Frame-based Expert System Implementation for Resource Conversion Processes Analysis

Anna A. Antonova and Konstantin A. Aksyonov

Ural Federal University, Yekaterinburg, Russia,
antonovaannas@gmail.com

Abstract. The paper considers the resource conversion processes, which include production, logistic, and business ones. An approach to development of an expert system based on application of the frames and production model for the knowledge representation is proposed. A frame expert system has been implemented basing on the proposed approach in the Bpsim.MSN decision support system. The inference machine of the expert system is implemented using the object-oriented approach and advanced sequence diagrams of the UML language. The developed frame-based expert system has been applied to solve the problem of alternative generation and evaluation in analyzing the processes of network communication equipment replacement.

Keywords: expert system, frame, resource conversion process, dynamic multiagent modeling, alternatives evaluation

1 Introduction

Resource conversion processes include such classes of their types: production, business, and logistic ones. The resource conversion process is a convertation of input resources into output ones subjected to availability of the free execution mechanisms and triggering the process start conditions. The purpose of the study is to develop and test a frame-based expert system for analysis of the resource conversion processes. This problem is related to the development of intelligent systems by the multiagent dynamic modeling. The apparatus of multiagent systems is used in analysis of the resource conversion processes for accounting the human factor and decision-making scenarios based on the corresponding knowledge. The multiagent systems consist of interacting intelligent agents, which are identified by the decision makers. Multiagent modeling for analysis and optimization of the resource conversion processes is already used in practice [1, 5–7, 12]. The developed frame-based expert system for analysis of the resource conversion processes will be in demand when implementing the intelligent agents with knowledge in solving the practical problems of the multistage decision-making.

2 Analysis of knowledge representation models for resource conversion processes

The main objects of the multiagent resource conversion process model are the following ones [2, 3]: operations, resources (RES), mechanisms ($MECH$), control commands (U), processes, resources senders and receivers, junctions, parameters, goals (G), messages, and agents. The description of the cause-effect relationships between the model elements is given by the Relation object. The apparatus of production systems has been used for the construction of the dynamic modeling system core. The structure of the production system (PS) of the multiagent resource conversion process is determined as follow:

$$PS = \{R_{PS} \cup B_{PS} \cup I_{PS}\}, \tag{1}$$

Here, B_{PS} is the set of rules for the resources and agent actions converting (knowledge base); I_{PS} is the inference machine consisting of a logical inference engine based on the agents knowledge base; R_{PS} is the current state of the resources, mechanisms, control commands, and goals (working memory).

$$R_{PS} = \{RES(t) \cup MECH(t) \cup U(t) \cup G(t)\}, \tag{2}$$

The simulating algorithm consists of the following basic steps: 1) determination of the current time $SysTime = \min(T_j), j \in RULE$ (where T_j is the activation time of the j–th transformation rule; $RULE$ is the set of resource conversion rules); 2) agent actions processing; 3) forming the conversion rule queue; 4) carrying out the transformation rules; 5) changing the state of the working memory. The simulator accesses the expert system module in order to diagnose the situations and generate control commands [1].

The basic requirements for the expert system of the multiagent resource conversion processes are the following:

1. orientation to the hierarchical processes of the resource conversion;
2. solving the problem of technical and economic design of the organizational and technical systems;
3. presence of communities of the intelligent agents that control the process;
4. accounting the decision-making scenarios based on knowledge;
5. support for the object-oriented approach;
6. visual means of working with the knowledge base and visual designer of the inference engine (UML language can be used as the basis of the visual language).

As the knowledge representation models for resource conversion processes, an integrated approach based on the frames and production model has been chosen.

The frame approach of A.N. Shvetsov [7] based on combining the frame-like structures with constructions of the conceptual graphs J.F. Sowa [8–10] has been used to construct a conceptual domain model and solving the problem of reducing the software development cost. Advantages of this approach are the frames division into active and passive frames and accounting for the object behavior.

3 Analysis of applicability of conteporary industrial relational databases for creation of frame-based expert system

The founder of the frame systems is M. Minsky. The frame-based model of the knowledge representation is well described in the work of H. Ueno and M. Ishizuka [11]. When implementing a frame-based expert system, the problem is to convert the frame structures to the tabular form. The methods of application of the relational databases to the frames implementation are not described in the literature.

In a simplified version, the frame model is a classic tree; thus, the problem of the frames implementation is reduced to the choice of the method for storing a tree-like structure. The simplest model is a tree with links of the "abstract-specific" type. There are several object instances in each node of the simplest tree. Such a structure has specific functions, it is rare but it is quite simple to implement. With a static frame structure, this structure is easily organized: a separate table for the data storing is formed for each template (each table line corresponds to an object instance). In this case, there is no need to create additional tables, monitor the uniqueness of table names, and there are no errors related with the integrity of the tree structure. The system constraints application allows ensuring the uniqueness of all records. The stored procedures allow the data encapsulating. However, the set of functions of such tree will be very limited.

Often, there is a need for more complex tree models. If it is necessary to dynamically change the tree model structure, there is need to enter the frame description table. For this purpose, a slot description table is proposed in [11] with the following columns: Slot name, Inheritance pointer, Attribute pointer, Slot value, and Daemon.

H. Ueno and M. Ishizuka in [11] suggest entering a similar table for each model object. This is justified if the frame system is created on a device that is oriented to the similar tasks. When implemented the frames in the MS SQL Server, we are faced with the following factors: 1) the slots values need to be predefined and monotonous, and need not to correspond to the attribute pointer; 2) there is no need for an attribute pointer, since information about it can be obtained from the system tables; 3) storage of the daemon in the table is impossible; 4) creation for each object a new table is irrational; 5) a lot of data in the tables is repeated.

For implementation of the frame-based expert system on the MS SQL Server, there is need to convert the slot description table structure as follows: create a separate table to describe the entire tree's slots and identify the slot by some name and object name. Since the frame structure assumes uniformity of the properties of all objects belonging to the same frame, it is possible to use one group of records for all similar objects. In this case, there is need to exclude the field "Slot value" from the slot description table and create a separate table for the slot values. The "Daemon" field in the slot description table can be omitted by creating the triggers for the table of frame instances, in the field "Stored procedure", the name of the stored procedure, and information about its parameters can be entered.

We obtain the following tables (Tables 1 and 2). In Table 1, the following optional fields are omitted: Tree name, Stored procedure, and Parameters.

Table 1. Slot description

Slot name	Frame name	Data type	Inheritance pointer	Attribute pointer
Slot 1	Frame 1	...	...	...
Slot 2	Frame 1	...	...	...
...	...	...	...	...
Slot M	Frame 1	...	...	...
Slot 1	Frame 2	...	...	...
Slot 2	Frame 2	...	...	...
...	...	...	...	...
Slot N	Frame 2	...	...	...

Table 2. Frame description

Object name	Slot 1	Slot 2	...	Slot M
Object 1	...	...	...	...
Object 2	...	...	...	...

To implement references and formulas, the following mechanism is used: 1) for each slot that contains a link or formula, the two fields are created in the table, *i.e.*, the field with the current value (this field has the data format of the slot) and the field containing the link or formula; 2) in the frame description table, the value that indicates whether the slot contains link or formula is set to the template attributes field.

The storage of inheritance information in the tree is implemented in Table 3.

Table 3. Tree lists

Tree	Ancestor	Descendant
Tree H	Frame 1	Frame 2
Tree H	Frame 2	Frame 3
Tree H	Frame 1	Frame 5

Using the similar structure to organize the frame leads to the problem of the data encapsulation. There are stored procedures for the data processing in tables when working with a static frame-model. But if there is a need to make changes to the model structure, this will lead to errors (the model damage). In this case, the following two approaches exist.

The first approach is that the data are entered into the slot description table initially. This operation is quite simple and for its implementation there is no a need in the additional built-in procedure. Then the procedure is started to generate a new table, register it, and change the data in the tables that describe the tree structure. The input data for this procedure will be the names of the new frame and its ancestor.

The second approach is universal; it allows simplifying the work of the user or programmer, who creates the application based on the database. The given approach is that at first the procedure, which accepts only quantity of fields in the new frame and creates a new procedure, is started. The new created procedure is one for the frame creation with the given quantity of slots. This new procedure performs all the actions to create and register a new database object, and then it is deleted.

The second approach is implemented using the Transact-SQL language. The Transact-SQL language provides the ability to use the multiple nested Execute statements that allow one to create the complex structures. Also, with the help of the Execute operator, the procedures are created for filling frames and processing the objects introduced into the model.

We highlight several groups of the stored procedures required when working with the tree model.

1. Procedures for model structure transformation (they are dynamically created together with the database).
2. Procedures for editing the model data (they are dynamically created).
3. The data conversion procedures (they are dynamically created together with the database).
4. Procedures attached to data (they are dynamically created).
5. Procedures for monitoring and maintaining the integrity of the model (they are created together with the database).

We consider application of the data conversion procedures and procedures attached to the data. To account for these procedures, a procedure pointers table is created for each tree with the following columns: PR identifier and Procedure name.

The stored procedure identifier is placed in the “Stored procedure” field and the table key is placed in the “Parameters” field of the slots description table. The slots description table consists of the following columns: PHR identifier, Parameter 1, Parameter 2, ..., Parameter N.

4 Transact-SQL application to implement the logical output function

We distinguish the following variants of using the Transact-SQL language to implement the function of logical inference in the multiagent resource conversion process model.

1. Intelligent agent output mechanism (algorithm) is fully or partially executed as a stored procedure and is called by the scheduler each time.
2. Parts of the intelligent agent rules contain either requests to the knowledge base in the Transact-SQL language or references to stored procedures that implement search and (or) calculation functions.
3. Transact-SQL language is used to solve search and computation problems in the frame-based system, since the models of real multiagent resource conversion systems have large dimension.

5 Application of the object-oriented approach and the UML language to the visual builder of expert systems and the inference engine designer

The UML class diagram is used as the basis for description of the frame-concepts structure when constructing a conceptual domain model. Further, we implement description of the conceptual graphs and filling the obtained conceptual model with the data form the knowledge base [1].

A sequence diagram of the UML language is used to implement the visual builder of the expert system inference engine. The sequence diagram graphically describes the sequence of the called methods between classes when solving a specific problem (script). This approach allows one to visually (in the form of a flowchart) describe the progress of the problem solution: the sequence of procedure calls (methods or daemons) from one frame to another.

The described approaches form the basis for implementation of the frame-based expert system in the Bpsim.MSN decision support system, which is the product of the Bpsim family [3]. The Bpsim family also includes a dynamic situation modeling system Bpsim.MAS [2].

The BPsim.MSN system provides development of the intelligent agents for finding solutions with the help of Wizard design technology. Here, we use the sequence diagrams based on the UML language and the database management language Transact-SQL. Search diagrams, which are an extension of the UML sequence ones, are used to implement the visual builder of the frame-based expert system output mechanism.

When the decision search diagram is processed, the mechanism of the built-in interactive expert system is used. Namely, the user answers a number of system questions in the dialog mode and specifies the data values required to solve the problem. Forms of dialogue that appear during interactive expert system work are preliminarily designed by the analyst at the stage of the description of the domain classes and decision search diagrams.

6 Application of the frame-based expert system to analysis of the processes of network communication equipment replacement

The equipment replacement problem is connected with the generation and estimation of the alternative plans for the equipment replacement with taking into account the work restriction for up to 20 days. A class diagram has been constructed and filled with the input data in the Bpsim.MSN system.

The decision search diagram of the equipment replacement problem is shown in Fig. 1.

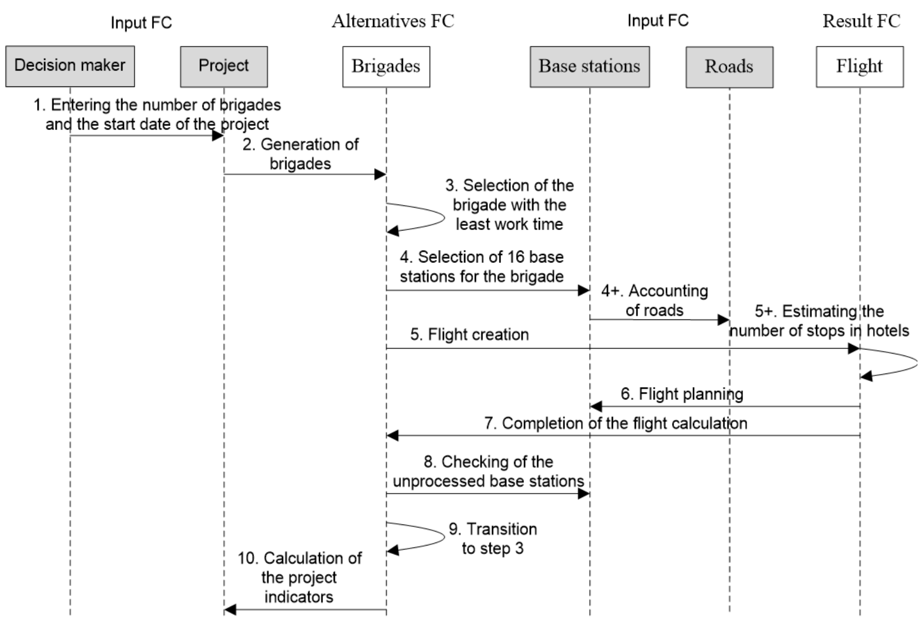

Fig. 1. Decision search diagram of the equipment replacement problem

We consider the application of the Bellman method [4] to analyzing alternative solutions obtained for the intelligent agent by using the frame-based expert system. To apply the Bellman method, it is necessary to solve the problem of the decision search diagram convertation into a classical decision tree. This requires performing the following actions.

1. Classify frames on a frame network in three groups.
   (a) Group 1 consists of the input frame-concepts (FC), which include frames containing the initial information for the problem solving. These frames are not displayed as nodes during the transition to the classical decision tree. For the equipment replacement problem, the Group 1 consists of the following frames (see Fig. 1): the "Project" frame that contains the initial

project conditions, the “Base stations” frame containing the description of the base station equipment for replacement, and the “Roads” frame containing the description of the road mesh matrices.
   (b) Group 2 consists of the alternative FC. For the equipment replacement problem, the group 1 consists of the “Brigades” frame.
   (c) Group 3 consists of the FC solutions (or result FC); these frames contain calculations of alternative solutions, and are subsequently used to solve the problem of the multistage decision making. For the equipment replacement problem, the group 1 consists of the “Flight” frame.
2. A fictitious initial and final tree nodes should be introduced in order to effectively apply the Bellman method (analysis of alternatives). The final node contains zero transition weights.
3. The characteristic describing the weights of the decision tree arcs should be chosen when solving practical problems. The weights of the arcs correspond to the labor intensities (costs) when passing from one node to another. For the problem considered, the duration of the project implementation in days has been chosen as the weights of the arcs.

The application of the Bellman method and the frame-based expert system consists of the following steps.

1. Select a search strategy (search in width or search in depth).
2. Motion on the classical decision tree in accordance with the strategy of search and signifying the frames values of the current search branch. The inference engine is allowed to access any attached frames of the network.
3. Evaluation of design constraints and alternatives exclusion for the current search step. The alternative paths exclusion is used to reduce the search space.
4. Application of the Bellman method after calculating all alternative variants of the search tree (signifying all frames belonging to the given decision tree).
5. Simulation is used in the Bpsim.MAS system for detailed calculations of the business cases.

The results of transformation of the object diagram of the decision search into the classical decision tree using the frame-based expert system and the results of calculations of the equipment replacment problem are shown in Fig. 2.

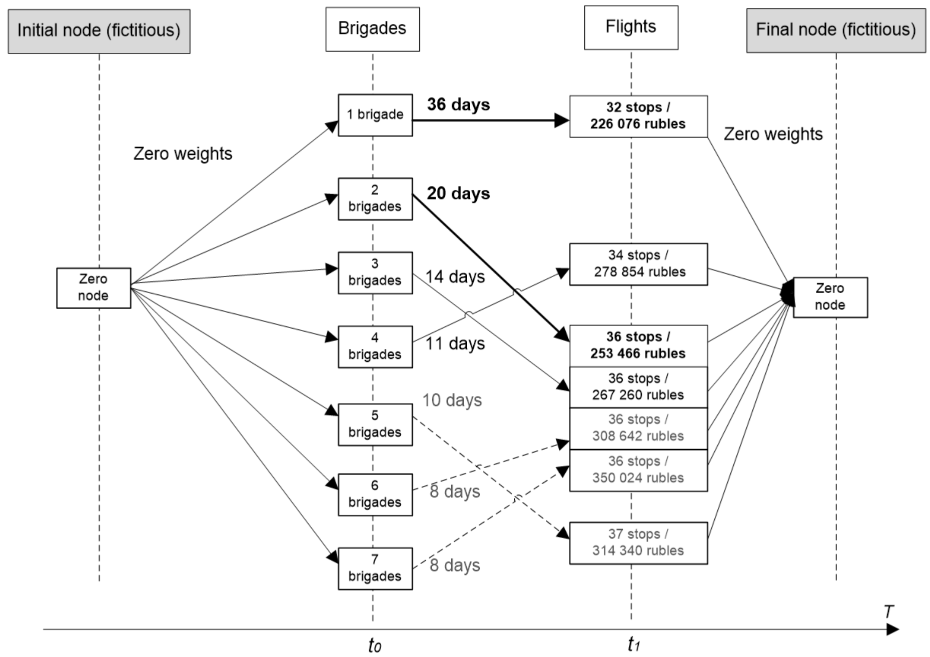

Fig. 2. Decision tree of the equipment replacment problem

Application of the Bellman method allows solving the problem of alternatives estimation. Taking into account the constraints, the optimal solution of the equipment replacement problem for the project duration criterion is the attract of 4 brigades, and for the cost criterion the optimal solution is the attract of 3 brigades.

7 Conclusion and future work

An approach to creating a frame-based system based on the relational database is developed. The advantage of the proposed solution is the use of the Transact-SQL language for designing the domain model in the form of the frame-system, constructing a conceptual domain model, inputting knowledge and data, implementing the inference mechanism, and querying to the knowledge base. This factor reduces the requirements for the skills of system programmers, analysts, and knowledge engineers that support the operability of the frame-based system, and also automates their work. Using the industrial database to store information from the knowledge base allows integrating the frame-based expert system with the corporate information system and effectively applying it in decision support systems.

The proposed approach is implemented in the Bpsim.MSN decision support system. The Bpsim.MSN system has been used to solve the problem of the alternatives development and evaluation when analyzing the process of the equipment replacement in the communication network. Practical recommendations have been obtained on the formation of the number of brigades with taking into account the optimization of the brigades application cost and the constraint on the performance of work up to 20 days. Recommendations have been received on the formation of 3 brigades for implementation of the network equipment replacement process.

Further work is related to the development and application of the method of joint use of dynamic multiagent modeling and frame-based expert system to solving the problem of resource conversion processes analysis.

Acknowledgments. The reported study was funded by RFBR according to the research project no. 18-37-00183.

References

1. Aksyonov, K., Bykov, E., Aksyonova, O., Nevolina, A., Goncharova, N.: Architecture of the multi-agent resource conversion processes extended with agent coalitions. In: IEE Int. Symposium on Robotics and Intelligent Sensors, 221–226 (2016)
2. Aksyonov, K., Bykov, E., Smoliy, E., Khrenov, A.: Industrial enterprises business processes simulation with BPsim.MAS. In: Winter Simulation Conference, 1669–1677 (2008)
3. Aksyonov, K., Spitsina, I., Bykov, E., Smoliy, E., Aksyonova, O.: Computer-supported software development with BPsim products family integration of multiple approaches. In: IEEE Int. Conference on Information and Automation, 1532–1536 (2009)
4. Chernorutskii, I.: Decision-making methods, St. Petersburg: BHV-Petersburg, (2005)
5. Klebanov, B., Antropov, T., Riabkina, E.: The principles of multi-agent models of development based on the needs of the agents. In: 35th Chinese Control Conference, 7551–7555 (2016)
6. Rzevski, G., Himoff, J., Skobelev, P.: MAGENTA technology: a family of multi-agent intelligent schedulers. In: Int conference on multi-agent systems: Workshop on Software Agents in Information Systems and Industrial Applications (2006)
7. Shvetsov, A.: Models and methods of corporate intellectual decision support systems building, DPhil research paper, Saint Petersburg, (2004)
8. Sowa, J.: Conceptual graphs for a database interface. IBM Journal of Research and Development, Iss. 20(40), 336–357 (1976)
9. Sowa, J. Conceptual structures: information processing in mind and machine, Reading, MA: Addison-Wesley, (1984)
10. Sowa, J.: Knowledge representation: logical, philosophical, and computational foundations, Pacific Grove, CA: Brooks/ Cole Publishing Co, (2000)
11. Ueno, H., Ishizuka, M.: Knowledge representation and usage: translation from Japanese, Moscow: Mir, (1989)
12. Zambonelli, F., Jennings, N., Wooldridge, M.: Developing multiagent systems: the GAIA methodology. ACM Transactions on Software Engineering and Methodology, Iss. 12(3), 417–470 (2003)