Available online at: https://ijact.in

Date of Submission	21/09/2020
Date of Acceptance	22/10/2020
Date of Publication	31/10/2020
Page numbers	3892-3899 (8 Pages)

ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

REVIEW OF METHODS FOR BUILDING AGENT SYSTEMS AND DECISION SUPPORT SYSTEMS

Marina Valeryevna Cherkasskaya[1], Alexey Anatolevich Artamonov[1], Andrey Igorevich Cherkasskiy[2]

[1]Plekhanov Russian University of Economics, Stremyanny Lane, 36, Moscow, 117997, Russia

[2]National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe Highway, 31, Moscow, 115409, Russia

Abstract: This article presents data on the properties and features of agents' behavior, ways of their communication and multi-agent systems based on them. Criteria for building decision support systems using modern tools, basic stages of designing and maintaining information repositories, technologies of operational and mining data analysis, as well as genetic algorithms and knowledge models in expert systems are considered. A set of tools is described that provide adequate reality forecasts that help to obtain the necessary information for making decisions in the conditions of market competition.

Keywords: multi-agent systems; decision support systems; distributed artificial intelligence; Data Mining; decision trees; neural network; genetic algorithm; expert systems.

I. INTRODUCTION

Under the conditions of constantly increasing volumes of stored and processed information and the need to transfer certain processing functions of this information to intelligent systems, agent technologies have become in demand in many subject areas. Logistic operations, concepts of intelligent energy systems [1], risk assessment and improving the efficiency of enterprises, issuing visas, population behavior in emergency situations, modeling the development of international transport networks [2], modeling socio-economic and political relations are just an incomplete list of areas in which agent technologies for data collection and processing have been introduced and are actively used. The basis for the development of such complex software and information systems is the definition and formulation of the requirements and tasks of users and developers, as well as the identification of the functionality that the system will have as a result to meet these needs.

II. THE CONCEPT OF AGENTS AND MULTI-AGENT SYSTEMS

According to S. Russell and P. Norvig [3], an agent is understood to mean any entity located in a certain environment and perceiving it using sensors, receiving data reflecting events occurring in the medium, interpreting these data and acting on the medium through effectors. Thus, in order to form an agent, it is necessary to distinguish the following factors forming criteria - environment, perception, interpretation and action. The agent has certain properties. It is autonomous, capable of setting goals for itself and performing actions for their implementation; adaptive, that is, self-learning; an agent can communicate with other agents - it is communicative, and can also interact with them in different ways [4].

To achieve any goals, agents can interact with each other and with the passive environment they form, it forming multiagent systems (MAS). Each agent of such a system has its own ideas about the outside world, tasks and logic that determine its behavior. In the process of operation, agents communicate with each other. Sensors agents are responsible for the collection and processing of information, and agents' effectors are affecting the medium. Agents can act independently of each other, or conflict for resources and communicating to resolve disputes [5].

Cognitive and reactive agents are distinguished depending on the development of the concept of the external world. The first are intellectual, have a highly developed idea of the outside world and act on the basis of its analysis. Systems made up more of intelligent agents form distributed artificial intelligence (DAI) [6]. Such MAS generally consist of a small (not more than tens) amount of highly intelligent agents. The intelligence of the whole system is based on the ability to solve complex tasks of agents individually. Cognitive agents have a developed idea of the external environment, have the ability to reason, can remember and analyze various situations, anticipate reactions to their actions, draw conclusions from it, that necessary for further actions and plan their behavior [7]. MAS of cognitive agents usually have a clearly defined goal that they need to achieve. It is the intellectual abilities that allow such agents to build virtual worlds, working in which they form action plans [8]. They are used for distributed solution of complex tasks, for example, for simulation of actions of participants of military operations, for implementation of remote training systems or in logistics planning at various stages of process forecasting.

Reactive agents act on the basis of the rules laid down in them and they form a system called artificial life (AL). These systems are built using tens to thousands of agents that have a simple internal structure and without complex behavior. As a result of the joint actions of a large number of agents and the probabilistic nature of their behavior, such MASs exhibit "swarm intelligence”. Their application has a great practical application in the management of socio-economic systems, where it is necessary to take into account the behavior of a large number of participants whose actions outline a general picture of the process, for example, purchasing activity in a particular market segment. The ultimate goal of an agent in such systems is search for approximation to the best solution to a particular problem.

A. *Agent communication in multi-agent systems*

When building MAS, it is very important to establish bilateral dynamic relationships between agents. Communication between agents depends on a protocol consisting of many rules [9]. Communication mechanisms are divided into direct, associated with the exchange of information through the transmission of messages, and indirect, implemented, for example, using the architecture of the "bulletin board". In the latter case, the agents try to fill out an intermediate module for the implementation of various interactions between the agents - the bulletin board, noting all known information and indicating the tasks to be solved. The bulletin board should have a message passing area for exchanging requests for various knowledge modules, information about the problem-solving process and the state of agents. In the simplest case, the bulletin board has a static structure containing slots describing various hypotheses and communication options. When working, knowledge modules generate, update, and evaluate these hypotheses related to a problem area. Various agents define how to formulate and prove the hypothesis and then verify it [7].

Exchange of information between agents allows them to coordinate their actions. One of the important criteria in the field of agent communication is the construction of protocols for negotiation. When agents are selfish, agreements between them are always mutually beneficial. If an agreement maximizes the utility functions of agents on the set of valid agreements, it is beneficial to all agents. More flexible altruistic agents must be included in the MAS to achieve MAS autonomy.

The construction of a negotiation protocol includes three stages: determining the space of possible agreements, introducing interaction rules and setting optimal strategies for agents. By sharing their action plans and communicating their goals, methods, and timelines, agents define common priorities, balance the workload, and ensure system-wide efficiency [10].

Consider the most common models of agent communication in distributed artificial intelligence. R. Smith developed a model of contract network [11], which is still current. It is based on the idea of market trading with two types of agents. The agent-manager sends out an announcement about the task, puts out the initial price, and agent-performers offer services with their price variants, participate in the competition to determine the best offers on the initial task. After that, the agent-manager selects the most profitable offers for himself and enters into an agreement with the selected agents by the performers who change to the status of agent-contractors. The contract network model is simple and easy to implement. At the same time, there is no well-thought-out contractor selection mechanism, and communication channels experience high load. Disadvantages also include the possible presence of speculator agents capable of buying and reselling the same task several times.

Among the protocols of negotiation in the DAI, a protocol of monotonous minimum concessions is widely distributed [12], which regulates the negotiation of two agents. Agents make offers in turn, starting with the most profitable for themselves. During the negotiation process, agents retreat from their original demands and come to the kind of agreement that maximizes the product of their utility functions. One of the greatest strengths of MASs is their ability to meet the most difficult task of developing complex programs - the interaction of many components of the program. This is achieved by integrating the most advanced achievements in the field of information technology, artificial intelligence, distributed information systems, computer networks and computer technology [13]. At the same time, further theoretical study of MAS remains relevant, creation of new approaches, methods of construction, as well as development of means of their description, research and implementation both of existing systems, and search for ways of their improvement and creation of new systems with certain properties.

III. Decision Support Systems

In order to support multi-criteria solutions in a complex information environment, decision support systems (DSS) are currently being actively developed, solving the problems of choosing the best solution from a variety of possible solutions and ranking the resulting decisions according to preference [14]. These systems are becoming the strategic tools necessary for the sustainable development of firms, organizations and enterprises. A decision support system is a computer system that, by collecting and analyzing a large amount of information, can influence the decision-making process of an organizational plan in business and entrepreneurship [15]. DSS allows, for example, to evaluate all information assets, values of sales volumes, to predict the organization's income with the possible introduction of new technologies, to see all possible alternative solutions. Assistance is provided to the decision maker (DM) in assessing the current situation and constraints of the external environment, identifying and ranking priorities, generating possible solutions and evaluating alternatives, analysing the consequences of decisions taken and choosing the best solution to the problems.

The preference of a particular outcome of the operation is evaluated by the value of a special numerical function - a criterion. It is significant for the decision maker, well interpreted by him to assess the possible outcome of the operation. The best option is one that provides the best value of the criterion or a compromise combination of the meaning of all the criteria, if there are several.

In practical situations, it is rarely possible to assess situations by only one criterion. For example, considering the activities of a trading company, you need to consider sales volumes, storage costs of goods, profit, and turnover of funds. These values allow you to build criteria. For example, profits need to be increased while minimizing storage costs. Obviously, these performance criteria are contradictory, resulting in that it is impossible to find the best solution for all criteria. The presence of uncertain factors in combination with multicriteria complicates decision making. All risks and losses in case of adverse circumstances must be taken into account.

The basic position of decision-making theory is the principle that there is no absolutely better solution. The optimal solution will be only for a certain decision-maker, for his goals, only in this place at a given time. The DSS does not replace the person in the decision-making process, it is aimed at helping to understand the difficult situation. Two areas of such support are identified. The first is to facilitate the interaction between data, data analysis and processing procedures and decision models, on the one hand, and the decision-maker as user of these systems, on the other. The second comes down to providing supporting information for solving poorly structured and unstructured problems [16].

Such systems become indispensable for decision makers and provide the specific needs of the decision-making process, can be adapted for group and individual use, for decision makers at various levels, support both interdependent and sequential decisions, implement different styles and methods of decision, flexibly adapt to changes in the organization and its surroundings should be easy to use and modify and present the results in a form that would contribute to a deeper understanding of the results [15].

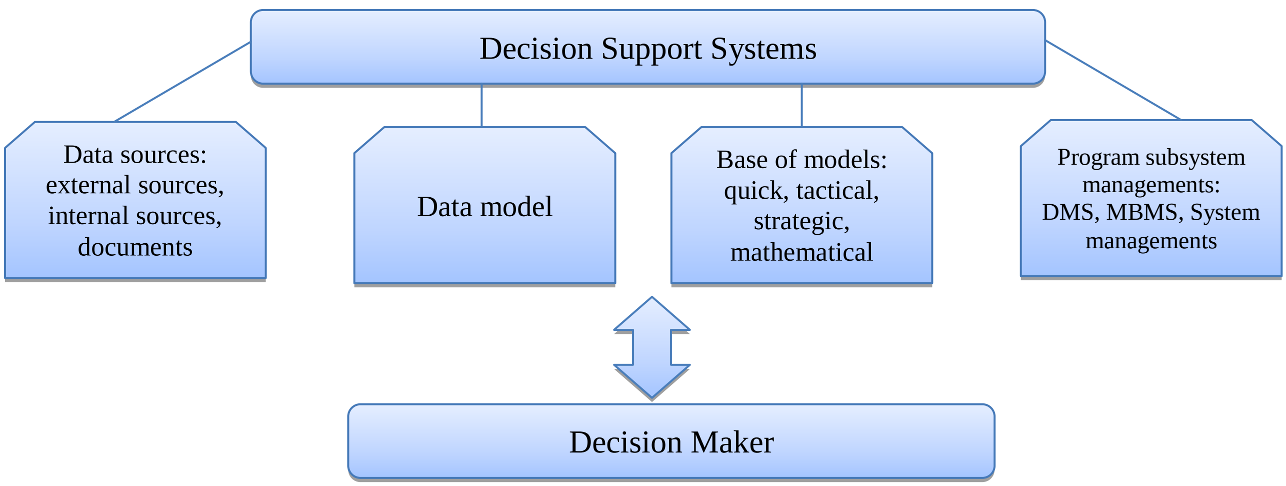

Figure 1: Structure of the decision support system

A. Composition of decision support systems

DSS includes the following components: data sources, data model, model database and software subsystem, which consists of a database management system, a model database management system and an interface management system between the user and the computer (Figure 1). In DSS, two types of models are used: models of the studied objects and decision-making models related to the organization of their search. The model of the second type is implemented according to a two-level scheme. The first level is built directly on the basis of the object model, these are optimizing strategies for experimenting with the object and calculations associated with the execution of processes such as "target-search" or "what-if". The second level is related with implementation of the scheme: statement of a problem → analysis of a problem → choice.

Object models are divided into formal, expressed by mathematical means, which describe the structural and dynamic properties of objects, and models based on processes or rules. The latter are based on the concepts of "object", "attribute" and "script" [16].

B. Approaches to building decision support systems

There are several approaches to the construction of the DSS. The basis of the logical approach is Boolean algebra and fuzzy logic. A system based on this principle is a theorem proving machine. The source data are stored in the database in the form of axioms, the rules of logical inference are stored as relations between them, there is a block for generating the target, the output system tries to prove this goal as a theorem. In the case of its proof, we obtain a sequence of actions necessary for the implementation of the task. When building a system with fuzzy logic, the veracity of the statement, in addition to the positive and negative version, has intermediate values. This approach is reminiscent of human thinking. Most logical methods are very time-consuming, as it may take a complete run of options to find evidence. Therefore, this approach implies efficient implementation of the computing process with a small database size.

Using a structural approach, DSSs are built by modeling the structure of the human brain. These systems are intelligent, select and make decisions based on previous experience and rational analysis of external factors.

The evolutionary approach of creating an initial model and rules by which it can evolve is widely used in the construction of the DSS. A model can be composed using a set of genetic algorithms, or a set of logical rules, or it can be an artificial neural network [15].

The simulation approach is another method of building DSS. The object whose behavior is simulated is a "black box", in which the known input and output data are significant. The main disadvantage of this approach is the low information ability of most models obtained with its help.

Strategic DSSs, based on the analysis of a large amount of information from different sources, providing information on problem-solving experiences, should be developed in depth and transformed for ease of use in the decision-making process. For DSS of this level, an obligatory component is a data warehouse focused on supporting the data analysis process, ensuring data integrity, consistency and chronology and a high speed of execution of analytical queries [15].

C. Intelligent Information Systems

Intelligent Information System (IIS) is an information system that is based on the concept of using a knowledge base and Data Mining models to generate algorithms for solving economic problems of various classes depending on the specific information needs of users.

Data Mining (DM) is the discovery of previously unknown, non-trivial, practically useful and accessible interpretations of the knowledge needed to make decisions in various spheres of human activity [17]. The basic methods of data mining include neural networks, decision trees, logistic regression, associative rules.

Intelligent information systems have developed communication skills in terms of end-user interaction with the system, they are self-trained, they can solve complex poorly formalized tasks using an original solution algorithm, depending on the specific situation [18].

There are several classes of IIS, consider each in more detail.

Intelligent interface system is IIS for searching for implicit information in a database or text for arbitrary queries compiled in a limited natural language. In the request, it is necessary to search by the condition formulated in the dialogue with the user.

Hypertext systems search for keywords in text information bases. The search engine, first of all, works with the knowledge base of keywords, then directly with the text.

In context assistance systems, the user can describe his problem, after which the system, in the course of additional dialogue, concretizes it and will search for recommendations itself. These systems belong to the class of dissemination of knowledge and are developed as an application to documentation systems.

Cognitive graphics systems implement the IIS interface using graphic images that are formed as a result of ongoing events.

A self-learning system is an IIS that forms units of knowledge based on real-world practices. Such systems are based on methods of automatically classifying examples of real-world practices over a period of time that constitute a learning sample. The latter can be both "with teacher," when for each example the value of the attribute of its belonging to some class of events is explicitly specified, and "without teacher," when by the degree of proximity of values, the system itself recognizes classes of events.

As a result of the system training, generalized rules and functions are automatically built, which the system itself uses when interpreting new emerging events, that is, the knowledge base for solving forecasting problems is automatically formed.

D. Decision trees

Decision trees are among the most powerful and widely used Data Mining tool to solve problems of classification, regression and forecasting of basic economic, social, environmental indicators. Decision trees are the most visual and easily interpreted models, as they use hierarchically solving "if-to" rules. Each tree node has the following parameters: the number of examples that have entered the node; percentage of examples in each class; the number of classified examples and the percentage of records correctly classified by this node. The work of trees is based on the process of recursive division of the original set of observations or objects into subsets that are associated with classes (Figure 2). The process begins by defining an attribute to be checked at the root of the tree. Each attribute is evaluated by how well it alone classifies the dataset.

When an attribute is defined, a tree branch is generated for each of its values, the dataset is split according to the value in each branch. This process is repeated recursively for each branch [15].

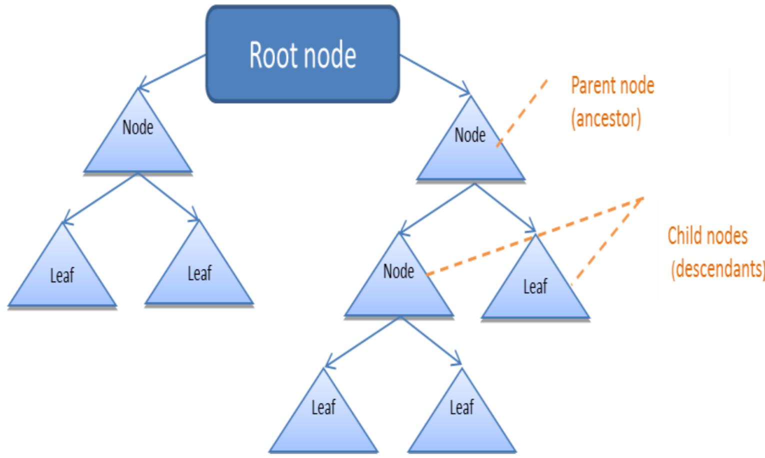

Figure 2: Decision Tree Structure

A number of conditions must be met for the solution tree to be built effectively. The data to be analyzed must be presented as a structured set containing all information about the object or observation and expressed by a set of characteristics describing the objects to be classified. The categories to which observations relate must be set in advance. It should be possible to establish the fact of belonging to a certain class, and the training set should contain a large number of different examples.

E. Neural networks

A neural network is a self-learning system that can analyze newly received information, find patterns in it, and perform forecasting. Computing structures are meant that simulate simple biological processes that resemble the processes of the human brain. The work of a neural network consists of converting an input vector into an output vector, this transformation is given by the weights of the neural network. The learning process boils down to adjusting the weights of neurons to find the state of weights that minimize the output network error in the training and test sets. The neuron model calculates the weighted sum of its inputs from other neurons, and if it is exceeded, an output signal is generated.

There are three types of networks. In fully connected networks, each neuron transmits its input signal to the rest of the neurons, including itself. In multilayer networks, neurons are combined into layers. In such networks without feedback, the neuron of the input layer receives the input signal, converts it, and transfers to the neuron one hidden layer, up to the output. In multi-layer feedback networks, information from subsequent layers is transmitted to the previous layer. In weakly connected networks, neurons are located in the nodes of a rectangular lattice [19].

Neural networks solve problems of classification, clustering, approximation of functions, prediction, optimization in tasks with large search space.

Neural network analysis begins with preparation of initial data and formation of training sample. The type of neural network architecture is then selected: the type of neuron with its activation function, the number of inputs and outputs, the number of layers and neurons in each layer. At the stage of data preparation, input/output coding takes place, if necessary, data normalization and ensuring independence between inputs of the neural network. Before training the network, its initial state is set. Then the model is checked for adequacy with real data. The operability check is carried out by using statistical consent criteria (in case of unsatisfactory result the network architecture type is revised). After that, a neural network is selected, which according to the results of training is most suitable for solving the problem.

Consider neural network learning algorithms. By training is meant a purposeful change in the weight coefficients of the synoptic connections of the network neurons based on the conditions of achieving the required network characteristics. In the course of training, the neural network detects complex dependencies between input and output data, makes a generalization. If the training is successful, the network can return the correct result based on data that was not in the training sample.

The error back propagation algorithm is a gradient learning algorithm for a multilayer perceptron, based on minimizing the mean square error of the network outputs. Training of artificial neural network takes place sequentially: direct propagation of input training image, calculation of its reverse propagation error and adjustment of weights. The error back propagation algorithm is a learning algorithm with a teacher. Based on the difference between the desired and target network outputs, the network output error is calculated.

F. Genetic algorithms

A genetic algorithm is a heuristic search algorithm for solving optimization and modeling problems by randomly selecting, combining, and varying the parameters sought using mechanisms similar to natural selection in nature. The method is based on computer simulation of biological evolution according to Darwin's theory. The main focus is on the use of the “crossover” operator, which performs the operation of recombining candidate solutions. The genetic algorithm is an iterative process that continues until the generations cease to differ from each other, or a predetermined number of generations or a predetermined time passes. The simplest algorithm involves the operation of randomly generating an initial population of chromosomes and a number of operators providing generation of new populations based on the initial. Such operators are reproduction, cross-linking and mutation [15]. During reproduction, chromosomes are copied taking into account the values of the objective function. Chromosomes with better values have a greater chance of falling into the next population. Crossover and mutation operators change the structure of chromosomes, up to the destruction of successful fragments of the solutions found. This reduces the likelihood of finding a global optimum. To resolve this problem, templates with fragments of decisions or chromosomes are used, which must be preserved during evolution.

Genetic algorithms have some differences from other optimization algorithms. They use not parameters, but encoded sets of parameters, the search is carried out not from a single point, but from a population of points, and the search process uses the value of the target function, not its increment. Probable rather than deterministic rules for finding and generating solutions also apply. A simultaneous analysis of various areas of space and solutions is performed, which makes it possible to find new areas with the best values of the objective function when combining optimal solutions from different populations [20].

Genetic algorithms have no significant mathematical requirements for types of target functions and constraints, linear and nonlinear constraints defined on discrete, continuous, and mixed universal sets can be used. Such algorithms work well with minimal information about the environment, can be combined with other artificial intelligence methods, increasing their efficiency. Most often, the genetic algorithm is used to solve optimization problems.

G. Expert systems

Expert systems are information systems that include the knowledge of specialists about a certain problem area and that are capable of making expert decisions within that area. Knowledge engineers (cognitologists), experts in the necessary subject area and end users are involved in their development. The main task of cognitive scientists is the selection of software and hardware tools for the project, assistance to the expert in the formulation of the necessary information and its implementation in an effective knowledge base. The expert of the subject area is the person who works in the sphere of the project, understands the principles of solving its tasks, knows the techniques of the solution, can provide management of inaccurate data, evaluation of partial solutions. The expert transfers these skills to a knowledge engineer who may not be familiar with the subject area of the project. The end user sets the basic design constraints. Development will continue until it is satisfied.

The purpose of research of expert systems is to develop such programs, which when solving difficult for expert-person, are able to obtain results, which are not inferior in quality and efficiency to decisions received by the expert. Expert systems are necessary for solving problems without an algorithm solution or difficult to formalize tasks. Such tasks may include: interpretation of data; diagnosis and troubleshooting of living and non-living systems; monitoring; designing system components that meet target conditions and multiple design constraints prediction of possible consequences of the given situation; planning to achieve certain objectives under specified conditions and time constraints assistance in the technical field education process; control the behavior of a complex environment or system testing of quality of work; execution of plans for organization of correction of detected defects [15].

Depending on the connection with real time, there are quasidynamic expert systems that interpret a situation that changes with a certain fixed time interval and static in which the knowledge base and interpreted data do not change in time are divided.

In terms of integration with other programs, there are autonomous expert systems operating directly in user consultation mode for specific expert tasks without involving traditional data processing methods, and hybrid expert systems aggregating standard application packages and knowledge manipulation tools.

The technology of processing expert systems consists of six stages (Figure 3). The identification phase identifies the tasks to be solved and the development objectives, as well as the experts and the type of users. At the stage of conceptualization, a meaningful analysis of the subject area is carried out, basic concepts and relationships are identified, methods of solving problems are determined. At the formalization stage, software tools for system development are selected, ways of representing all types of knowledge are determined, and basic concepts are formalized. At the stage of implementation, experts of knowledge bases are filled, in which knowledge is "transferred" by experts, knowledge is organized for effective operation of the expert system. During the testing phase, the expert and the knowledge engineer verify the competence of the system until the expert decides that the system has reached the required level. At the trial operation stage, the suitability of the system for end users is checked.

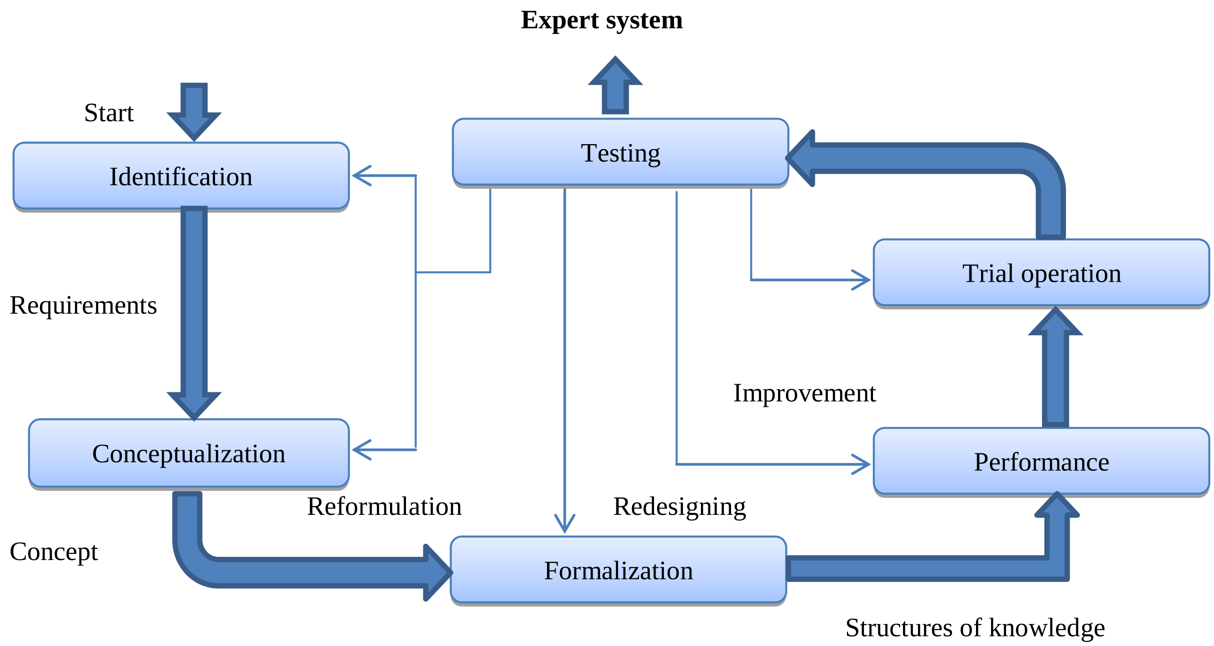

Figure 3: Expert System Development Technology

The scheme of a generalized expert system is as follows. Linguistic processor interacts with user in natural language for him. The knowledge base provides storage of knowledge presented using one of the models - logical, production, frame, network. The working memory stores data related to the task analyzed by the system. The output machine on the basis of input data, product rules and general facts about the problem area forms the problem solution. The knowledge acquisition component is used to automate the process of filling the expert system with knowledge, when adjusting the knowledge base, updating, adding to or excluding knowledge elements. There is also an explanatory component to justify the actions of the system. The expert system operates in knowledge acquisition mode and problem-solving mode.

One of the main issues of building knowledge-based systems is the choice of the form of their presentation in a computer-interpreted form. Next, let 's take a look at today 's data presentation models.

In the production model, knowledge is represented using facts and rules using the "if" (condition) - "then" (action) construct. If the facts in the task satisfy the criteria of the «if» rule, then the action defined by the «then» part occurs. This can be an impact on the outside world, or an impact on the management of the program, or indicates to the system to add new facts or hypotheses to the database. The model works with facts and rules, is aimed at solving simple and homogeneous tasks. Disadvantages include the inability to effectively describe a rule with exceptions.

The model of representing knowledge using predicate logic uses the mathematical apparatus of symbolic logic. The basic formalism of the representation of predicates is "term," which establishes the correspondence of sign characters to the object described, and predicate to describe the relationship of entities in the form of a relational formula containing terms. The model is universal, works with facts and rules, but cannot be used to create intelligent systems that must simultaneously manipulate special knowledge from different subject areas.

A knowledge representation model using semantic networks consists of vertices called nodes, corresponding to objects, concepts, or events, and the arcs linking them, which describe the relationships between the objects in question. The IS-A (the "is" relation) and the HAS-PART (the "has part" relation) arcs are used to represent the hierarchy. They also establish the inheritance hierarchy in the network. This is to save memory by eliminating the need to repeat inherited property information on each node of the network. The model is universal, easy to adjust.

The frame model is similar in its organization to the semantic network. It is also a network of nodes and relationships organized hierarchically. Upper nodes are general concepts, and their subordinate nodes are private cases of these concepts. In frame-based systems, the concept in each node is defined by a set of slot attributes and the values of those attributes. Each slot can be associated with special procedures that are executed when attribute values change. The model is universal, but training of frame systems is difficult. Acquisition of new knowledge is possible only in systems with complex frame structure, their creation requires serious time and money [15].

IV. Conclusion

Each of the models described has disadvantages such as insufficient universalism, difficulty in obtaining new knowledge, the ability to obtain conflicting knowledge, and the difficulty of building up the model. Therefore, recently much attention has been paid to the combination of different models and the creation of hybrid systems of solution support, combining methodologies of expert systems, theory of fuzzy sets, artificial neural networks, genetic algorithms. An example of hybrid systems is the training of artificial neural networks using genetic algorithms. This provides a high-speed adjustment of network weights and small learning errors. Thus, in support of managerial decision-making, a set of means can be used to provide adequate reality forecasts that help to obtain the necessary information for decision-making in a market competitive environment. Therefore, the development of such projects is a relevant and promising direction.

V. Acknowledgements

The study was carried out at the expense of the Russian Science Foundation grant (project # 19-71-30008).

VI. References

[1] Galperov, V. I. 2007. Methods, models and algorithms for constructing multi-agent systems in the energy sector: on the example of the task of assessing the state of electric power systems (PhD thesis). Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia.

[2] Yesikova, T. N. and Zaytsev, I. D. 2010. Development of the agent model “Evaluation of the strategic directions of the backbone transport network of Russia with different geoeconomic architectonics of the WORLD Economic System”. Large-scale Systems Management (MLSD'2010): Proceedings of the Fourth International Conference. Moscow, Russia, pp. 107-114.

[3] Russell, S. and Norvig, P. (2009). Artificial intelligence: a modern approach (3-d edition). Pearson Education Limited.

[4] Garro, A., Mühlhäuser, M., Baldoni, M., Bergenti, F., and Torroni P. 2018. Intelligent Agents: Multi-Agent Systems. Encyclopedia of Bioinformatics and Computational Biology, 1: 315-320.

[5] Zaytsev, I. D. 2014. Multi-agent systems in the modeling of socio-economic relations: the study of behavior and verification of properties using Markov chains (PhD thesis). A.P. Ershov Institute of Informatics Systems (IIS), Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia.

[6] Chaib-Draa, B., Moulin, M., Mandiau, R., and Millot, P. 1992. Trends in Distributed Artificial Intelligence. Artificial Intelligence Review, 6: 35-66.

[7] Tarasov, V. B. 1998. Agents, multi-agent systems, virtual communities: a strategic direction in computer science and artificial intelligence. Artificial Intelligence News, 2: 5-63.

[8] Bolotova, L. S. 2012. Systems and methods of artificial intelligence: knowledge-based models and technologies. Moscow: Finance and Statistics,.

[9] Cohen, P. R., and Levesque, H. J. 1990. Intention is Choice with Commitment. Artificial Intelligence, 42: 213-262.

[10] Mitrakov, A. A. 2013. Approaches to agent-based systems designing. Materials of the V International Student Scientific Conference "Student Scientific Forum". Moscow, Russia, pp. 1-15.

[11] Smith, R. G. (1980). The Contract Net Protocol: High Level Communication and Control in a Distrubuted Problem Solver. IEEE Transactions on Computers, 29(12): 1104-1113.

[12] Rosenshein, J. and Zlotkin, G. 1994. Rules of Encounter: Designing Conventions for Automated Negociation Among Computers. Cambridge MA: MIT Press.

[13] Gorodetsky, V. I., Grushinsky M.S., and Khabalov A.V. 2008. Multi-Agent Systems (Overview). Artificial Intelligence News, 2: 64-105.

[14] Paklin, N. B. and Oreshkov, V.I. 2010. Business Intelligence: From Data to Knowledge. Saint Petersburg: Piter.

[15] Prokopenko, N. U. 2017. Decision Support Systems Based on Deductor Studio Academic 5.3. NNSAGU, Nizhny Novgorod.

[16] Zaskanov, V. G., Ivanov, D.U., and Grishanov, G. M. 2013. Decision Support Systems. Samara National Research University, Samara.

[17] Barsegyan, A. A. 2007. Data Analysis Technologies: Data Mining, Visual Mining, Text Mining, OLAP. Saint Petersburg: BHV-Petersburg.

[18] Cao, L., Gorodetsky, V. I., and Mitkas, P. A. (2009). Agents and Data Mining. IEEE Intelligent Systems, 24(3): 64-72.

[19] Kulik, S. 2015. Neural network model of artificial intelligence for handwriting recognition. Journal of Theoretical and Applied Information Technology, 73(2): 202-211.

[20] Kulik, S., and Protopopova, U. 2020. Educational Intelligent System Using Genetic Algorithm. Procedia Computer Science, 169: 168-172.