# Mivar technologies in knowledge representation and reasoning

**Varlamov O.O.** [1] **Danilkin I.A.**[2] **Shoshev I.A.**[3]
«Mivar» Ltd., 72 Oktyabrskaya street, 127521 Moscow, Russia,
1) o.varlamov@mivar.ru, www.mivar.ru, 2) i.danilkin@mivar.ru, www.mivar.ru
3) Bauman Moscow State Technical University, ul. Baumanskaya 2-ya, 5, 105005, Moscow, Russia, shoshevw@gmail.ru

***Abstract**: Today a lot of expert systems moved out into routine use. But still there are not enough flexibility and sometimes understandability. The development of tools for representing knowledge and new mechanism for logical inference may significantly change the situation. In this paper the Mivar networks as a knowledge representation mechanism is considered and the logical inference based on mivar network is examined.*

***Keywords**:* expert systems, mivar networks, logical inference.

## 1. PAPER SIZE

Expert systems are still a rapidly developing area of artificial intelligence. Accumulating knowledge of experts about subject domains, such systems emulate their actions and methodology. Modern expert systems are capable of advising when controlling complex operator consoles, when searching for faults in electronic devices, when diagnosing, they are also used in meteorology, geology, engineering, etc. [1].

Since decisions made on the basis of expert systems are extremely important, the reasoning process of such a system should be open to verification. It should be noted that the crucial difference between expert systems and other software is knowledge base, in which knowledge about a corresponding subject domain is stored. This knowledge should be represented in a form available for specialists in the corresponding subject domain and these systems should provide the possibility of modifying the knowledge base, adding new knowledge and changing already accumulated knowledge in particular. Expert systems developers often note that easy knowledge base modification is often the main criterion for success of expert systems produced.

Development and integration of the systems are especially urgent when there is a lack of specialists and when performing a small task requires involvement of many specialists.

Fig. 1 shows standard modules comprising a typical expert system.

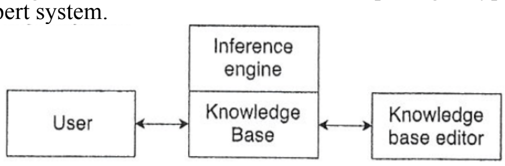


**Fig.l - Basic structure of typical expert system**

It should be noted that real expert systems can have more complex structure, however, the components described above are present in every real expert system. As it is known and also shown in the Fig.l expert systems represents knowledge-based systems and do not exist without knowledge base and inference engine. Therefore, development of knowledge representation methods and knowledge processing methods is crucial in the process of expert systems development. It should be also pointed out that although there are wide opportunities for expert systems application, the design of expert systems is still costly and labour-intensive process, and therefore, modern developments in the field of expert systems are aimed to solve tasks in highly specialized domains. Therefore, further development of theory and methods of knowledge representation and knowledge operation is a topical task for further expert systems development.

Knowledge representation method should ensure natural structure of knowledge, make this knowledge understandable for a computer and assist a developer to describe this knowledge. Moreover, such a method should allow us to solve problems arising in the subject domain [2].

Systems available today widely apply technologies based on ontologies and cognitive maps. Application of such approaches often lead to high computational complexities when searching for solution, moreover, it is often necessary to involve a large number of professional experts of high level to fill the knowledge base and solution search requires considerable expertise accumulated [3].

The paper considers the mivar knowledge representation model [4, 5], as well as logical inference mechanism. Using these tools in the process of expert system development allows us to increase the speed and improve the quality of knowledge processing, simplify knowledge acquisition and knowledge accumulation since there is often no necessity of expert involvement and changing functionality of logical inference methods.

## 2. KNOWLEDGE REPRESENTATION. MIVAR NETWORKS.

Subject domain description is a priority task in the process of developing expert systems. The concept of mivar network is used when applying mivar-based approach to data representation. It is such a network that ensures formalization and representation of human knowledge.

*Mivar network* – is a method for representing objects of the subject domain and their processing rules in the form of a bipartite directed graph $G=(V,E)$ consisting of objects *(P)* and rules *(R)*. In the aggregate these objects and rules form the model of the subject domain.

The set of the nodes of $V$ graph coincides with the set of objects of the subject domain *(P),* the set of arcs $E$ of the graph coincides with the set of rules of the subject domain *(R).*

The important characteristic of mivar network is that for each variable all the information is stored in explicit form about all the rules *R*, for which it is an input variable *(X)* or an output variable *(Y)* with the indication of that.

To build this network it is needed in the considered subject domain to pick out all necessary objects and to find connections between them.

Mivar network is constructed by binding sets of different types according to the following rules: "object-rule" and "rule-object". Interconnections such as 'object-object' and 'rule-rule' are forbidden.

Bipartite graph of mivar network can be represented in the form of two-dimensional matrix *P*R,* where *n* is the number of parameters (objects) of the subject domain, *m* is the number of rules connecting objects of the subject domain (Table 1).

**Table l. Mivar networks. Matrix representation**

| Parameters / Rules | 1 | 2 | 3 | 4 | 5 | … | n-2 | n-1 | n |
|---|---|---|---|---|---|---|---|---|---|
| 1 | X | X | X | | | | | Y | Y |
| 2 | | | X | Y | Y | | | X | X |
| … | | | | | | | | | |
| m | | | | | | | Y | | |

For each rule *R* information about all its input and output variables *P* is stored in explicit form, including the information about the number of input *(X)* and output *(Y)* variables. The storage of all the necessary information in such a network is organized on the basis of database technologies adapted for working with mivar networks.

Thus, in general form the interconnection can be represented as 'object(s)-rule-object(s)'. The element, from which the interconnection goes, is indicated first. The element, to which the interconnection moves, is indicated second. Such design determines directionality of the graph and exclude possibilities of misinterpreting or converting objects through backtracking. Designed in such a way, mivar network is scalable, as any time it is possible to add elements of sets of any types available without the necessity to change processing methods. Moreover, describing mivar network often does not require the involvement of an expert. In most cases it is sufficient to translate objectively existing objects and connections (rules) in the mivar form.

## 3. LOGICAL INFERENCE IN MIVAR NETWORKS

Representing subject domains in a way described above, it is possible to build logical inference paths in the knowledge base. Thus, being in any place of mivar network, it is always obvious from where we can move to it, and where we can move from it, which eliminates exhaustive search when searching for logical inference on the mivar network.

Such a method for building logical inference path contains the following stages:

1. Formation of the aggregate of known parameters (Z) and required parameters (W);
2. Processing is carried out taking into account each known parameter that was not processed:
   2.1. Determining fired rules, in which all input parameters are known and which were not fired before;
   2.2. Simulation of firing such rules, the results of which are used to add the list of known parameters. If all the required parameters are found, the processing described above is stopped and the sequence of fired rules is formed. The designed sequence represents the logical inference path.

Simulation of firing rule means that this rule do not launch physically but all parameters that can be found after the using of corresponding rule are marking in mivar matrix. In the presented paper this action is implemented by marking the corresponding parameter with *Z* in the additional row (see below).

Here can occur following situations. All required variables are found, the logical inference path is found. Also can occur the situation than not all required parameters are found, it means that there is not enough input data to solve the problem or that the necessary rule is missing in the considered subject domain.

Let us illustrate the scheme described above using a simple example from the subject domain "Geometry. Triangles". To do this we will give the steps of solving the following task from this subject domain:

*It is needed to find the lengths of the sides AB and AC of the triangle ABC, if the perimeter of triangle is 28 cm, the side BC is 4 cm less than AB and 9 cm less than AC.*

In the subject domain "Geometry. Triangle" variables are any sides, angles, segments of the triangle. Rules are different interconnections between these variables such as definitions, theorems, axioms, etc. The part of mivar network of the considered subject domain is represented in the form of the matrix *M* (Table2).

**Table 2. The part of mivar matrix. Geometry**

| Parameters / Rules | Side AB | Side BC | Side AC | Perimeter P | … | AB is (...) greater than BC | AB is (...) greater than BC |
|---|---|---|---|---|---|---|---|
| The perimeter of the triangle (using the lengths of three sides) | X | X | X | Y | | | |
| The side C using the perimeter and the ratios of the side | | Y | | | | X | X |
| The side AC using the ratio of the sides | | X | Y | X | | | X |
| The side AB using the ratio of the sides | X | X | | | | X | |

It should be noted that in the task described above input parameters are the perimeter of the triangle and the ratios between the sides of the triangle. It is required to find the sides of the triangle *ABC*. It can be represented in the matrix *M*. To do this let us add an additional service row and a service column to the matrix. The row is designed to track the changes in known data. The column is designed to track the rules used. The result of the first step of working with mivar matrix is represented in Table 3.

**Table 3. The mivar matrix after first step.**

| Parameters / Rules | Side AB | Side BC | Side AC | Perimeter P | … | AB is (...) greater than BC | AB is (...) greater than BC |
|---|---|---|---|---|---|---|---|
| The perimeter of the triangle (using the lengths of three sides) | X | X | X | Y | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| The side *BC* using the perimeter and the ratios of | | Y | | | | X | X |
| The side *AC* using the ratio of the sides | | X | Y | X | | | X |
| The side *AB* using the ratio of the sides | X | X | | | | X | |
| *Service row* | W | | W | Z | | Z | Z |

According to the matrix represented above, on the second step the rule "The side *BC* using the perimeter and the ratios of the sides" can be launched, which is indicated in the corresponding cell of the service column. Output of the rule after launching indicated as known variable. The results of this step are represented in Table 4.

**Table 4. The mivar matrix after first step.**

| Parameters / Rules | Side *AB* | Side *BC* | Side *AC* | Perimeter *P* | … | *AB* is (...) greater than *BC* | *AC* is (...) greater than *BC* | *Service column* |
|---|---|---|---|---|---|---|---|---|
| The perimeter of the triangle (using the lengths of three sides) | X | X | X | Y | | | | |
| The side *BC* using the perimeter and the ratios of the sides | | Y | | | | X | X | ✓ |
| … | | | | | | | | |
| The side AC using the ratio of the sides | | X | Y | X | | | X | |
| The side AB using the ratio of the sides | Y | X | | | | X | | |
| *Service row* | W | Z | W | Z | | Z | Z | |

Thus, the parameter *BC* became known and now it is possible to launch the rule "The side *AC* using the ratio of the sides". The results of this step are represented in Table 5.

**Table 5. The mivar matrix after second step.**

| Parameters / Rules | Side *AB* | Side *BC* | Side *AC* | Perimeter *P* | … | *AB* is (...) greater than *BC* | *AC* is (...) greater than *BC* | *Service column* |
|---|---|---|---|---|---|---|---|---|
| The perimeter of the triangle (using the lengths of three sides) | X | X | X | Y | | | | |
| The side *BC* using the perimeter and the ratios of the sides | | Y | | | | X | X | ✓ |
| … | | | | | | | | |
| The side AC using the ratio of the sides | | X | Y | X | | | X | ✓ |
| The side AB using the ratio of the sides | Y | X | | | | X | | |
| *Service row* | W | Z | Z | Z | | Z | Z | |

Thus, the parameter *AC* became known and now it is possible to use the rule "The side *AB* using the ratio of the sides". The results of this step are represented in Table 6.

**Table 6. The mivar matrix after third step.**

| Parameters / Rules | Side *AB* | Side *BC* | Side *AC* | Perimeter *P* | … | *AB* is (...) greater than *BC* | *AC* is (...) greater than *BC* | *Service column* |
|---|---|---|---|---|---|---|---|---|
| The perimeter of the triangle (using the lengths of three sides) | X | X | X | Y | | | | |
| The side *BC* using the perimeter and the ratios of the sides | | Y | | | | X | X | ✓ |
| … | | | | | | | | |
| The side AC using the ratio of the sides | | X | Y | X | | | X | ✓ |
| The side AB using the ratio of the sides | Y | X | | | | X | | ✓ |
| *Service row* | Z | Z | Z | Z | | Z | Z | |

Another words, by doing corresponding actions, we obtain the following algorithm to solve the set task: "The side *BC* using the perimeter and the ratios of the sides" → "The side *AC* using the ratio of the sides" → "The side *AB* using the ratio of the sides".

The algorithm complexity depends on the amount of parameters and rules in the considered domain. And in reality, for the solution of real problems the complexity is even less and is *O(k),* where *k* - is the amount of elements involved in the logical inference $(k \leq m+n)$.

## 4. CONCLUSION

Expert systems are applied to solve problems the major complexity of which is associated with using expert knowledge and where logical information processing is crucial. Modern expert systems are complex systems, in which selected knowledge representation models and knowledge processing models have key role. The paper presents a method for constructing logical inference path in the knowledge base, which contains representation of subject domain model in the form of objects and connections converted into a directed bipartite graph. Objects contain parameters and connections contain rules. Each rule has at least one input variable and one output variable; each parameter connected with the rule is its input variable or output variable. According to the method, an aggregate of known parameters is formed and one or more required parameters are set. Processing is carried out for each known parameter that was not processed before to find required parameters. Processing includes the following steps; determining fired rules, in which, known parameter is input variable, for which all other input variables are known as well as those that were not launched before; these defined rules and adding output variables of launched rules to the aggregate of known parameters. If all the required parameters are found, the processing is stopped.

Then the sequence of launched rules in the order of firing is constructed, which is logical inference path. The method may involve filtering the constructed sequence of rules such that there are only rules that were fired when searching for required parameters. The mivar-based method aims to increase the speed and improve the quality of knowledge processing, simplify knowledge base formation.

## 5. REFERENCES


[1] G.F. Lugher. *Artificial intelligence: structures and strategies for complex problem solving.* Pearson Education. Boston, 2009. p. 784.

[2] G. Lakemeyer, B. Nebel. *Foundation of knowledge representation and reasoning.* Springer-Verlag. Berlin. 1994. p. 361.

[3] M. O. Chibirova, *Analysis of existing approaches: ontology, cognitive maps and mivar nets.* Radio Industry. Vol.3, pp. 55-66. (In Russian)

[4] O.O. Varlamov, *Multidimensional informational space of data and rules representing overview,* Information Technologies. Vol.5, pp. 42-47.

[5] O.O. Varlamov. Mivar technologies of the development of intelligent systems and the criterion of the active multi-subject online MIVAR encyclopedia. *Pattern recognition and information processing. Proceedings of the 11th International Conference (PRIP'2011),* Minsk, Republic of Belarus 18-20 May 2011, pp. 326-32