International Review of Automatic Control (I.RE.A.CO.), Vol. 11, N. 6
ISSN 1974-6059 November 2018

Wi!Mi Expert System Shell as the Novel Tool for Building Knowledge-Based Systems with Linear Computational Complexity

Oleg O. Varlamov

Abstract – *Expert systems (ES) are very effective tools for solving various complex problems, which usually require human intelligence. In fact, these systems emulate a human expert's decision-making process. However, development of various ESs from scratch is a tedious and costly process. This article describes a novel ES building tool. This tool belongs to the class of the so-called Wi!Mi ES shells–programs that allow significantly simplify and accelerate the ES development process. It uses a new knowledge representation model, which is based on the bipartite graphs as well as on the significantly improved but simple inference engine. This tool makes it possible to process large systems involving millions of variables during several seconds. This article presents both theoretical basis and inner workings of the Wi!Mi tool, as well as several examples of various subject domains. Finally, this tool is evaluated by testing on the prototype including up to 3 million variables and rules.* ***Copyright © 2018 Praise Worthy Prize S.r.l. - All rights reserved.***

Keywords: *MIVAR, MIVAR Net, Logical Inference, Computational Complexity, Artificial Intelligence, Intelligent Systems, Expert Systems, General Problem Solver, Knowledge Representation, Expert System Shells, Inference Mechanism, Bipartite Graphs*

I. Introduction

At the very beginning of researches in the artificial intelligence sphere (during the 1950s), scientists have faced the "dimensional curse", when the problems of logical inference and automatic construction of algorithms were solved by a complete search having NP-complete computational complexity, for example, in the study of the General Problem Solver [1].

This restriction has hindered development of the expert systems, as well as has forced researchers to develop various heuristics in order to decrease quantity of options for a full-text search. To their great regret, any "settings" and "certain sorting" of the rules made it possible to solve only limited tasks, and in general "fell into the same trap of dimensional curse", when the logical conclusion could take many hours, years or more to solve simple problems. In general, logical inference belongs to a class of sequential tasks that cannot be run in parallel on several processors or cores. Therefore, complete search of just 15 rules with the help of the state-of-the-art computers requires more than 10 seconds, therefore, there are no possibilities to solve any real-time problem [2]. This article is devoted to the brand-new shell, which is called Wi!Mi and purpose of which is to ensure development of various expert systems. The article is focused on the analysis of the knowledge representation technology, which is used in this system. It also describes new and unique inference engine, which allows solving any given problem (if it can be suitably formulated) in linear time.

Practical use of these new technologies can essentially simplify creation of the ESs in various subject areas and decrease quantity of experts, who are involved in development of these systems. However, most importantly, these technologies significantly reduce the period of time, which is required for development of the solution algorithm for the ready-to-use ES.

II. Literature Review

As a common example of a good model problem, the subject area "Triangles" from school geometry was used during the 1980s [3]. Within this area, main objects (variables) are defined as side lengths, angles, radii, etc., in total 72 variables, as well as all the basic equations and dependencies (rules) including formulas for the sum of the angles of a triangle, area, perimeter, etc., summing up to, 237 unique rules for the tasks of the 8th grade of the Soviet school. This problem, which can be easily solved by schoolchildren, will have a complexity of "classics" 237! (factorial), which is "beyond the power" of the modern "pre-MIVAR" ES. Therefore, NP-complexity hampers development of the entire sphere of AI. Various mathematical methods (full search with heuristics, Rete-networks) and software products (CLIPS [4], Prolog [5], Nexpert Object[5], Exsys Corvid [6], Jess [7]) do not allow solving many practically significant tasks. As concerns these shells and methods, "Triangles" turned out to be really "beyond their power". In 2002, a theoretical solution with a linear computational

https://doi.org/10.15866/ireaco.v11i6.15855

complexity of the logical inference based on the MIVAR networks was proposed [8]. This solution is based on the transition from Petri productions and networks to the bipartite oriented MIVAR networks, which are "schematically" shown in the diagram (Fig. 1). The P versus NP ("P=NP") problem [2] is the most important but open question within the complexity theory. There are no doubts that this problem is of great importance for various spheres of knowledge but it could not be solved for more than 40 years. It belongs to the so-called "the millennium problems" (officially known as "the Millennium Prize Problems"). It is believed that if this problem will be solved, this will mean that it would be theoretically possible to solve many complex tasks much faster than it is possible now. There exist many problems, which are connected with the NP-complete class. The logical inference problem belongs to the set of those problems. The transition from the NP-complexity of inference search (using the "IF-THEN" productions) to the linear computational complexity of the MIVAR method was accomplished and published in 2002 [9].

Since 2002, the scientific sphere steadily continues to evolve conducting research, publishing articles and reports [10]-[14], defending dissertations [15]-[16].

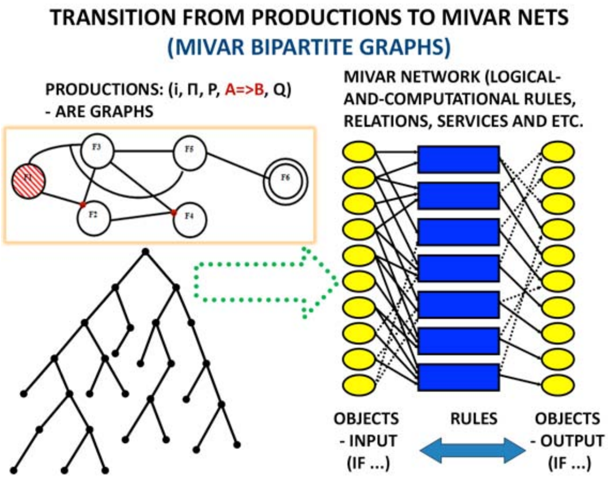

Fig. 1. Productions versus MIVAR nets

Thus, a linear complexity solution was proposed for the NP-complete problem, and, hence, since 2002 the logical inference on productions has ceased to be a complete-enumeration task [11]. This means only that one of the NP-complete tasks has passed to the class of linear problems, but all the remaining NP-complete problems have remained in this class. The cartoon https://www.youtube.com/watch?v=KxjOYyefJdU shows a qualitative comparison of three generations of the ES: a full search, a set of pre-resolved tasks in the database and the MIVAR Razumator (the Reasoner in Russian). In Russia, studies on the use of the MIVAR networks for various directions in the AI sphere, which are schematically shown in Fig. 2, have been conducted.

For example, it has been shown that neural networks could be included in the logical inference path as separate rules in accordance with the principle "input - black box of a neural network - output". There are several useful results of the joint use of the MIVAR networks and neural networks, which make it possible to understand meaning of the images published in Russia, but these applications go far beyond the scope of this article and they will be analysed in detail in future publications.

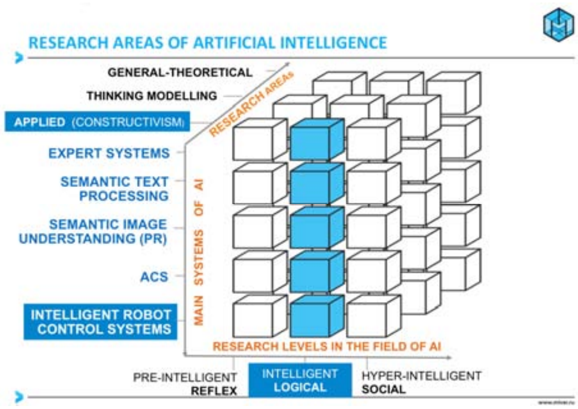

Fig. 2. Structural scheme of the AI sphere and positioning of the MIVAR technology

Representatives of the well-known Watson project of IBM have conducted preliminary consultations with authors and some details of the Watson project are known to the authors of this paper. In particular, it is known that this project combines various methods and mechanisms in the AI sphere. Total quantity of these methods exceeds 30 methods and this quantity is constantly growing. There is no direct analogy to Wi!Mi software in any part of IBM Watson at present time, but in the long term our product can be included in its composition as a separate cloud service in order to create the MIVAR knowledge bases and various ES on the subject areas of interest [17]. This, again, obviously goes beyond the scope of the scientific article, but from the scientific point of view, the MIVAR networks can complement well the IBM Watson product. It is worth to note that the main advantage of the MIVAR approach is the linear computational complexity of the logical inference on productions. Until 2002, this task was considered as the NP-complete, i.e. dramatic reduction in the complexity of solving problems of logical inference and automatic construction of algorithms has been achieved. The MIVAR networks and use of the Wi!Mi Razumator have been tested in practice and it was proved that they are efficient in the sphere of ES (solving "Triangles" problems in real time and ES "Analysis of accidents").

At present time, it is used for creation of the virtual consultants (intelligent chat bots) for the Russian banks, for logical situational management of groups of robots [12]-[13], [18]-[19], as well as for decision-making in unmanned vehicles [20] and for analysis of traffic rules violations [14]. For example, Fig. 3 highlights the MIVAR model of the ES "Analysis of road accidents" (courtesy of Dmitry Chuvikov).

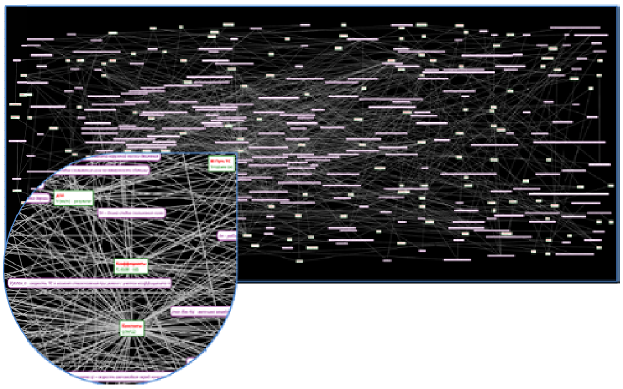

Fig. 3. MIVAR model of "Analysis of road accidents" subject area (Created by Chuvikov D.A.)

The MIVAR virtual consultants belong to the category of "question-answering systems", which are sometimes called as the "chat bots". They are based on the new approach, which uses the expert systems for creation of knowledge bases ("models of the world") and presentation of knowledge in the form of the multivariate MIVAR multigraphs. At present, the software platform Tel!Mi is developed, on the basis of which several Virtual Consultants were implemented in order to understand meaning of the Russian-language texts in natural language. For example, there is a prototype of the "Living Book", which allows users to communicate with an artwork, as an independent interlocutor, who answers many questions within the meaning of the work. A Virtual Consultant for the Russian Bank (Alfa-Bank) was created which improved the speed of answers to questions from employees of Alfa-Bank (http://futurebanking.ru/post/3565 - in Russian, the press release of Alfa Bank) by a factor of 50. Neural networks work as the "black box" model and they are based on the statistical methods of the decision-making process, which is quite typical for the reflex level in the AI sphere. On the other side, the MIVAR ESs are connected with the logical level of research in the AI sphere and they are based on the "white box" model, where the question "why this decision was made"? can always be answered.

Algorithm of the solution for each subject area (briefly considering, for example, road traffic situation) and the model "Given initial data input –Find more data", which is built into the MIVAR ES, ensure detailed logical justification for the decision. The MIVAR systems make decisions based on the cause-effect relationships, which are expressed in the form of the Traffic Laws (TL), while the neural network approaches are based on the statistics and events that have already occurred. Both these approaches complement each other pretty well: neural networks are good for assessing the traffic situation in vision systems, while logical rules of the ES are necessary for compliance with the TL. Researches with the purpose of development of the MIVAR systems for unmanned vehicles are the longer-term project but practical experiments in January 2017 on the streets of Moscow showed that they have practical feasibility. The diagram of an unmanned vehicle, which was created under the leadership of Sergey Shadrin [21], has shown (Fig. 4 and Fig. 5) the layout of the control levels.

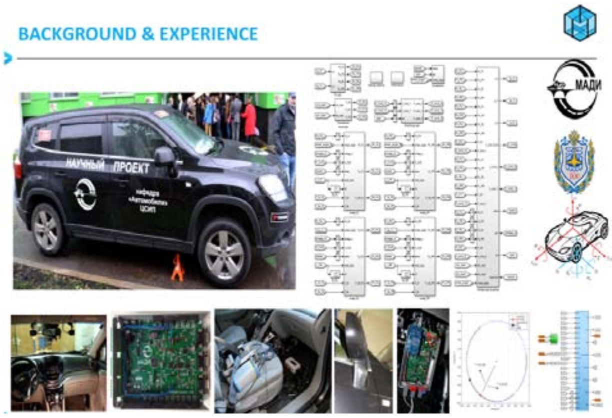

Fig. 4. Prototype of the unmanned vehicle (under supervision of Sergey Shadrin)

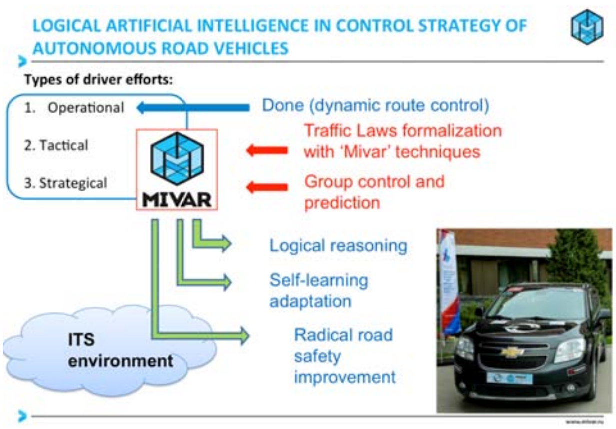

Fig. 5.Layout of the control levels for the unmanned vehicle and usage of the MIVAR ES

The MIVAR Razumator (the Reasoner in Russian) underlies the decision-making system (DMS) while the Wi!Mi is used in order to create a knowledge model. The DMS differs from the ES because of the DMS includes two additional subsystems. Firstly, it includes the subsystem for data input and automatic start of logical inference procedures and, secondly, it includes the subsystem of automatic transmission of the received values of variables (objects) to the actors of the robot or the unmanned vehicle. In addition, this new MIVAR DMS was created for the "situational management" of the movement of the autonomous robot group in order to ensure that the route of each robot is built along the input video stream in the real-time mode (automatically and adaptively with the help of the usual laptop) taking into account the appearance of various obstacles, as well as, moreover, taking into account the properties of these obstacles, e.g.: passable and retractable obstacles and the obstacles that must be circumvented [12]-[13].

References to the videos of our practical experiments are duly presented at the following web sites: https://www.youtube.com/watch?v=o_0K_fdCaBk and https://www.youtube.com/watch?v=p2pY7YXCM9A.

Several publications on this topic are already prepared. Researches in the sphere of the MIVAR networks and capabilities of these networks continue, therefore, of course, many factors are still unknown.

However, it is obvious that transition to the qualitatively new level in the AI sphere has been already accomplished due to the drastic reduction in the complexity of the logical inference, which was reduced from the NP-complete complexity to the linear complexity [8]. Several words should be used in respect of the inner structure of the Expert Systems (ES) in order to highlight some key areas of interest. The Expert Systems (ES) are the bright and rapidly progressing area in the sphere of the artificial intelligence (AI) [2].Capability of the ES to solve problems from various areas of the human activity in a verifiable way is the main reason for interest in further development and implementation of such systems. It is unlikely that there exists any problem area, in respect of which proper attempts to create the ES have not been made [22]. The ES is the set of the application programmes or the entire software, which performs functions of the human expert through solving problems in the area of his/her competence [23]. The ES operates in the course of its work with the knowledge, which was provided to it by the relevant human expert. The knowledge of a subject area required for the ES is formalized in a certain way and this knowledge is represented in the computer memory in the form of the knowledge base that can be changed and supplemented in the course of further system development. The ES is capable to: give advices; analyse; perform classifications, and make predictions.

They focus on solving the problems, for which involvement of a human expert is usually required. Since the ES make decisions in the same way like people do, the ES must store, similar to human brain, a huge amount of facts and rules for the later use. In addition, the ES must provide the decision-making mechanisms, which allow to drop the unnecessary rules, which are not connected with the problem that must be solved at any given point. Therefore, all typical ESs usually consist of the modules that are presented in Fig. 6 [1]. The user interacts with the system through the relevant user interface, which includes many complex factors and mechanisms, including the internal structure of the knowledge database, as well as the used inference engine.

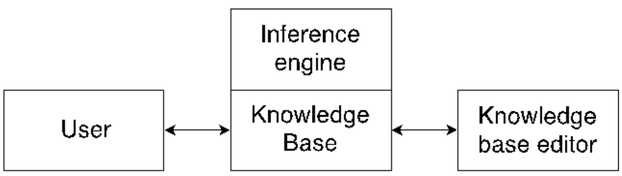

Fig. 6. Main structure of an expert system

A key role in this structure (Fig. 6) is played by the ES knowledge base and a suitably chosen way of this knowledge representation, as well as the selected inference engine. Inference engine allows the system to develop algorithms in order to obtain a problem solution in the relevant subject domain. Creating and bringing the ES into production in the early days have required much time and money. The main breakthrough in the ES development was connected with implementation of the instrumental shells, which enabled essential reduction in the development and setup time. The most part of the currently existing instrumental shells is presented by the general-purpose shells. The most popular ones include the classic universal systems such as CLIPS [4], the Jess [24], Nexpert Object [7], Exsys Corvid [6], Prolog [5] and several other systems. On the other hand, there is a new generation shell (G2[25]), which has constricted resources for the knowledge representation and the data output and which is capable to adapt itself to the specific problems within given domain. It is very difficult to perform the detailed comparative analysis of these shells (because of it would be quite subjective analysis and it is, in general, outside the scope of this article),while some interesting data and comparisons can be found in [26].

However, it is necessary to note that the main common disadvantage of all these systems is the mechanism of the inference engine (this mechanism is difficult for understanding). In principle, the implemented algorithms carry out multiple call-backs to certain nodes, involve complicated bookkeeping of the rules positioning in the database, and may combine several different algorithms in the course of traversing through the decision trees (this statement applies to the following systems: CLIPS, Prolog, Nexpert Objects, Exsys Corvid, Jess, and G2). In particular, the Rete-algorithm, which is implemented in the Jess, requires a lot of physical memory, and it may significantly slow down operation of the entire system, if the "wrong" order of rules accidentally occurs in the knowledge database [24].

III. Theoretical Basis of the State-of-the-Art Expert System Shell. MIVAR Model of the Knowledge Representation

The most important aspect of the knowledge-based systems is connected with the chosen model of the knowledge representation [28]. The knowledge representation (KR) is one of the fundamental problems of the AI, which is difficult for developers to this day - "the KR research is aimed at providing those theoretical foundations, with the help of which useful, comprehensible, and reliable systems can be build"[27].

Currently, there are many approaches to the knowledge representation, including predicate description, semantic networks, production rules, neural networks, evolutionary, agent-oriented, stochastic, and many others [1]-[2], [28]. As a rule, all these approaches are designed in such a manner in order to ensure achievement of the reasonable compromise between the efficiency and the representation expressiveness.

Moreover, if the ES "from scratch" would be developed, one model or another one of representation will be chosen. When using the ES shells, such choice is often missing, whereas the representation of knowledge in the shells plays an important role. The Wi!Mi shell uses its own MIVAR model of the knowledge representation, which was developed by the author. The basis of the MIVAR approach to the knowledge representation is adaptive and dynamic description of the modelled domain. The key concept of the MIVAR approach to the knowledge representation is the notion of the MIVAR network [29]. This network provides formalization and representation of the human knowledge in the course of development of the expert systems with the help of the Wi!Mi shell. The *MIVAR* network is the method for representation of the information part of the MIVAR space (objects and rules for their processing) as the bipartite directed graph, consisting of the objects *P* and the rules *R* (Fig. 7).Taken together, these objects and rules form the domain model [12]-[13].

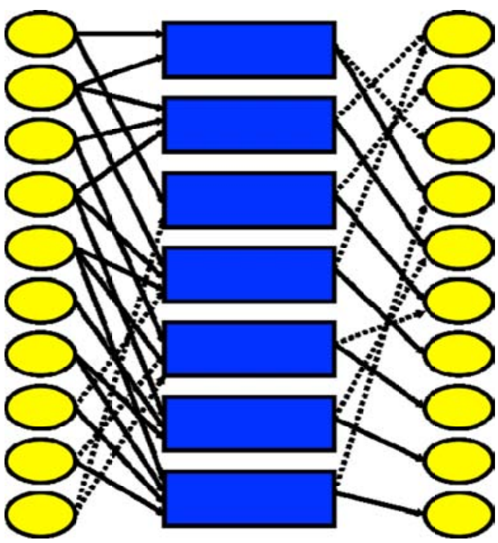

Fig. 7. Bipartite graph of the MIVAR network. Objects are represented by yellow ovals and rules are represented by blue rectangles

The MIVAR network has the following key features:

1. The network consists of two distinct types of elements: the objects (*P*) and the rules (*R*);
2. As concerns each variable, relevant information concerning all the rules *R*, for which it is either the input variable *X* or the output variable *Y*, is clearly indicated;
3. As concerns each rule *R*, information concerning all its input and output variables *P* is stored, including the information on the quantity of the input *X* and output *Y* variables;
4. All information is stored in the database layer, which is specifically adapted for work with the MIVAR networks;
5. As concerns each object, of the MIVAR network, all related objects and rules are clearly identified. As concerns the MIVAR nodes within the MIVAR network, all possible transitions to and from these nodes are known, thus eliminating the need for exhaustive look up in the course of searching the inference path in the MIVAR network.

The MIVAR bipartite graph network can be represented graphically as the two-dimensional matrix of size $M \times N$, [12]-[13], where *N* is the quantity of parameters (objects) within the subject area, while *M* is the quantity of the rules that link the objects of the subject area (Table I). Within each row of this Table, all input parameters are marked with the symbol *X*, while all output parameters are marked with the symbol *Y*.

TABLE I
THE MIVAR NETWORK MATRIX

Rules \ **Parameters**	1	2	3	4	5	…	*N-2*	*N-1*	*N*
1	X	X							Y
2			X		Y			X	X
…									
M		X		X	X		Y	Y	

Various subject areas can be described with the help of the MIVAR networks. As a simple example, it is possible to consider the subject areas "Geometry. Triangles" and "Movement of robots". The review of the domain "Geometry.Triangles" will begin. Upon entering to the knowledge base, the data from the simplest geometry textbook, the network of 200 rules and 532 parameters can be formed and represented as the200 × 532 matrix. A part of this MIVAR network is shown in Table II. For example, the above network can be used as the basis for construction of the ES-based geometry tutor.

The next example of the MIVAR network from the subject area "Movement of robots" is going to be analyzed. It is supposed that the map of the area, within which the robots will move, is already known. This area is associated with the table, within which rows and columns are marked with letters *A*, *B*, *C*, ... and so on.

Then it will be possible to create the MIVAR network of movements (Table III).

The MIVAR network is constructed by connecting the sets belonging to two different types according to the following rules: "object-rule" and "rule-object". The relationships like "object-object" and "rule-rule" are prohibited. In general, the relationship has the following form: "object(s)-rule-object(s)". The first specified element is the element *from* which comes the corresponding relationship. The second element is the element, within which relevant relationship exists. Due to this arrangement, the graph becomes oriented and this fact eliminates the possibility of misinterpretation or incorrect transformation of the objects because of the occasional backtrack passage on the relationship. The ES of this structure is inherently scalable since it is possible, at any point in time, to add new sets of elements of any type, without having to change their processing methods.

In addition, in order to describe such MIVAR network, in many cases it is possible to type in the currently existing objects and relationships (rules).

TABLE II
GEOMETRY. TRIANGLES. A PART OF MIVAR NETWORK

Rules \ Parameters	Side *AB*	Side *BC*	Side *AC*	Perimeter *P*	Area *T*	…	Height *BH*	Median *AM*	Bisection *AL*
The perimeter of the triangle (with the help of the lengths of three sides)	X	X	X	Y					
The area *T* of the triangle with the help of the perimeter and lengths of the sides	X	X	X	X	Y				
…									
The height *BH* with the help of the triangle area and the length of the side *AC*			X				Y		
The length of the median *AM*		X						Y	

TABLE III
MOVEMENT OF ROBOTS. THE SIMPLEST EXAMPLE.
A PART OF THE MIVAR NETWORK

Rules \ Parameters	*AA*	*AB*	*BA*	*BB*	*CA*	*AC*	…	*BC*	*CB*
Go to the right (*AA*)	X	Y							
Go forward (*AA*)	X		Y						
Go to the right (*BB*)				X				Y	
Go forward (*BB*)				X					Y
Go to the left (*BB*)			Y	X					
Go backward (*BB*)		Y		X					
…									
Go to the right (*BA*)			X	Y					
Go forward (*AB*)		X		Y					
Go forward (*BA*)			X		Y				

IV. The MIVAR Inference Method

The above-described MIVAR knowledge representation provides the possibility to construct the algorithm (method), which allows building the scheme for the solution of various problems, which have occurred within the subject domains that are to be analysed. With the help of the above representation of the domain knowledge, it is possible to construct the algorithm, which allows searching for the information within the MIVAR network. The MIVAR network data model allows determining the open and hidden relationships between objects within the network, as well as building algorithms for calculation of the inference path for any given task within the relevant subject area.

The proposed method is the corner stone of the inference mechanism, which is used within the Wi!Mi software environment. The essence of this method is connected with the fact that proper matrix is being built for any network of the rules, which are presented in a tabulated form. It is possible to identify the inference path with the help of analysis of this matrix or, otherwise, it is possible to find out that the inference path does not exist. In the cases, where several possible paths exist, then it is necessary to choose the best (the shortest) path based on some optimality criteria.

Fig. 8 shows a schematic description of a preferred implementation of the method for construction of the inference path with the help of the proposed MIVAR approach.

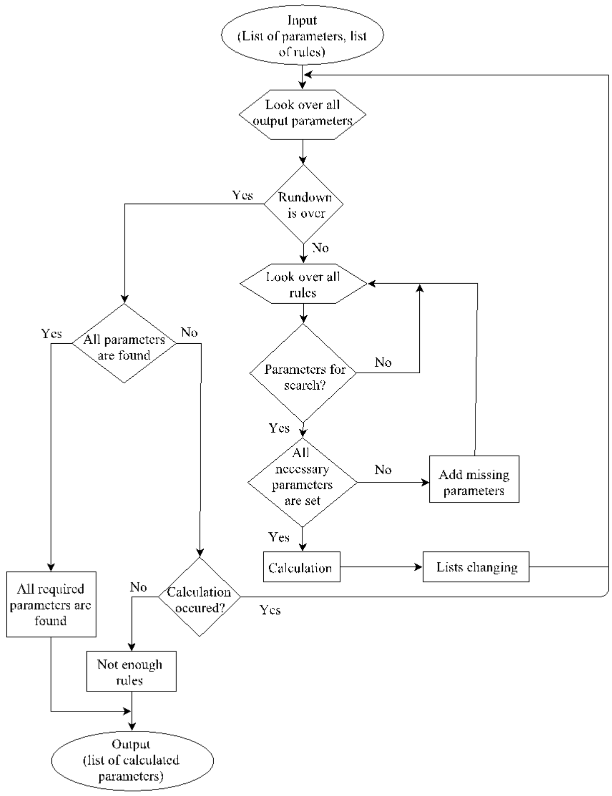

Fig. 8. Flowchart showing the method for construction of inference path

An example of the algorithm derivation for solving a new problem with the help of the MIVAR inference mechanism will be provided, considering the task of managing behaviour of a multi-agent robotic system. It requires employment of two types of robots, namely robots-cooks and robots-waiters in order to organise operation of a small canteen. It is also necessary to assign proper roles, such as preparation of the incoming order or delivery of the prepared order (Fig. 9).

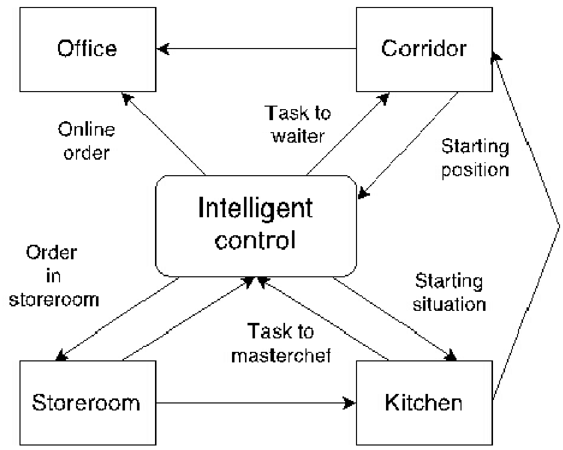

Fig. 9. Scheme of the robotic canteen

The management group level is centralised; quantity of the input parameters is limited, the movement map (map with the static obstacles) is known, and there is no time limit for preparation of the order. Algorithm of operation is not determined from the very beginning, however, it is constructed on the basis of the order parameters and other initial conditions. Presence, location and other parameters of the state for all objects are determined in the beginning of calculations. Variations in robot operations depend on the above-mentioned initial conditions. With the purpose of execution of the order, it is necessary to assign one unoccupied robot-chef. If the lack of objects is detected in the kitchen, then an order is formed for the delivery of the necessary object from the warehouse. Certain section of the MIVAR network, which arises in the course of analysis of this subject area, is presented in Table IV. Supposing that green tea should be made, knowing that there is a water source, a cup, a green tea in the closet and, in addition, an empty kettle. In the first stage, the known data in the extended MIVAR matrix can be noted (Table V). As it can be seen, the rule, "Pour water into the teapot" can be activated during the first step. Result of this step will be appearance of a cold teapot. Then, it is possible to perform the following operation - "Boil the teapot". Appropriate operations are presented in Table VI. Acting in a similar way the following sequence of operations can be obtained: *"Pour-water into the teapot" ->"Boil the A teapot" ->"Get the tea leaves out of the closet" ->"Make tea"*. In the course of derivation of this algorithm of operations, some solutions may include unwanted superfluous steps. In most cases, presence of such extra steps is not critical for the calculation time. However, in order to ensure further acceleration of calculations, the inference mechanism implemented in the Wi!Mi is supplemented by a specialized block, which eliminates all unnecessary calculation steps (for this purpose, each detected parameter contains information about which rule was launched to define this parameter, and, on the other hand, each rule is supplemented with the information about "parent" parameter).

V. Wi!Mi Shell

The Wi!Mi shell described in this article is the tool for development of the ES in various areas. There are examples of the knowledge represented in the form of productions. There are some production constructions. The general form is as follows:

$$i, \Pi, \mathrm{P}, \mathrm{A} => \mathrm{B}, Q$$

A=>B is an ordinary production "*if…, else…*",which is called the production core. *P* characterizes the external condition or applicability conditions of the production determined by the factors, which are not included in *A*.

TABLE IV
EXAMPLE. ROBOTIC CAFÉ. SECTION OF THE MIVAR NETWORK

Rules \ Parameters	Empty teapot	Cold Tea pot	Hot Tea pot	Water	Green tea in the cup board	Green tea on the table	…	Cup	No green tea	Green tea order
Pour water into the teapot	X	Y		X						
Boil the teapot		X	Y							
…										
Make tea			X			X		X		Y
Get the tea leaves out of the closet					X	Y			X	

TABLE V
EXAMPLE. ROBOTIC CAFÉ. RESULTS OF THE FIRST STAGE.IN THE *M + 1* SERVICE LINE, KNOWN PARAMETERS ARE MARKED AS *Z*, AND ONES THAT NEED TO BE COMPUTED - AS *W*

Rules \ Parameters	Empty teapot	Cold teapot	Hot teapot	Water	Green tea in the cupboard	Green tea on the table	…	Cup	No green tea	Green tea order	*Service column*
Pour water into the teapot	X	Y		X							
Boil the teapot		X	Y								
…											
Make tea			X			X		X		Y	
Get the tea leaves out of the closet					X	Y			X		
Service row	Z			Z	Z			Z	Z	W	

TABLE VI
EXAMPLE. ROBOTIC CAFÉ. RESULTS OF THE SECOND AND THIRD STAGES

Rules \ Parameters	Empty teapot	Cold teapot	Hot teapot	Water	Green tea in the cupboard	Green tea on the table	…	Cup	No green tea	Green tea order	*Service column*
Pour water into the teapot	X	Y		X							✓✓
Boil the teapot		X	Y								✓✓
…											
Make tea			X			X		X		Y	
Get the tea leaves out of the closet					X	Y			X		
Service row	Z	Z	Z	Z	Z			Z	Z	W	

Condition P allows to choose the required productions from all productions with A in the left part of the core. Π characterizes the sphere of the subject domain of the knowledge base, or the pre-conditions of the production applicability. These pre-conditions are not different from P, however they form the formal system, within the frame of which the logic reasoning will be drawn. Q characterizes the post-conditions of the production indicating the changes that need to be brought into the base of knowledge and the system of productions after implementation of this production. The MIVAR nets can be represented as the bipartite graph that consists of the objects-variables and the rules-procedures. First of all, two lists forming two non-intersecting parts of the graph are being prepared: the list of the objects and the list of the rules. There are three basic stages in the course of the MIVAR data processing:

1) Creation of the MIVAR matrix in order to ensure description of the subject domain;
2) Processing the matrix in order to construct (design) the algorithm, which ensures solution of the required problem;
3) Execution of all computations according to the algorithm, which was constructed at the previous step.

The first stage can be seen as the formalization of the subject domain in the form of productions with the following transition to the MIVAR rules:

"input objects – rules/procedures/services – output objects "

Currently, it is the most difficult stage that requires participation of the human specialist (expert) for creation of the MIVAR model of the subject domain. At the second stage, the automated construction of the algorithm and logic inference is carried out. The input data is represented in the form of the MIVAR matrix reflecting description of the subject domain, which is specified with the help of the input objects-variables ("GIVEN"), as well as the required objects-variables ("TO BE FOUND"). At the third stage, the obtained algorithm is executed and the actual calculation is being done. To summarize, the matrix for the MIVAR net of the represented logic rules is to be constructed as the list.

Then, based on the analysis of this matrix, the fact of existence of the successful path of the logic inference is being established. At the last stage, it is necessary to choose the shortest path (or the most optimal path on the basis of the given criteria of optimality). The system that contains m rules and n variables is now defined. Then the matrix $V(m \times n)$ (each row of which corresponds to a rule and contains information about the variables used in this rule) contains all the relations about the rules and the variables. In each row, all the input variables of the rule on the corresponding positions are marked with x, all output – with y, all the variables that have already obtained some concrete value in the process of logic inference – with z, all required (output) – with w. In addition, one more row and one more column are used to keep service information. There is the matrix $V(m+1)\times(n+1)$ that contains the structure of the net of rules. This structure can be changed at any moment of time, so it is changeable, or evolutional. An example of such a matrix is shown on Fig. 10.

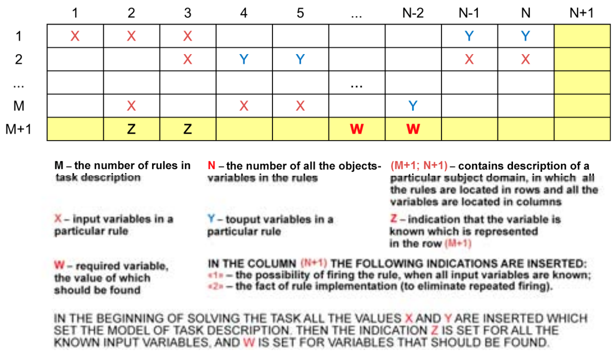

Fig. 10. The initial matrix V of (m+1)×(n+1) with the structure of the net of rules

An example of the work of the method (Fig. 11) is described.

The following operations have been implemented to search the path of the logic inference.

1. In the row $(m+1)$ all known variables are marked with z, all required – with w. For example, on the Fig. 11 positions 1, 2, 3 are marked with z, position $(n\text{-}2)$ – with w. Simultaneously, Fig. 11 shows the same operation with the help of the bipartite graph of the MIVAR net.
2. The search for the rules that can be activated (i.e. the rules with all known input variables) is implemented upside down. If there are no suitable rules, then the path of the logic inference does not exist and the request for additional data should be sent. If some rules exist that can be fired, a corresponding service row is marked. For example, it is possible to mark them with a special sign "1" (Fig. 11, cell (1, n+1)). Fig. 11 shows this operation on the bipartite graph.
3. If several rules exist that can be activated, the choice of the rule, which has to be activated at first, is implemented following predetermined criteria. Several rules can be processed at the same time provided that necessary resources are available.
4. The imitation of the launch of the rule (procedure) is being implemented by the assignment of the value "known" to the output variables of this rule (z in our example). Launched rule is being additionally marked with the sign "2" (not necessary). An example is shown on Fig. 11.
5. If in the service row any required/sought variables left are not determined (signed with "w"), the search for the path of logic inference must be continued. If there are no more sought variables, the problem is considered as the successfully solved one. All launched rules composing the path of the logic inference are now in the order they were launched.

Fig. 11 shows only one launched rule, but there are still required (undefined) variables, so it is necessary to proceed to the next step.

6. After finding new values on the previous step, one has to check the existence of new rules, which could be launched. If there are no such rules, the path of logic inference does not exist and the following operations are the same as in the step 2. If suitable rules are available, then the search for the path has to be continued. In our example, such rules do exist (see Fig. 11). The cell (2, n+1) is marked with 1, in order to underline the possibility of the lunch of corresponding rule. Fig. 11 shows the same situation on the bipartite graph.
7. The next step is similar to step 4. Then, following steps 5 and 6 all the operations are repeated until the result is obtained. If it is necessary, all the steps from 2 to 7 can be repeated until the result is obtained. The result may be positive – the path of logic inference exists, – and negative – there is no such path. The execution example is continued gradually.
8. In the cells (m+1, 4) and (m+1, 5), the sign that variables 4 and 5 are deductible is obtained. The cell (2, n+1) is marked with 2, as the rule was launched. After that, it is clear that not all required variables are known. Therefore, the processing of the matrix $V(m+1)\times(n+1)$ has to be continued. Analysis shows that the rule m can be launched (Fig. 11).
9. After the launch of the rule m, new values are obtained (Fig. 11).
10. There are no more required rules in the service row. There are new values in the cells of the matrix: 2 in the cell (m, n+1) and z instead of w in the cell (m+1, n-2). So, the positive result is obtained, the path of logic inference exists.

It has been shown that bipartite graphs of the MIVAR nets can be used for creation of expert systems with the help of productions as the main basis. From scientific point of view, the main difficulty in with the help of expert systems is conceptual description of subject domain in terms of productions and the formation of two necessary lists: objects and rules for the MIVAR logic nets.

The processing itself is rather easy and a universal mechanism is described here. Rules-procedures can be represented as productions corresponding to traditional approach. The universal possibilities of the MIVAR approach are caused by the fact that it unifies all known data models, including the model "entity-relation", Petri nets, semantic models, and ontology. The advantages of the MIVAR approach are:

1) lineal computational complexity and real-time calculations;
2) ability to solve logic and/or computational (and other) tasks;
3) input data flow control and data-driven search;
4) adaptive description and ongoing problem solving;
5) active work with the requests or redeterminations of input data.

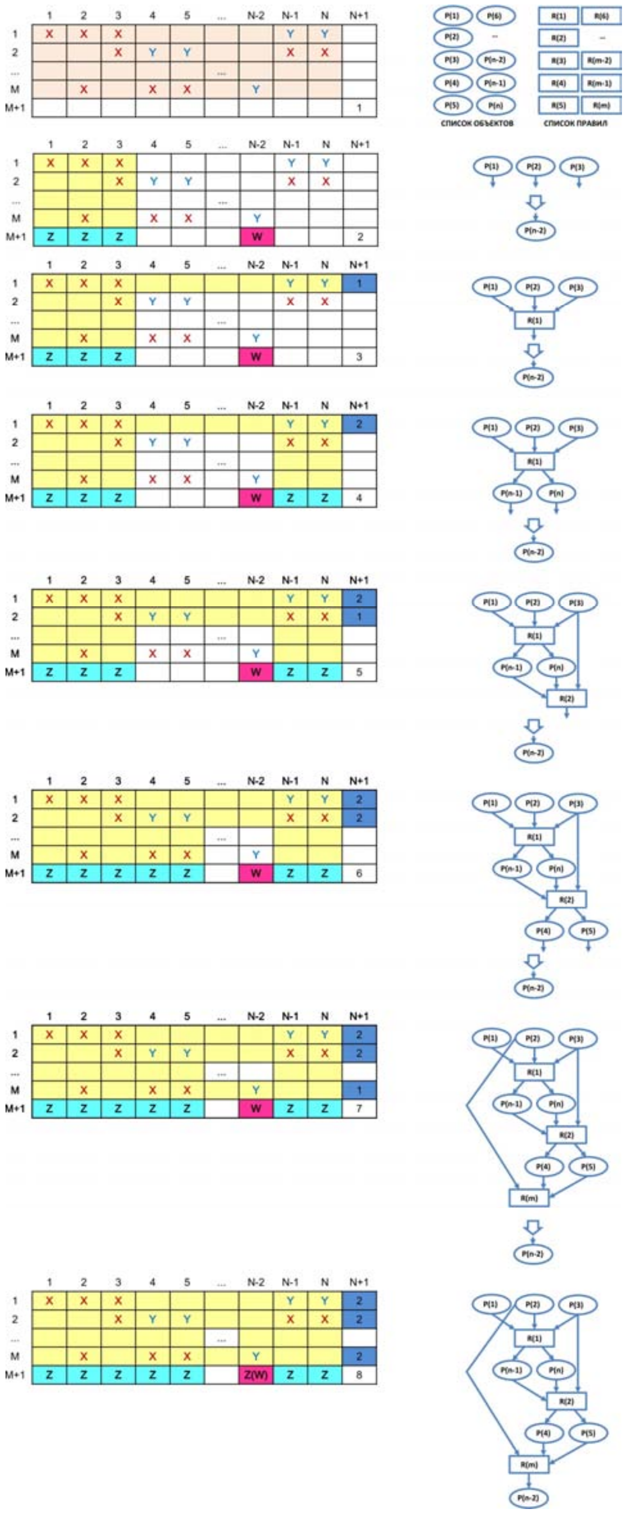

Fig. 11. An example of matrix processing for solving the problem

The technology that is built into the Wi!Mi allows to calculate the required parameters from the original ones by finding appropriate computing algorithms. This may involve calculations of all necessary intermediate parameters. During the calculations the Wi!Mi attempts to assign values only to the parameters required for the active rule, thus avoiding unnecessary calculation of full universe of parameters. With this approach, a high throughput and performance of the program can be achieved even with limited computational resources. In addition, when constructing an algorithm for a particular problem, the previously calculated values of the intermediate parameters can be reused. The Wi!Mi program provides the following features:

1) Create and edit descriptions of subject area models:
   - domain objects (parameters and classes);
   - relations and rules linking these objects;
2) Assessment and validation of input data, including data completeness checks;
3) Data analysis;
4) Review of resulting inference algorithm;

The big advantage of the Wi!Mi shell is its simplicity.

It requires from the users to have only basic subject domain knowledge to be able to use the program. Day-to-day work with the system does not require specialist skills. The simple and transparent definitions are being used to describe the relationships between objects.

Nevertheless, it is possible to take advantage of more complex relationships that can be written in JavaScript. Model creation and further interoperation with the Wi!Mi shell is done via the Wi!Mi interface (Fig. 12) with the help of the following stages:

1. Build the MIVAR network for the subject area and parameterise the model via objects.
2. Create a model and establish links between the domain objects, which is achieved in Wi!Mi by:

- assigning relations (e.g., $y = ax$)
- defining set of rules;
- specifying the conditions of input data correctness (setting appropriate limits or constraints on the data).

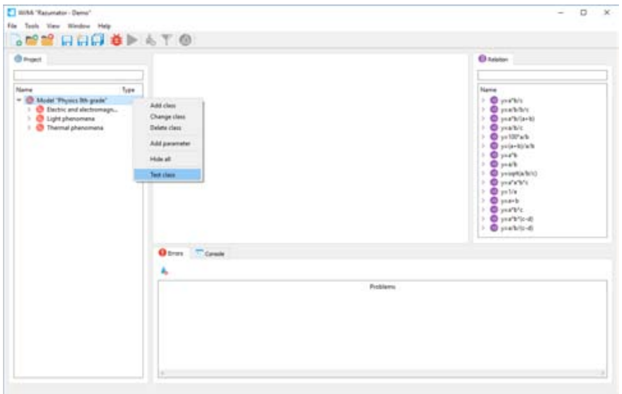

Fig. 12. The Wi!Mi interface

Besides the built-in model tools for testing the model, the user interface provides convenient way to interact with the core of the ES by allowing the use of the third-party interface. To date there are very limited quantity of expert shells present on the market. Majority of them can only work on a single platform, or worse, the speed and resource usage efficiency is lagging behind. There are no such issues for Wi!Mi suite described in this article. It is cross-platform and has extremely fast and efficient algorithm for inference path calculation, irrespective of the type of the tasks being used.

VI. Computational Complexity of the MIVAR Method

A theoretical estimation of the MIVAR method computational complexity is performed. The total quantity of operations used in the MIVAR method is determined by the sum of the operations at each stage:

1) assignment of known z and required w to the cells of the service row (m+1) (total quantity of operations no more than n);
2) assignment of the sign of the processing of rules in the service column (n+1) (total quantity no more than 2m, but can be no more than m);
3) assignment of the sign that the variable is known (z) to the cells of the service row (m+1) (total quantity no more than n);
4) assignment of the new values to the cells of the row (m+1) (no more than n operations).

Operations on the steps 1,3 and 4 are accomplished over the row (m+1). The total quantity of operations is not more than the total quantity of the cells in this row, as already processed values are "erased" and not processed again. In the whole, the total quantity of operations (KD) in the MIVAR method, i.e. its computational complexity does not overwhelm the quantity of cells in service parts of the matrix:

$$O(m+n) \text{, i.e. } KD \le (m+n)$$

In the worst (complicated) case when the proposed reduction of the computations cannot be realized computational complexity of this method is:

$$KD = O(mn)$$

The articles [9]-[13] and the web resources http://www.mivar.ru/en/ show the results of the practical realization of the MIVAR nets in the UDAV software complex. Thus, these works prove that the MIVAR approach is universal for the solution of various practical problems with simultaneous logical and computational data processing. Therefore, the contradiction between logic inference and computational processing is successfully overcome by means of the MIVAR nets.

Testing the ES created by Wi!Mi was conducted for:

a) real subject domains;
b) randomly generated MIVAR networks.

The random generation of the MIVAR networks was necessary to test the computing speed for larger MIVAR networks where the quantity of rules exceeds 5000. The tests were carried out on a PC with the following specifications: ASUS Maximus V Extreme motherboard with Intel® Core ™ i7-3770K processor and 8GB of Corsair DDR3 PC12800 RAM, NVIDIA GeForce GTX 670 2048M graphics card, 120GB Intel 520 SSD and operating system Windows 7 x64. For real domain tests, the area "Geometry. Triangle" was chosen. It could be described as a 33×161 MIVAR network. The searches for inference path to solve various problems in such a

MIVAR network have taken less than 5ms. Further testing of the prototype has been carried out on a variety of subject areas similar to real life domains. The quantity of rules described in the subject area "Triangle" was approximately three times greater than the quantity of parameters. To generate a random MIVAR network, the quantity of parameters was set as <quantity of rules> / 3 (Tables VII and VIII). The memory footprint and the performance results are presented below in Table VII for regular requirements (i.e., it was required to find 0.03% of all parameters existing in the considered subject domain).Table VIII shows details of a "heavy" test (it was required to find 80% of all parameters existing in the considered subject domain). In those generated random models, the quantity of objects (parameters) is set as <quantity of rules> / 3. (As a prototype of this system, an ES in subject domain Geometry was used where the quantity of rules was approximately three times greater than the quantity of parameters).

TABLE VII
TESTS OF MIVAR NETWORK PRODUCTIVITY WHILE INCREASING THE NUMBER OF RULES. NUMBER OF RULES VARIED BETWEEN 400,000 AND 1,000,000 WITH INCREMENTS OF 100,000. IN ADDITION, THE EXAMPLE WITH 3,000,000 RULES IS SHOWN. 0.03% OF TOTAL NUMBER OF PARAMETERS WAS DEFINED AT THE END OF EACH RUN

Number of rules in the ES	Number of the parameters found	Time to build inference tree in ms	Memory footprint MB
400,000	133	1	142
500,000	166	1.45	181
600,000	200	3.02	218
700,000	233	1.71	255
800,000	266	2.6	293
900,000	300	4.6	322
1,000,000	333	3.7	386
3,000,000	1,000	14.04	1,056

TABLE VIII
TESTS OF MIVAR NETWORK PRODUCTIVITY WHILE INCREASING THE NUMBER OF RULES. NUMBER OF RULES VARIED BETWEEN 400,000 AND 1,000,000 WITH INCREMENTS OF 100,000. IN ADDITION, THE EXAMPLE WITH 3,000,000 RULES IS SHOWN. 80% OF PARAMETERS WERE DEFINED DURING EACH RUN

Number of rules in ES	Number of parameters found	Time for breadth-first search in ms	Time to build inference tree in ms	Memory footprint MB
400,000	106,666	2,762.01	934.19	199
500,000	133,334	4,112.68	1,217.19	181
600,000	160,000	5,923.57	635.51	218
700,000	186,666	7,664.87	1,768.87	255
800,000	213,334	10,038.80	1,148.22	293
900,000	239,997	12,838.90	891.93	322
1,000,000	266,665	15,547.70	1,162.70	386
3,000,000	800,000	135,571	3,608.99	1,539

The computational complexity of proposed algorithm is theoretically estimated with the help of another set of arguments. It can be assumed that at least one new parameter will be defined on each iteration. Thus, in the worst-case scenario, after n steps all parameters will be defined in the well-built MIVAR network. On the other hand, at most m rules must be scanned during each iteration. Therefore, the computational complexity of the search for an algorithm in the worst case is proportional to mn or $O(mn)$ (where m is the quantity of rules and $n$is the quantity of parameters).An estimate for breadth-first search, which is used for traversal of the MIVAR networks, gives similar results, i.e. $O(|V| + |E|)$is where $|V|$ is the quantity of vertices while $|E|$ is the quantity of edges. In the worst-case scenario, we need to visit every vertex and every edge. $|V| = m + n$ for bipartite graph and if the graph is sparse (which is normally the case) then $|E|\sim(m + n)$ and, hence, overall complexity is virtually linear with time complexity $O(m + n)$.

VII. Conclusion

In this study it is demonstrated that the Wi!Mi shell can be used for the development of expert systems belonging to various subject areas. The Wi!Mi is a C++ cross-platform software (however, it requires individual building or compilation for each platform that it supports). To build complex rules (connections between objects) in the ES a developer can use JavaScript language. The Wi!Mi system was able to handle 3,000,000 rules and construct an algorithm for solving problem is less than four seconds. This is nearly 100 times faster than was achieved with the earlier prototypes of the system [8]. This shell can be used for software development in industry, robotics, and medicine[18] or as a tutor or assistant in solving school problems [19] and in many other areas of human knowledge. The authors foresee great opportunities in developing systems where the Wi!Mi based expert system will represent an integral part of a more complex system. The Wi!Mi's computing speed makes it a natural choice for systems where hundreds of thousands of parameters and rules are present.

Acknowledgements

We would like to thank Dr. Lyudmila Artemyeva and Dr. Boris Strokopytov for their help in preparation of the manuscript. In addition, the author would like to thank E. Zhdanovich for help in preparation of robotics café model.

References

[1] F. G. Luger, *Artificial Intelligence: Structure and Strategies for Complex Problem Solving* (Boston: Pearson, 2009).

[2] S. J. Russel, and P. Norvig, *Artificial Intelligence: a Modern Approach* (3rd ed.) (Boston: Pearson, 2010).

[3] P. D. Antonov, M. O. Chibirova, E. A. Zhdanovich, G. S. Sergushin, and D. V. Eliseev, A Practical Example of Using the Mivar Approach to Create an Expert System in the Domain of "Geometry", *Radio Industry,* Vol. 3: 131-145, 2015. (in Russian).

[4] *Tool for Building Expert Systems, 2000*. Retrieved 11 16, 2017, from http://clipsrules.sourceforge.net

[5] I. Bratko, *Prolog Programming for Artificial Intelligence* (4th ed.) (Pearson Education Canada, 2012).

[6] *Analytic Technologies,* 2010. Retrieved 11 16, 2017, from http://www.fico.com/en/predictive-analytics/analytic-technologies/rules-based-systems

[7] *Expert System Development Tool,* 2011. Retrieved from http://www.exsys.com/exsyscorvid.html Accessed 17.11.16.

[8] H. E. Friedman, *Jess in Action: Java Rule-Based System* (Greenwich: Manning Publications, 2003).
[9] O. O. Varlamov, Linear Matrix Method for Determining a Route on the Adaptive Network of Rules, *Proceedings of the Universities, Electronics,* Vol. 6: 48-53, 2002a. (In Russian).
[10] O. O. Varlamov, Evolutionary Databases and Knowledge for Adaptive Synthesis of Intelligent Systems, *The Mivar Information Space* (Moscow: Radio & Communication, 2002b).
[11] M. O. Chibirova, Structural Development of the Mivar Approach: Classes and Relations, *Radio industry,* Vol.3: 44-54, 2015.
[12] O. O. Varlamov, The Role and Place of the Mivars in Computer Science, Artificial Intelligence, and Computer Science, *Radio industry,* Vol. 3: 10-27, 2015.
[13] O. O. Varlamov, M. O. Chibirova, A. M. Khadiev, P. D. Antonov, S. S. Sergushin, I. A. Shoshev, and K. V. Nazarov, *Workshop on the Creation of Mivar Expert Systems*, Textbook (Moscow, Russian Federation: "Belyi veter", M., 2016).
[14] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf & J. Wagenaar, An Analysis of Recovery Models and Algorithms for the Real-Time Railway Rescheduling, *Transportation Research Part B: Methodological,* Vol. 63: 15-37, 2014.
[15] O. O. Varlamov, *System Analysis and Synthesis of the Data Models and Methods of Information Processing in the Self-Organizing Complexes of the Operational Diagnostics* (Dissertation for the degree of Doctor of Technical Sciences. Moscow, 2003).
[16] K. Jo, J. Kim, D. Kim, C. Jang, M. Sunwoo, Development of the Autonomous Car—Part II: A Case Study on the Implementation of the Autonomous Driving System Based on Distributed Architecture, *IEEE Transoperations on Industrial Electronics,* Vol. 62(Issue 8), 5119-5132, 2018.
[17] O. O. Varlamov, R. A. Sandu, A. N. Vladimirov, A. V. Nosov, and M. L. Overchuk, The Mivar Approach to the Creation of the Multi-Subject Active Expert Systems for Izvestiya SFU, *Technical science,* Vol.11 (Issue112): 226-232, 2010.(in Russian).
[18] O. O. Varlamov, V. M. Lazarev, D. A. Chuvikov, and P. Jha, On the Prospects for the Creation of the Autonomous Intelligent Robots Based on the Mivar Technologies", *RadioIndustry,* vol. 4: 96-105, 2016. (in Russian).
[19] E. A. Zhdanovich, P. K. Chernyshev, K. A. Yufimychev, D. V. Eliseev, and D. A. Chuvikov,Calculation of the Arbitrary Algorithms for Functioning of Service Robots on the Basis of the Mivar Approach, *Radio industry,* Vol. 3: 226-242, 2015.
[20] S. S. Shadrin, A. M. Ivanov, and D. V. Nevzorov, Autonomous Wheeled Vehicle as a Part of the Intelligent Transport Systems, *Natural and technical sciences,* Vol. 6 (Issue 84): 309-311, 2015. (in Russian).
[21] S. S. Shadrin, O. O. Varlamov, and A. M. Ivanov, Experimental Autonomous Road Vehicle with Logical Artificial Intelligence, *Journal of Advanced Transportation,* Vol. 2017: 10, 2017.
[22] R. O. Duda, and E. H. Shortliffe, Expert Systems Research, *Science,* Vol. 220: 261-268, 1993.
[23] J. C. Giarratano, and G. D. Riley, *Expert Systems: Principles and Programming*. (4th ed.) (Course Technology, 2004).
[24] *Jess*, Retrieved from question, 1999: http://www.jessrules.com/jess/FAQ.shtml Accessed 17.11.17
[25] *Gensym. Real-Time Management of the Mission-Critical Systems,* 2017. Retrieved from http://www.gensym.com/
[26] R. T. Plant, and J. P. Salinas, Expert Systems Shell Benchmarks: the Missing Comparison Factor, *Expert Systems,* Vol.27: 89-101, 1994.
[27] R. Davis, H. Shrobe, and P. Solovits, What is a Knowledge Representation?, *AI Magazine,* Vol.141: 17-33, 1993.
[28] G. Nebel, and B. Lakemeyer, *Foundation of Knowledge Representation and Reasoning* (Boston: Pearson, 2009).
[29] O. O. Varlamov, *MIVAR: Transition from Productions to the Bipartite Graphs of the MIVAR Nets and Practical Realization of the Automated Constructor of Algorithms*, 2017.
[30] P. A. Jaquesa, H. Seffin, G. Rubi, F. De Morais, C. Ghilardi, I. Bittencourt, and S. Isotani, The Rule-Based Expert Systems to Support the Step-by-Step Guidance in the Algebraic Problem Solving, *Expert Systems with Applications,* Vol. 40 (Issue 14): 5456-5465, 2013.
[31] E. A. Zhdanovich, P. A. Antonov, A. M. Khadiev, G. S. Sergushin, and M. O. Chibirova, The Diagnosis of the Symptoms on the Basis of the Mivar Approach, *Radio industry,* Vol. 3: 122-130, 2015.

Authors' information

Bauman Moscow State Technical University, Moscow Automobile and Road Construction State Technical University (MADI), RI MIVAR, Moscow, Russia.

Oleg O. Varlamov is the Doctor of Technical Sciences. He is a Professor, Director of the Research Institute of MI!VAR, Information Processing and Management Systems (IU-5) Department at the Bauman Moscow State Technical University; Moscow Automobile and Road Construction State Technical University (MADI); RI MIVAR, Moscow, Russia.

E-mail: oleg.varlamov@yahoo.com