моделі правил у вигляді І/АБО-графа. Запропоновано ІТ з використанням редактора правил для розробки знання-орієнтованих систем контролю, за допомогою якої проводиться автоматизоване перетворення правил в мову продукційного програмування*

*Ключові слова: знанняорієнтована технологія контролю, модель правил контролю, І/АБО-граф, експертні системи контролю*

*Для автоматизации анализа состояний объектов использованы правила для представления знаний, а для их отображения, интерактивной обработки и проверки усовершенствованные модели правил в виде И/ИЛИ-графа. Предложено ИТ с использованием редактора правил для разработки знаниеориентированных систем контроля, с помощью которой проводится автоматизированное преобразование правил в язык продукционного программирования*

*Ключевые слова: знаниеориентированная технология контроля, модель правил контроля, И/ИЛИ-граф, экспертные системы контроля*

UDC 004.02 : 004.825 : 004.942

DOI: 10.15587/1729-4061.2017.98727

# DEVELOPMENT OF INFORMATION TECHNOLOGY FOR THE GENERATION AND MAINTENANCE OF KNOWLEDGE-ORIENTED CONTROL SYSTEMS

V. Ruvinskaya

PhD, Professor\*

E-mail: iolnlen@te.net.ua

A. Troynina

PhD, Associate Professor\*

E-mail: anastasiyatroynina@gmail.com

\*Department of system software

Odessa National Polytechnic University  
Shevchenka ave., 1, Odessa, Ukraine, 65044

## 1. Introduction

Present-day requirements to the level of safety for trouble-free operation of objects necessitate improvement of automated control processes. Thus, in addition to monitoring, the control functions include analysis of the object param-

eters to determine its status and provide the decision maker (DM) with processed information.

The paper considers control systems with the following properties:

- limited number of object states;
- parameters can be logical or qualitative;

- the systems under investigation are not real-time systems;
- control conditions may vary.

The listed properties of control systems are characteristic for such object domains (OD) as labor protection, ecology and economics, medicine and sanitation, engineering and others. The control systems under investigation can work both independently and be integrated as a component in existing control systems, particularly, in automated monitoring and management systems.

To make decisions on the state of objects, rules are applied which are usually built into the control system. However, the control conditions for most systems change quite often, and a need of rule adjustment appears in maintaining such systems. Adaptation of traditional systems requires significant costs which leads to a slowdown in making changes and smaller decision validity. In addition, each parameter in many control systems is evaluated separately with no connection to others and this negatively affects validity of decision making concerning results of monitoring.

It is proposed to use knowledge-oriented systems in complex analysis of objects with a possibility of making rapid changes. This is due to the fact that the control task is not completely formalized and the control conditions are usually described declaratively and require an accumulated experts' experience for automation [1].

However, implementation of such systems is constrained on the one hand because of inconvenient mode of formation and updating of knowledge and on the other hand because of availability of only specialized tools for development of control systems, that is, the tools aimed at specific ODs. Consequently, there is a contradiction between the potential capabilities of intellectual knowledge-oriented systems and the insufficient level of their practical use due to imperfect information support tools for maintenance of knowledge bases (KB) and creation of control systems. Therefore, the scientific and technical task of developing new models, methods and IT for creating and maintaining control systems based on a knowledge-oriented approach is urgent and unresolved.

---

## 2. Literature review and problem statement

---

Expert control and monitoring systems for various ODs were analyzed. The issues of choosing implementation options for knowledge-oriented control systems embedded in a data network for improving service quality are elucidated in paper [2]. A knowledge-oriented monitoring, diagnostic and control systems for monitoring technical condition of transformer equipment have been developed in work [3]. In article [4], approaches are presented that realize control of the dynamic state of machines based on vibroacoustic vibrations. In work [5], a knowledge-oriented system for automation of temperature control in a glass furnace is described. An expert system REXPERT [6] is used in training students to control their knowledge. In paper [7], an expert system for monitoring level of driver vigilance is described. Knowledge-oriented systems are known that allow monitoring dangerous situations, particularly on roads [8].

Thus, it was concluded in the analysis of open printed sources devoted to this issue that there are control systems that implement information analysis for various ODs but no generally accepted and tested in practice common approach

to generation of control systems with an integrated expert analysis component was found. Therefore, the work sets the task of developing an IT which will generalize the processes of creating and maintaining knowledge-oriented control systems for various ODs.

Based on the analysis of the knowledge presentation models [9], it was decided to use production rules for control. Such a model favorably differs from others by low complexity, versatility, high modularity, small storage space and visualization abilities [10]. However, despite the similarity of such presentation to natural language presentation, it is necessary to note that this does not always mean that it is easy to formulate rules for use in knowledge-oriented systems. In addition, checking the rules in a text form is difficult if there is a multiplicity of rules. In particular, it is extremely difficult to monitor and verify relationships between rules. Therefore, methods of visualization and, on this basis, methods of rule verification, in particular, when constructing component configurations [11] are offered currently. The authors of work [12] proposed to use for the purpose of visualizing rules and interacting with them the AND/OR-graph which is an oriented circuit-free graph with all vertices divided into three disjoint classes: AND-vertices, OR-vertices and final (target) vertices [1]. For knowledge-oriented *control* systems, one vertex corresponds to the state of the object, and the rest are the values of the object's parameters, and each path represents a set of conditions under which the monitored object state falls outside the acceptable limits.

However, there remains an open question of automatic transfer of the constructed graph representation of rules into the language of production programming. Therefore, the work study is aimed at creating ways of such transformation.

Within the framework of solving problems of validation and verification of production knowledge-oriented systems, the following dominant approaches were distinguished [13]: survey, empirical estimation, static verification, empirical testing.

Two of the approaches listed (survey and empirical estimation) are based on manual, rather than automated, validation and verification procedures [14]. The other two (static verification and empirical testing) are focused on automating validation and verification of systems.

Note that the KB checks are most effective at the early stages when a description of the basic concepts and OD interrelations in a natural language is created because an expert can work with knowledge at this stage. With static verification, so-called anomalies are detected: a static sample in the KB which involves presence of an error in the knowledge presented. Classification of methods of static verification for the knowledge representation model in a form of rules proposed in [15] was most popular. These methods are based on decision tables, generation of labels and graphs. However, most of the existing methods of static verification of rules require additional work from the user, in particular, in the construction of problem-oriented constraints [16]. Thus, for control rules, it is necessary to develop steps for verification of knowledge bases (KB) taking the methods proposed by the authors in [17] as a basis.

---

## 3. The objective and tasks of the study

---

The study objective consisted in raising reliability of control through developing a new IT for generation and

maintenance of control systems based on a knowledge-oriented approach.

To achieve this goal, the following tasks have been accomplished:

- creation of an IT for construction and maintenance of knowledge-oriented control systems;
- transformation of rules from the AND/OR graph into the language of production programming;
- approbation of the developed methods and IT and their study for effectiveness when used in making decisions concerning the control results.

#### 4. Technology of creation and maintenance of knowledge-oriented control systems

Basic information technology for creating knowledge-oriented systems includes the following stages [18]:

- the stage of obtaining knowledge: it includes analysis of the OD, concepts and their interrelations, determining feasibility of using a knowledge-oriented system in the selected area;
- the stage of structuring knowledge: an informal visual description of knowledge about the object domain is developed in the form of a graph, a table, a diagram or a text

reflecting basic concepts and interrelations between the concepts of the object domain;

- the stage of formalization: organization and presentation of knowledge in a form convenient for work of a knowledge-oriented system;
- the stage of implementation: creation of one or several prototypes of a knowledge-oriented system that solve the set tasks;
- the testing stage: the chosen method of representing knowledge of a knowledge-oriented system as a whole is evaluated;
- the maintenance phase: the process of improvement, optimization and correction of defects in the knowledge-oriented system after its commissioning.

An applied information technology for development of control systems is proposed (Table 1) based on the above mentioned basic technology using a tool in the form of a rule editor for creating, verifying and supporting control rules. The technology is based on the knowledge presentation model in a form of production rules as well as on the proposed model and methods for checking quality of control rules [17]. The proposed approach makes it possible to partially move the stage of testing knowledge to the earlier stages of development.

Table 1

IT for creation and maintenance of knowledge-oriented control systems

Basic IT	Applied IT for control systems
<p><i>Stage 1</i> The acquisition of knowledge:</p> <ul style="list-style-type: none"> <li>– analysis of OD;</li> <li>– highlighting concepts;</li> <li>– expediency of developing a knowledge-oriented system</li> </ul>	<ul style="list-style-type: none"> <li>– compiling an OD vocabulary;</li> <li>– identifying parameters of the objects of control;</li> <li>– selection of monitored states;</li> <li>– determination of feasibility of development in accordance with the criteria from the basic IT</li> </ul>
<p><i>Stage 2</i> Structuring:</p> <ul style="list-style-type: none"> <li>– visual representation and decomposition of concepts and relationships in a form of graph, table or text</li> </ul>	<ul style="list-style-type: none"> <li>– construction of a hierarchy of concepts for OD, division into terminal and intermediate ones;</li> <li>– breaking knowledge into groups by the object states, one group for each state is used; with a large volume of the group, its division into subgroups is made;</li> <li>– construction of rules in a natural language and in a form of an AND/OR graph with the ability to convert from one form to another (in the rule editor);</li> <li>– automated checking of rules for consistency, completeness, availability of states (in the rule editor)</li> </ul>
<p><i>Stage 3</i> Formalization of knowledge:</p> <ul style="list-style-type: none"> <li>– representation of knowledge in a form that the inference engine uses</li> </ul>	<ul style="list-style-type: none"> <li>– automatic transformation of each group (subgroup) of rules into the language of production programming taking into account the decrease in the time of logical inference due to the optimal assignment of priorities to the rules</li> </ul>
<p><i>Stage 4</i> Coding</p>	<ul style="list-style-type: none"> <li>– choice of an existing inference engine or development of an own one;</li> <li>– completion of each group (subgroup) in accordance with the rules received at the stage of formalization;</li> <li>– combining of groups (subgroups) of rules into a common modular structure taking into account frequency of errors and triggering in two modes:                             <ol style="list-style-type: none"> <li>1) until the first discrepancy;</li> <li>2) a complete check;</li> </ol> </li> <li>– integration of KB and inference engines with two components of the control system;</li> <li>– the first one: for setting modes and obtaining parameters from the monitored object;</li> <li>– the second one: for outputting work results and organizing graphical user interface</li> </ul>
<p><i>Stage 5</i> Testing</p>	<ul style="list-style-type: none"> <li>– debugging each group of rules: all conditions for emergency states as well as conditions for normal operation;</li> <li>– debugging the modular structure obtained by combining groups (subgroups) of rules;</li> <li>– integration testing of the control system</li> </ul>
<p><i>Stage 6</i> Maintenance</p>	<ul style="list-style-type: none"> <li>– changing the rules of control (in the rule editor)</li> </ul>
Steps 1–6 are performed iteratively until the system meets the user's requirements	

*Stage 1.* Obtaining knowledge on the object domain of control.

1. 1. Automated compilation of the object domain dictionary using tools. To create a quality dictionary, it is necessary to choose a set of texts that would contain a maximum number of terms specific for the given object domain. However, in the most cases, the texts contain a large number of service words and terms from other or related object domains [19], so it is expedient to invite an expert in the given object domain to compile the dictionary. Such a dictionary helps to avoid a large number of errors at all subsequent stages when building a knowledge structure, especially if connecting it and the possibilities of working with it to the rule creation tool. The expert can view the terms of the dictionary, select necessary ones and add them to the rules. Such a dictionary of the object domain helps to avoid a large number of errors when drawing up rules of the expert system and it reduces time of their development as well.

In work [20], a method is proposed for automated creation of dictionary of the object domain terms which is based on syntactic and semantic analysis of texts. On the basis of this method, a freeware program was created which implemented dictionary compilation using the mentioned method. The program makes it possible to view highlighted words and phrases, the number of their occurrences, frequency and also write the results to an .xml file.

1. 2. Definition of the input data, that is, the system parameters and their possible values. Moreover, the values are used later as the premise of the rules.

1. 3. Identification of the monitored values of the parameters for which control is carried out, as well as the conditions for their fall beyond the permissible limits.

1. 4. Determination of appropriateness of using a knowledge-oriented system for control in a certain OD.

*Stage 2.* Structuring knowledge for control.

2. 1. Construction of a conceptual knowledge structure in a form of a parameter hierarchy for control and their values. Breakdown of concepts, that is, parameters and their values for each group for selection of the so-called finite (terminal) from the analysis of which the conclusion is immediately made about falling of the monitored parameter values beyond the permissible limits and intermediate ones by help of which final conclusion can be drawn in several steps.

2. 2. Breakdown of knowledge into groups when each group corresponds to one monitored state of the object. Thus, when constructing rules, there will be rules in each group describing one situation for which control is performed. In turn, if the group is large, then it can be divided into subgroups of knowledge sets independent or weakly dependent from each other. This modularity makes it visually easier to create and maintain KBs.

2. 3. Construction of the functional structure of the object domain of control modeling the expert's reasoning in a form of rules for each group (subgroup) in natural language in a text form or as an AND/OR graph with the ability to automatically switch from one type to another and preserve/download rules. To do this, an automated tool like a rule editor is used. The editor enables interactive work with the rules presented in a graphical form and their automatic conversion to Boolean expressions for verification.

2. 4. Automated rule checking:

– consistency check of each rule as well as the entire set of rules using the Boolean formula satisfiability task (SAT). This defines rules that can never be executed or will always

be executed. Both the first and the second cases are inadmissible and show contradictions in the system of rules;

– checking the rules for incompleteness based on the automatic construction and visualization of “inverse” rules describing in which cases the monitored state of the subject is normal;

– checking reachability of the object state vertex from the set of rules.

*Stage 3.* Formalization of knowledge for control.

3. 1. Automatic transformation of the rules represented in a form of an AND/OR graph to a textual form in natural language with the help of the rule editor. If necessary, scanning the KB by the developer, correction in a text form and reverse automatic conversion to the AND/OR graph. Thus, the rules can be “seen” from one more point of view.

3. 2. Transformation of the rules represented in a form of an AND/OR graph into the production programming language for a particular platform.

*Stage 4.* Implementation of knowledge-oriented control systems.

4. 1. Choosing a ready inference engine or developing an own one, setting the inference engine.

4. 2. Programming of each group (subgroup) of the rules in accordance with the rules obtained during the formalization stage 3. 2.

4. 3. Combining groups (subgroups) of rules into a common modular structure for the possibility of their actuation in a preset sequence.

The order of execution of the modules is assigned based on the frequency of error appearance (the group for which violations are most frequently detected during control is started first).

To ensure an optimal combination of the knowledge-oriented system modules, various modes of module actuation were identified and two combination modes were proposed on their basis. In the *first mode* (fast mode), the decision that the monitored state of an object is beyond the tolerance limits is made according to the first group of rules which were violated and the modules are not started further. The second mode of operation of a knowledge-oriented system involves actuation of all modules which makes it possible to identify all the problems.

4. 4. Integration of the KB and the inference engine with other components of the control system by building them into existing control system by adding two components to them:

– the first one is for setting the modes and obtaining parameters from the controlled object;

– the second one is for outputting the work results and organizing graphical user interface.

*Stage 5.* Debugging knowledge-oriented control systems.

5. 1. Debugging for each group of rules. As many tests are performed as there are rules in the group; thus all conditions of exceeding the limits of the investigated object state are tested. Next, tests for checking cases when the state of the object does not go beyond the permissible limits are performed.

5. 2. Debugging the general modular structure of the rules. Each control rule and all their possible combinations are tested.

5. 3. Integration testing of the control system.

*Stage 6.* First of all, maintenance includes changing of the control rules. It is supposed to make prompt changes to the rules, preferably by the expert (in the rule editor).

Some subitems of stages 1 to 6 can be performed in different sequences, and often iteratively. It is convenient, for example, to the knowledge engineer together with the expert to imagine first what rule groups can be, then define the entities and states for each group and finally construct the rules. Further, rules must be checked and when discrepancies and mistakes are corrected, return to the previous steps is made.

In the practice of creating knowledge-oriented systems, the sequence of steps can be somewhat changed. For example, first build the rules, next select parameters and their values from them simultaneously correcting the rules and then the rules are divided into groups and checked.

As a result of browsing the visualized graph, the knowledge engineer along with the expert can review the rules, identify inconsistencies and logical errors in the earlier built rules, parameters and their values, and fix them.

Similarly, stages 4 and 5 are performed iteratively, like it is done when developing any software systems.

The proposed IT is designed *in corpore* for creation of KBs analyzing precise parameters of the controlled object. However, if parameters are fuzzy [21], then most of the IT stages are executed in full. There are tools for coding knowledge-oriented systems with unfuzzy rules, in particular Clips as well as FuzzyClips for fuzzy rules and the control rules obtained in stage 2 are converted into production languages for these systems.

### 5. Transformation of rules from the AND/OR graph into the language of production programming

For formalization of knowledge, freely distributed development environment CLIPS (C Language Integrated Production System) was chosen [9].

The facts in CLIPS are a piece of data placed in the current list of system facts (working memory). The rules consist of premises and conclusions. Premises of the rules are satisfied depending on the presence or absence of some specified facts in the list of facts. Conditional elements in a form of patterns consist of a list of field constraints, wildcards and variables that are used to find a set of facts that correspond to a given pattern. A rule effect is represented by a set of actions to be performed in a case when the rule is applicable to the current situation. Priorities (salience) can be assigned to rules.

According to the proposed IT, each group (subgroup) of rules generated in the rule editor at the structuring stage is automatically transformed into the CLIPS production programming language at stage 3.2 taking into account reduction of the logical conclusion time due to the optimal assignment of priorities to the rules.

To reduce the number of rules actuated during the control, higher priorities are assigned to the rules based of the following considerations:

- the rules containing premises common to many rules must be fulfilled earlier, since such premises are the most significant in the control and thus, if they are executed, it is more likely that some rule will be actuated and the control will end;
- the rules that have a smaller number of premises should be fulfilled earlier, since there will be fewer checks in this case.

A template of the module pre-designed for the control rules has been formed, the same for any control rules and consisting of control rules designed to output the result,

functions necessary to obtain input data and the structures necessary for storing and processing facts. It is possible to create both a console and a graphical user interface. And then rules from the specific AND/OR graph are added to the template module.

For example, Fig. 1 presents rules of the ‘Network’ subgroup of the ‘Problems in the computer network’ object domain as a graph. It can be seen that there are 4 input parameters on the left: traffic (normal or low), number of queries (high or low), query processing time (high or low), CPU load (high or low). On the right, there is the verdict whether there are any problems in the network. Between the input parameters and the verdict, intermediate conclusions are given.

Fig. 2 shows the levels of subgroup rules, since they are represented in the resulting program in CLIPS. Each previous level of nesting depends only on the following. The first level: the rules that determine values of the input parameters, level 4: terminal level, levels 2 and 3: intermediate levels.

Consider the stages of rule transformation.

The description of the AND/OR graph for the rule editor is stored in an XML file. The first step is to select names of the parameters and their possible values from the file. The received information automatically generates the first level rules which determine the values of the input data for control.

An example of a CLIPS rule that defines the query processing time is shown below:

```
(defrule dop-three
(not (av (a query_time) (v ?)
(not (result ?))
=>
(if (yes-or-no-p “Is query processing time high? (yes/no) “)
then
(assert (av (a query_time) (v High)))
else
(assert (av (a query_time) (v Normal))))))
```

If there is still no data in the facts database on the query processing time and there are no results of inference, the question is asked to the user and depending on the answer, a fact is established. The yes-or-no-p function is used from the previously prepared template. Of course, the time for processing queries in the knowledge-oriented system built into the network management system, is determined automatically and sent to the input of the rule base without asking the user. There may be a more complicated case when there are not two but more answers. In this case, another function is used from the template.

*The next step* is formation of intermediate rules, there may be several of them: Fig. 2 shows two levels with two and three rules at each, respectively. *The last step* is creation of a terminal rule in the conclusions of which contain the state of the controlled object, e. g., the rule of the following type:

```
(defrule final1
(or (av (a connection_instability) (v Connection_instability))
(av (a network_instability) (v Network_instability))
(not (result ?))
=>
(assert (av (a state) (v Problem in network)))
(assert (result “Problem in network”)))
```



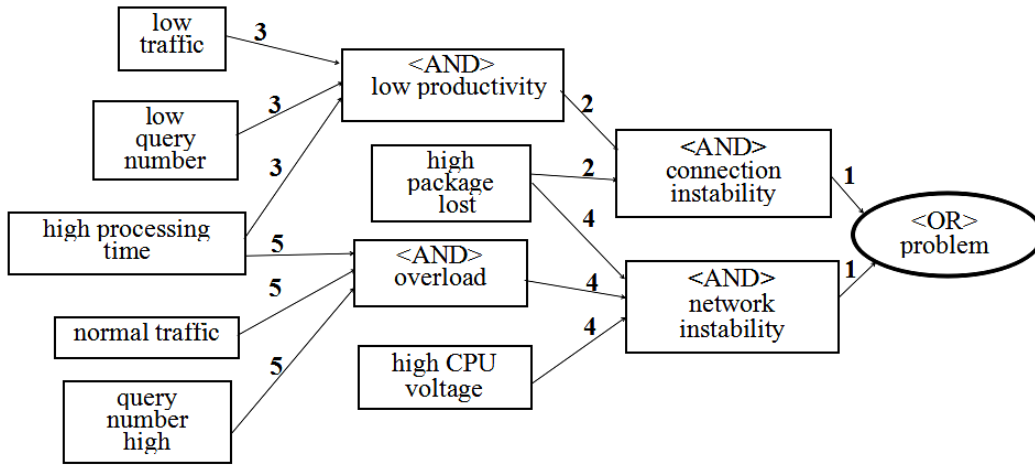


Fig. 1. AND/OR graph for the 'Network' subgroup of rules

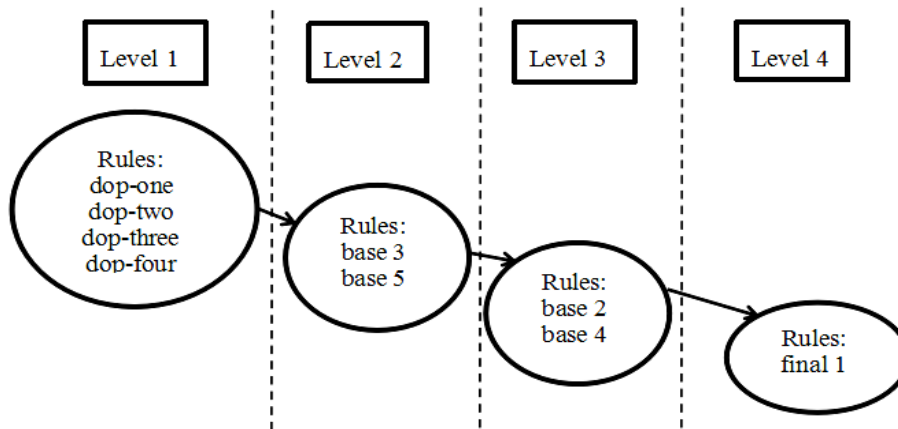


Fig. 2. Levels of rules for the 'Network' subgroup

By the example of network control given in Fig. 3, all branches are shown for which the 'Network' subgroup was tested.

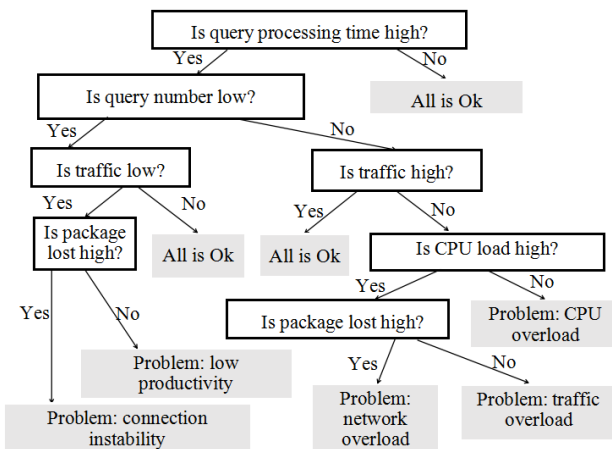


Fig. 3. The decision tree used for testing ES

Fig. 4 shows a fragment of the consultation when the user must answer a number of questions after which the consultation result will be displayed.

Based on the proposed rule editor and IT, it is possible to create a knowledge-oriented system for user consulting with the use of a graphical interface without writing a single line of code.

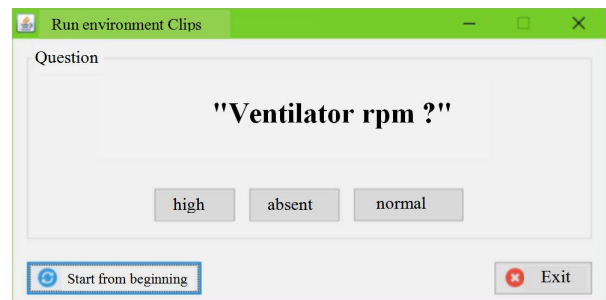


Fig. 4. User interface

## 6. Discussion of study results

The presented work is a continuation of the research related to the development of models and methods for creating and maintaining knowledge-oriented control systems that formed the basis of the information technology proposed here.

The work was carried out within the framework of the state budgetary themes of the Ministry of Education and Science of Ukraine, scientific research of the Department of System Software of the National Polytechnic University (ONPU), Odesa, Ukraine, titled "Methods, Models and Means for Analyzing and Improving Performance of Computer Systems" No. DR 0110U0008193.

On the basis of the IT and the rule editor, effective prototypes of ES control were created.

ES for safe work with electrical installations is used in making decisions by the dispatcher about the possibility of carrying out works with electrical installations. Data about the parameters of objects are entered manually by the dispatcher or automatic control means are used that measure room temperature, humidity and other parameters. The ES determines whether the safety requirements are met and, in the event of violations, notifies the dispatcher.

When the system was tested by an expert, the dispatcher's work log was analyzed two months before the system was first used and for two months during the system operation (Fig. 5–8). The expert looked through the dispatcher's decisions in the reports and identified how many mistakes

of the first kind were made when the dispatcher made the wrong decisions that the work cannot be performed, and mistakes of the second kind when the dispatcher made wrong decisions that works could be done. Validity of correct decisions was also calculated as a probability measure determining the ratio of correct decisions to the total number of decisions (before the system was introduced and with its application, Fig. 9).

Validity of correct dispatcher's decisions before applying the ES was 0.85; validity of correct decisions within two months after the ES application was 0.97 (Fig. 9). Thus, the validity of the dispatcher's decisions increased by 12 %:  $(0,97/0,85-1)\times 100=12$ .

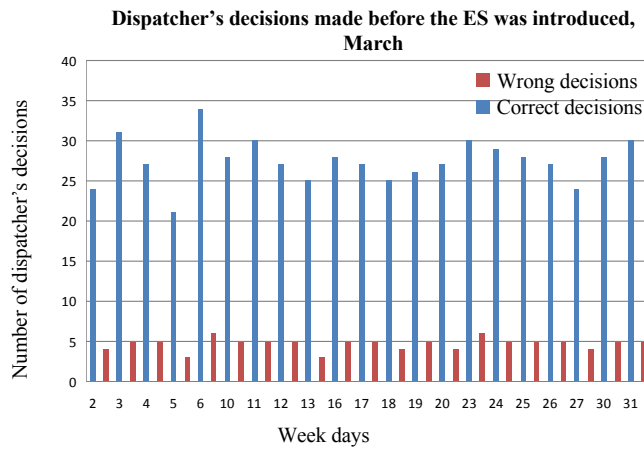


Fig. 5. Dispatcher's decisions made before the ES was introduced, March

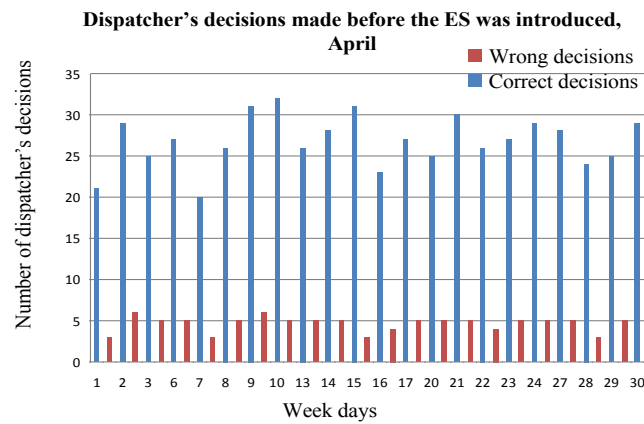


Fig. 6. Dispatcher's decisions made before the ES was introduced, April

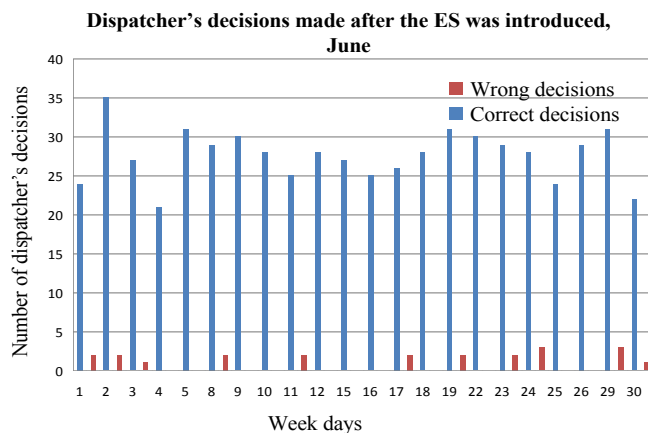


Fig. 7. Dispatcher's decisions made after the ES was introduced, June

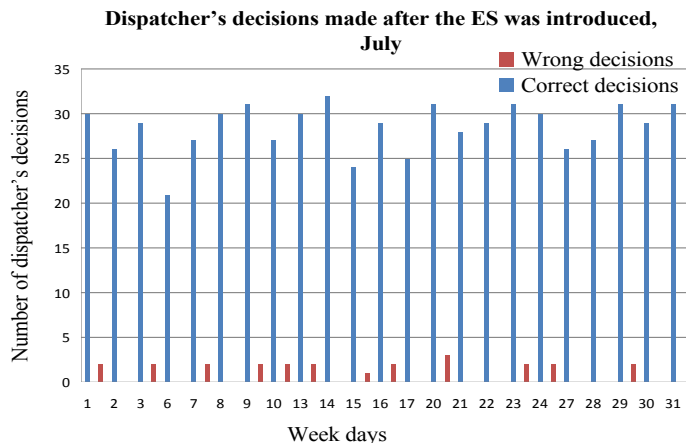


Fig. 8. Dispatcher's decisions made after the ES was introduced, July

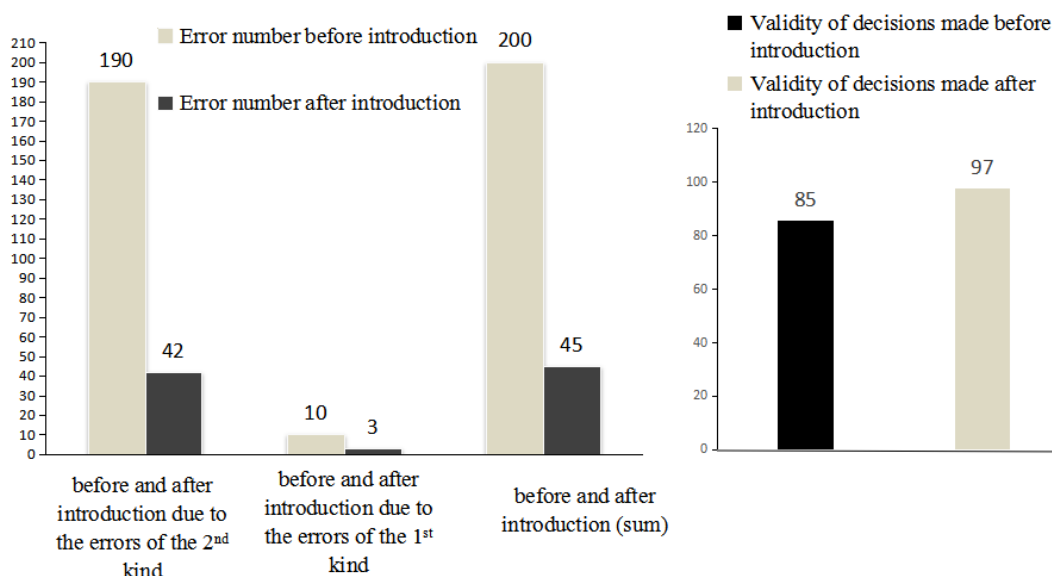


Fig. 9. Validity of decisions made before and after using the ES

IT is universal in nature and is supposed to be applicable for a wide range of object domains described earlier, therefore further research areas involve analysis of features and development of knowledge-oriented control systems for various applications: not only for decision-making but also for training.

### 7. Conclusions

The work is devoted to the development of an IT for generation and support of knowledge-oriented control systems. The development process was greatly simplified due to the visualization of knowledge and its automated verification.

1. The editor of control rules has been developed to provide the main stages of rule creation which allows visualization of the rules, interactively process and automatically

check them. On its basis, an IT was proposed for development and maintenance of knowledge-oriented control systems.

2. As one of the main stages of IT, an automatic transformation of each group (subgroup) of rules created by the rule editor into the language of production programming was suggested. At the same time, logical inference time was reduced due to the optimal assignment of priorities to the rules. This approach makes it possible to create and verify rules at an early stage which ensures rapid change of rules.

3. On the basis of this IT, working prototypes of ES have been developed. The control system for safe operation of electrical installations includes 5 subgroups and 54 rules. When using ES for safe operation of electrical installations, validity of dispatcher's decisions has increased by 12 % compared to the period when the dispatcher made decisions without its help.

### References

1. Dzharatanno, D. Ekspertnye sistemy: principy razrabotki i programmirovaniya [Text] / D. Dzharatanno, G. Rajli. – 4-oe izd. – Moscow: OOO «I. D. Vil'yams», 2007. – 1152 p.
2. Yazlovetskiy, Ya. S. Sravnitelnyiy analiz ekspertnyih sistem kontrolya kachestva obsluzhivaniya v setyah peredachi dannyih [Text] / Ya. S. Yazlovetskiy, L. N. Velichko // Vesnik svyazi. – 2015. – Issue 2 (130). – P. 49–54.



3. Ekspertnaya sistema monitoringa, diagnostiki i upravleniya transformatornym oborudovaniem «ESMDU-TRANS» [Electronic resource]. – Available at: [http://www.ztr.com.ua/files/ztr\\_d69-esmdy-trans--2014.pdf](http://www.ztr.com.ua/files/ztr_d69-esmdy-trans--2014.pdf)
4. Harisov, A. R. Ekspertnaya sistema upravleniya ekspluatatsionnoy nadezhnostyu metallurgicheskogo oborudovaniya Bogoslovskogo aluminievogo zavoda [Text] / A. R. Harisov, V. I. Kuzyakin // Nauchnye trudy XI otchetnoy konferentsii molodykh uchenykh GOU VPO UGTU – UPI. – Ekaterinburg, 2007. – P. 38–39.
5. Kutsenko, V. P. Mikrovolnovaya ekspertnaya sistema kontrolya temperatury v steklovarennoy pechi [Text] / V. P. Kutsenko // Artificial Intelligence. – 2014. – Issue 3. – P. 155–161.
6. Zaytseva, T. V. Primenenie ekspertnoy sistemy kontrolya znaniy «REXPert» v uchebnom protsesse [Text] / T. V. Zayceva, N. N. Smorodina, N. V. Vasina // Nauchnye vedomosti Belgorodskogo gosudarstvennogo universiteta. – 2013. – Vol. 28, Issue 22-1 (165). – P. 231–235.
7. Sigari, M.-H. A Driver Face Monitoring System for Fatigue and Distraction Detection [Text] / M.-H. Sigari, M. Fathy, M. Soryani // International Journal of Vehicular Technology. – 2013. – Vol. 2013. – P. 1–11. doi: 10.1155/2013/263983
8. McCarthy, J. D. An Integrated GIS-Expert System Framework for Live Hazard Monitoring and Detection [Text] / J. D. McCarthy, P. A. Graniero, S. M. Rozic // Sensors. – 2008. – Vol. 8, Issue 2. – P. 830–846. doi: 10.3390/s8020830
9. Havrylova, T. A. Bazy znaniy yntellektualnykh system [Text] / T. A. Havrylova, V. F. Khoroshevskiy. – Sankt-Peterburg: Pyter, 2005. – 384 p.
10. Yin, W. Standard model of knowledge representation [Text] / W. Yin // Frontiers of Mechanical Engineering. – 2016. – Vol. 11, Issue 3. – P. 275–288. doi: 10.1007/s11465-016-0372-3
11. Tidstam, A. Development of Industrial Visualization Tools for Validation of Vehicle Configuration Rules [Text] / A. Tidstam, L.-O. Bligard, F. Ekstedt, A. Voronov, A. Knut, J. Malmqvist // Proceedings of 9th International Symposium on Tools and Methods of Competitive Engineering. – Karlsruhe, 2012. – P. 305–318.
12. Troynina, A. S. Redaktor znan dlya ekspertnykh system monitorynhu [Text] / A. S. Troynina, V. M. Ruvinska, M. S. Nikolenko // Vestnik Hersonskogo natsionalnogo tehniceskogo universiteta (HNTU). – 2013. – Issue 1 (46). – P. 183–185.
13. Yalovets, A. L. Predstavlenie i obrabotka znaniy s tochki zreniya matematicheskogo modelirovaniya. Problemy i resheniya [Text] / A. L. Yalovets. – Kyiv: Naukova dumka NAN Ukrainyi, 2011. – 339 p.
14. Meseguer, P. Assessing the Role of Formal Specifications in Verification and Validation of Knowledge-Based Systems [Text] / P. Meseguer, A. D. Preece // Achieving Quality in Software. – 1996. – P. 317–328. doi: 10.1007/978-0-387-34869-8\_26
15. Preece, A. D. Foundation and application of knowledge base verification [Text] / A. D. Preece, R. Shinghal // International Journal of Intelligent Systems. – 1994. – Vol. 9, Issue 8. – P. 683–701. doi: 10.1002/int.4550090804
16. Solihin, W. Classification of rules for automated BIM rule checking development [Text] / W. Solihin, C. Eastman // Automation in Construction. – 2015. – Vol. 53. – P. 69–82. doi: 10.1016/j.autcon.2015.03.003
17. Ruvinskaya V. M. Rules of expert system for safety monitoring: checking on completeness and consistency [Text] / V. M. Ruvinskaya, A. S. Troynina, E. L. Berkovich, O. O. Bilovzorov // Odes'kiy Politehnichnyi Universytet. Pratsi. – 2015. – Issue 2. – P. 103–110. doi: 10.15276/opu.2.46.2015.19
18. Troynina, A. S. Expert system for monitoring of computer network work rules checking automation [Text] / A. S. Troynina, V. M. Ruvinska, E. L. Berkovich, A. Yu. Chernenko // Elektrotekhnycheskiye y kompyuternyye systemy. – 2014. – Issue 14 (90). – P. 94–104.
19. Lyashevskaya, O. N. Chastotnyiy slovar sovremennogo russkogo yazyika na materialah Natsionalnogo korpusa russkogo yazyika [Text] / O. N. Lyashevskaya, S. A. Sharov. – Moscow: Izdatelskiy tsentr «Azbukovnik», 2009. – 1087 p.
20. Kungurcev, A. B. Method of creation of domains dictionaries for extraction of the facts from texts in the natural language [Text] / A. B. Kungurcev, S. N. Borodavkin, A. P. Golub // Eastern-European Journal of Enterprise Technologies. – 2010. – Vol. 1, Issue 4 (43). – P. 32–36. – Available at: <http://journals.uran.ua/eejet/article/view/2550/2355>
21. Leonenkov, A. B. Nechetkoe modelirovanie v srede Matlab i fuzzy Tech [Text] / A. B. Leonenkov. – Sankt-Peterburg: BHV – Peterburg, 2005. – 736 p.