

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**

Open access books available

**122,000**

International authors and editors

**135M**

Downloads

Our authors are among the

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



# Decision Support based on Multi-Agent Simulation Algorithms with Resource Conversion Processes Apparatus Application

Konstantin Aksyonov, Eugene Bykov,  
Leonid Dorosinskiy, Elena Smoliy and Olga Aksyonova  
*Ural Federal University  
Russian Federation*

## 1. Introduction

The chapter focuses on one of possible approaches to decision support problem, which is based on multi-agent simulation modelling. Most decision support cases generally consist of a set of available alternatives definition, estimation and selection of the best one. When choosing a solution one needs to consider a large number of conflicting objectives and, thus, to estimate possible solutions on multiple criteria.

Multiple objectives become now one of decision support problem regular features, which brings to that decision making people need to estimate multiple forces, influences, interests and consequences that may result from a decision.

We can name increase in performance and reliability, decrease in cost and risk, estimate of system sensitiveness to factor change, structure optimization and much more among the problems of existing organizational technical systems operation and the new ones design. Difficulties in understanding the cause-and-effect dependencies of a complex system lead to ineffective system organization, errors in its design, large costs of error correction. Today modelling becomes the only practically effective tool of an optimal or acceptable decision search in a complex system, the tool for responsible decision making.

Use of situational models in process control facilitates efficiency and taken decisions quality growth, decision taking time decrease, resource consumption rationalization. Dynamic situations modelling systems design is one of perspective directions of decision support systems development. Currently multi-agent systems area is under major research; one of its features is agent collaborations interaction, such agents identify decision making people. An important area of multi-agent technologies application is simulation. Multi-agent systems engineering approaches can be distinguished into two types:

1. Based on object-oriented methods and technologies and
2. Use of traditional knowledge engineering methods.

Object-oriented method extensions and multi-agent systems engineering technologies are developed in methodologies of first type. Certain CASE (Computer-Aided Software Engineering) tools support information systems development based on object-oriented

methods (All Fusion, Rational Rose). Object oriented agent behaviour definition language UML-RT is utilized in multi-agent simulation system AnyLogic. Methodologies of second type are built on basis of traditional knowledge engineering methods extension. An actual task is development of dynamic situations modelling system, based on object-oriented technologies.

The main idea of the chapter is situational, multi-agent, simulation and expert modelling methods and tools integration in order to increase the decision support effectiveness in situational control of resource conversion.

## 2. Multi-agent resource conversion processes in organizational-technical systems

Analysis of various resource conversion processes (RCP), including industrial, organizational-technical, business-processes, etc., reveals their following features.

1. Objects of organizational-technical systems have complicated structure and behaviour algorithms, rely on multiple parameters, which obviously results in their models complexity; this requires complicated hierarchical modular patterns definition at model design stage, as well as intrasystem processes definition utilization (Sovetov & Yakovlev, 2001).
2. Process may be represented accurately within elemental resource conversion operations at lowermost decomposition levels (Aksyonov & Goncharova, 2006).
3. Information flows parameters estimation, defined normalized data structures setting for decision support might be complicated enough in organizational-technical systems. Such systems are characterized with probabilistic behaviour, caused by multiple objective and subjective factors impact, high unsteadiness of information sources and targets, frequent changes in documents provision nomenclature and form, weak routes and information processing methods formalization within an organization, lack of qualified professionals in IT area. All this results in intellectual decision support system requirement, which is able to undertake all formalized functions of executives and provide considerable support in hardly-formalizable tasks. Organizational tasks in many cases have no precise solution algorithms, but are solved within the scope of certain scenarios, generally known by executives, but varying in every specific situation. Such scenarios can hardly be defined with algorithmic models; knowledge representation might be more adequate, as long as it allows behaviour rules modification and provides logical output based on knowledge base contents (Shvetsov, 2004).

In this research, we will define the resource conversion process (RCP) as the process of an input conversion (resources necessary for process execution) into output (products – outcomes of process execution). The main objects of discrete Multi-agent RCP are presented on Fig. 1: operations (*Op*), resources (*Res*), control commands (*U*), conversion devices (*Mech*), processes (*PR*), sources (*Sender*) and resource receivers (*Receiver*), junctions (*Junction*), parameters (*P*), agents (*Agent*). Process parameters are set by the object characteristics function. Relations between resources and conversion device are set by link object (*Relation*). The agents existence resumes availability of the situations (*Situation*) and decisions (action plan) (*Decision*).

Agents control the RCP objects. There is a model of the decision-making person for every agent. An agent (software or hardware entity) is defined as an autonomous artificial object,

demonstrating active motivated behaviour and capable of interaction with other objects in dynamic virtual environment. In every point of system time a modelled agent performs the following operations: environment (current system state) analysis; state diagnosis; knowledge base access (knowledge base (KB) and data base (DB) interaction); decision-making. Thus the functions of analysis, situations structuring and abstraction, as well as resource conversion process control commands generation are performed by agents (Fig. 2).

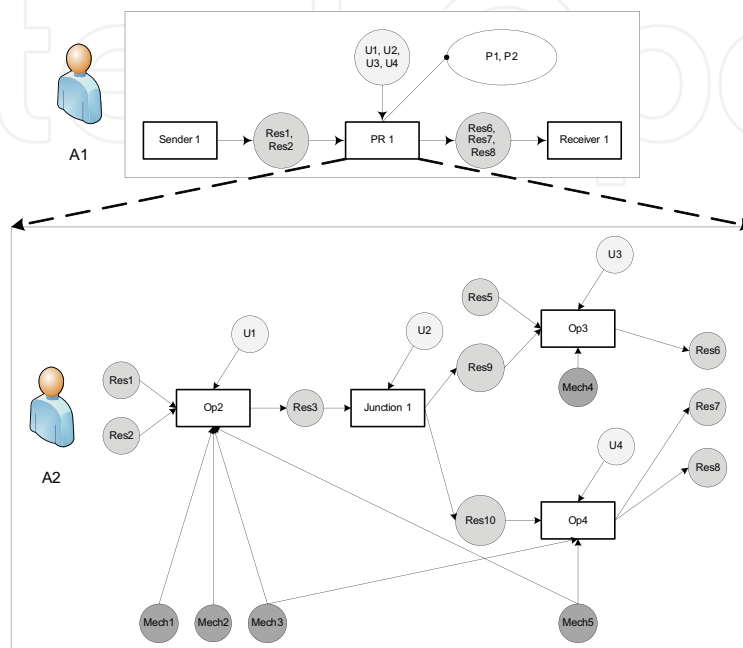


Fig. 1. Hierarchical multi-agent RCP

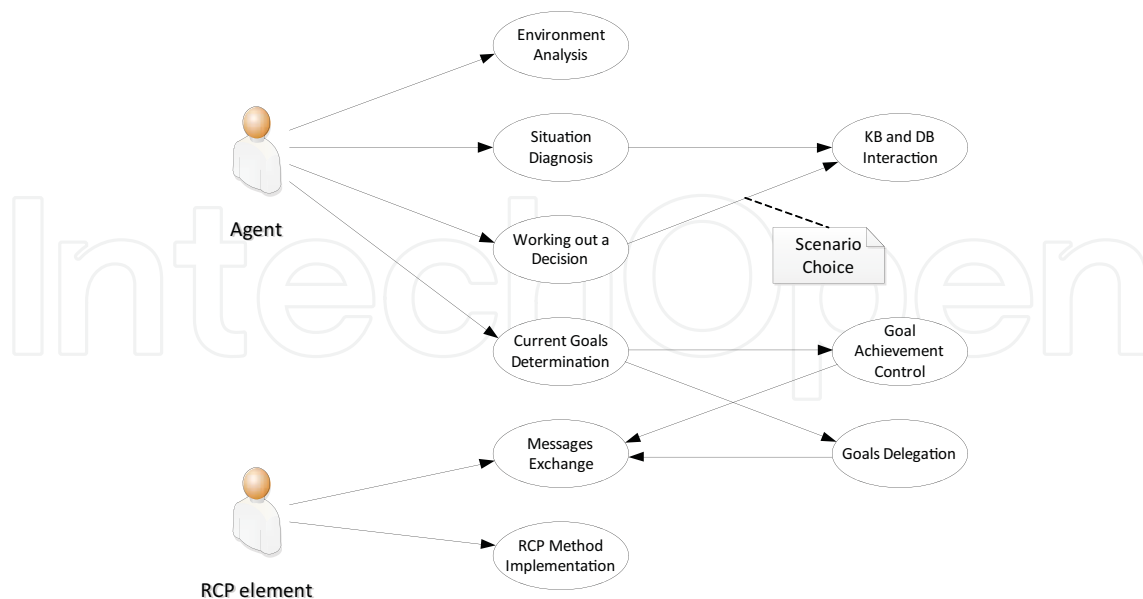


Fig. 2. Use-case diagram determines relations between agents and RCP elements

More detailed information about multi-agent resource conversion processes apparatus is presented in (Aksyonov et al., 2008a).

Current state of multi-agent resource conversion processes dynamic situations modelling systems is defined in next section.

### 3. Current state of dynamic situations modelling systems

Dynamic situations modelling systems area state analysis reveals unavailability of resource conversion processes oriented systems. Nearest functionality analogues include simulation and expert modelling tools, particularly real-time expert system G2 (G), multi-agent simulation system AnyLogic (L), business-processes modelling system ARIS (T), simulation system Arena (A). Results of these tools comparative analysis are presented in Table 1.

Comparison criteria	T	G	L	A
Subject area conceptual model design	○	○	○	○
RCP description language	●	●	●	●
Systems goals definition:				
- Graphical	●	●	○	○
- Balanced scorecard based	●	○	○	○
- Hierarchical process model	●	●	●	●
Commands description language	○	●	○	○
Use of natural language for model definition	○	●	○	○
Multi-agent modelling availability				
- "Agent" element	○	○	●	○
- Agents behaviour models	○	○	●	○
- Agent's knowledge base support	○	○	○	○
- Message exchange language	○	○	○	○
Simulation modelling	●	●	●	●
Expert modelling	○	●	○	○
Situational modelling	○	●	○	○
Object-oriented approach				
- Use of UML language	●	○	○	○
- Object-oriented programming	○	●	●	○
- Object-oriented simulation	○	●	●	○
- Subject area conceptual model and object-oriented simulation integration	○	○	○	○
Retail price, ths \$	50	70	8	4

Table 1. Modelling tools comparison

As we can see, all current systems lack support of some features that might be useful in effective simulation. For example, problem domain conceptual model design and agent-based approach implementation is limited. Another disadvantage of two most powerful systems, ARIS ToolSet and G2, is a very high retail price, which might stop a potential customer. Also systems such as AnyLogic and G2 require programming skills from users. So, from a non-programming user's point of view, no system has convenient multi-agent resource conversion process definition aids. Again, AnyLogic and G2 make use of high-level programming language, which results in these products being highly functional.

Simulation, situational modelling and expert systems are used in modelling, analysis and synthesis of organizational-technical systems and business processes. Multi-agent resource conversion processes theory (Aksyonov & Goncharova, 2006) may be used for organizational-technical systems definition from decision support point of view, a dynamic component of business processes, expert systems, situational modelling and multi-agent systems. Decision support system development requires selection or development of mathematical apparatus.

#### **4. Hierarchical models of control systems**

Conceptual analysis or knowledge structuring phase is usually bottleneck in intelligent systems design lifecycle. Structuring methodology is close to large-scale systems (Gig, 1981) or complex systems theory (Courtois, 1985), where engineering process is traditionally emphasized (Bertalanffy, 1950; Bouling, 1956). A major contribution to this theory was made by object-oriented analysis classics (Buch et al., 1993). System analysis is closely interwoven with system theory and includes a set of complicated systems (technical, economical, ecological, etc.) oriented research and modelling methods (Chastikov et al., 2003).

Hierarchical approach (Mesarovich & Takahara, 1978) is traditionally used as a methodological approach for formal system definition decomposition into tiers (or blocks/modules) in complex systems engineering and structuring methods. Top hierarchy tiers are populated with the least detailed views, reflecting only the most common features and characteristics of designed system. Detail level increases on lower hierarchy tiers, while the system is no longer regarded as a whole, but in separate blocks (Chastikov et al., 2003).

Each tier introduces own views on the system and its elements. K-th tier element is considered a system for (k-1)-th tier. Tier-to-tier advancement is severely directional and is defined by engineering strategy – deductive descending «top-down» or inductive ascending «bottom-up» (Chastikov et al., 2003).

In relation to processes formalization deductive descending engineering strategy is used in IDEF0, IDEF3, DFD, EPC notations, as well as in dynamic processes hierarchical models design (aggregates, Petri nets, extended Petri nets). Inductive ascending strategy is used in high-level integration system graphs.

High-level integration system graphs (Avramchuk et al., 1988) are used for multi-agent resource conversion processes hierarchical structure definition (Aksyonov & Goncharova, 2006). An example of such hierarchy is presented on Fig. 3.

From dynamic simulation point of view, only those elements are simulated which were not decomposed at system analysis phase. When using system graphs apparatus all data required for simulation is obtained on first step of dynamic system model design (0-level integration).

#### **5. Object-structural approach to control systems models design**

Object-structural approach, offered by T. A. Gavrilova (Gavrilova, 1989), allowed consolidation of these two, traditionally anticipated, engineering strategies. Strategies synthesis, as well as introduction of iterative returns to previous decomposition tiers, allowed a dual concept, offering wide capabilities at knowledge structuring phase to analysts, together for subject area conceptual and functional structure definition.



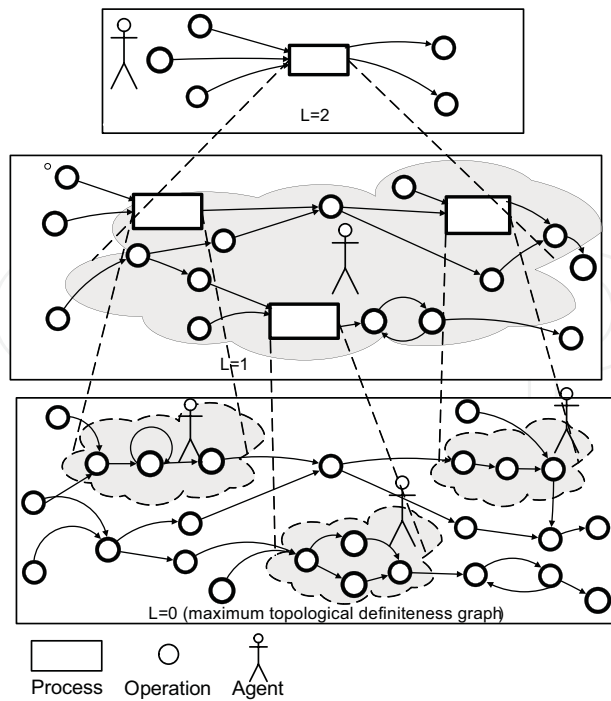


Fig. 3. Multi-agent resource conversion processes hierarchical structure

Fig. 4 illustrates this concept applied to functional structure engineering of an expert system for assistance in multi-service network (further referenced as MSN) development.

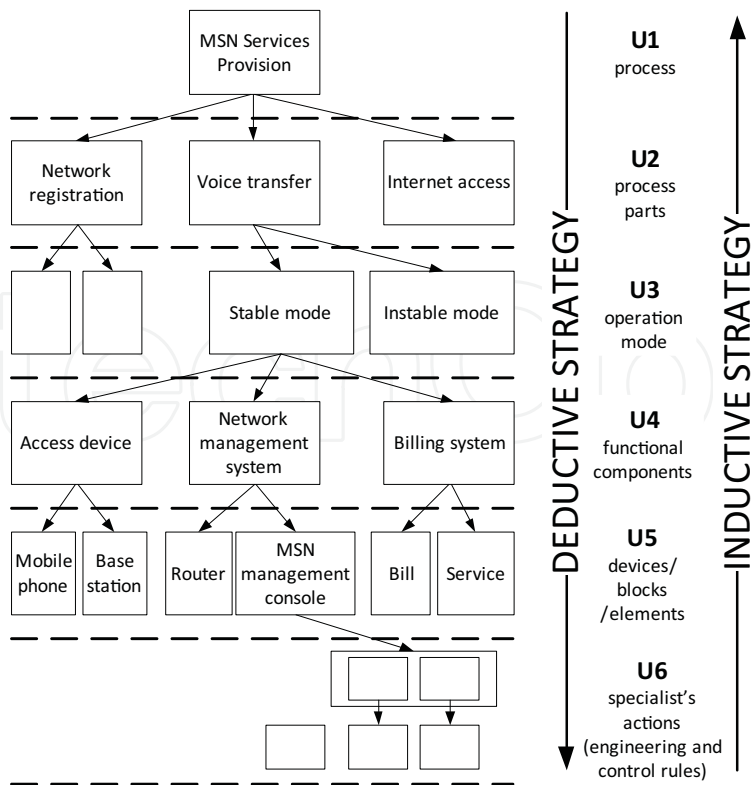


Fig. 4. Dual engineering strategy

## 6. Knowledge representation model

One of the most important problems in intelligent systems engineering is selection and design of subject area knowledge representation models, that allow the easiest native transition from nonformalized knowledge and views to formal models and knowledge base. Knowledge elicitation and acquisition process might be very complicated in intelligent systems engineering.

Knowledge structuralization complexity is revealed in requirement for subject area model, that allows the most adequate transition to technical implementation with least effort (Shvetsov, 2004).

Subject area conceptual model needs subject area structure to be defined, available objects and subjects behaviour determined, logical interaction models designed. Minsky defined a frame as a structure for stereotyped (standard) situations representation (Minsky, 1975). This structure is filled with various information: defining objects and events expected in certain situations as well as providing guidelines on use of information, contained in a frame. Main idea is to concentrate all knowledge, related to specific objects or events class, in a common data structure, but not to distribute it between a multitude of small structures like logical formulae and productive rules. Such knowledge is either concentrated within the structure itself or available from the structure (e.g. stored within a related structure) (Jackson, 1998).

Each frame is associated with various information (including procedures), e.g. information defining frame use, expected results of frame execution, directions for actions when expectations are not fulfilled, etc.

Frame-based approach reveals the following advantages: frame concept is natively integrated with subject area conceptual modelling; frame structures are easily defined within object-oriented design; inheritance capabilities are effectively supported; subject area hierarchical representation is provided. Thus, frame-based approach selection might be a reason for object-oriented approach and object programming languages application in dynamic situations modelling system development. Such approach minimizes costs for software development as well.

Analysis of Shvetsov's object-oriented design and programming-related research reveals three main model classes, implementing class-based representation formalism: semantic networks and frames-based models; database theory and semantic data patterns-based models; abstract data types-based models. Frame-based languages extend semantic and object-oriented data models capabilities, which substantiates their application and further research in this area.

This research makes use of frame approach, based on frame-like structures association with J.F.Sowa's conceptual graphs constructions (Sowa, 2000), in order to design subject area conceptual model and achieve software development costs reduction. Active and passive frames distinction and agent behaviour consideration are among approach advantages.

Basic frame-concept (FC) construction is presented on Fig. 5 (left) *Frame name* is a unique identifier, used within subject area conceptual model.

Frame-concept level application information contains informal verbal definition of available frame-concept application situations, behaviour scenarios, selection features, etc. Dynamic subject area components and agents behaviour is defined in behaviour scenarios structure, containing scenario selection block that allows current frame alternative behaviour options generation.



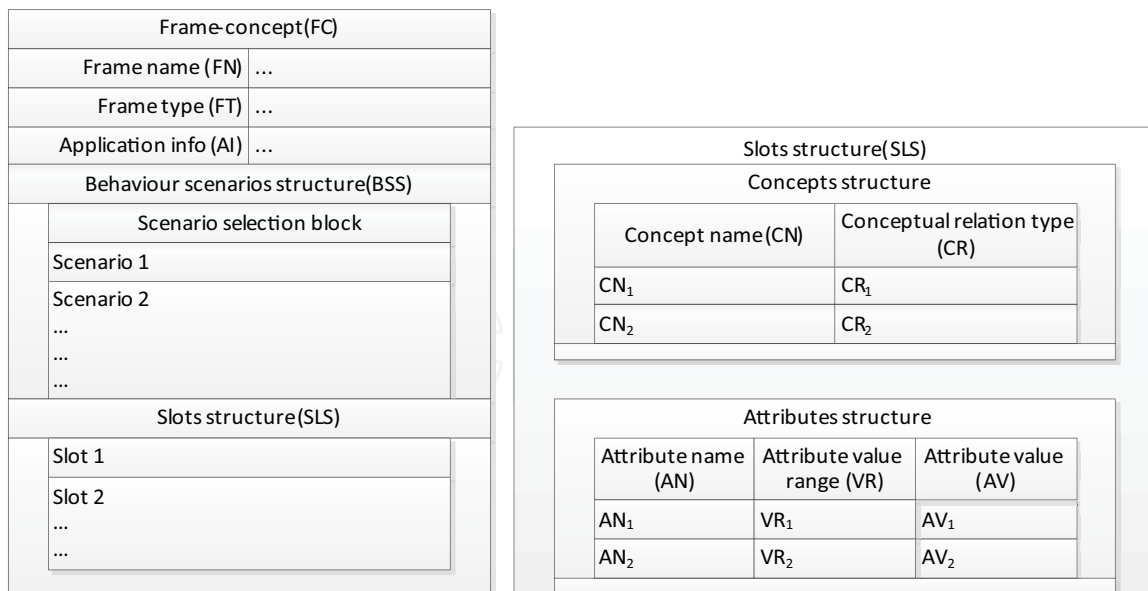


Fig. 5. Frame-concept (left) and slots (right) structure

Slots structure consists of two structures: concepts structure and attributes structure (Fig. 5, right). Concepts structure contains list of frame-concepts, in some way embedded into or descendent from current frame-concept; relation type is indicated in «conceptual relation» field, being relation of specific concept (SC) SC<sub>i</sub> to current frame-concept FC, SC<sub>i</sub> is i-th frame-concept name. Frame-concepts are combined into conceptual graphs structures to form subject area logical organization.

Conceptual graph is a bipartite graph with two types of nodes: concept nodes or conceptual nodes and conceptual relations nodes. Thus, frame-semantic knowledge representation is used.

Frame-concept model is defined in the following way:

$$\begin{aligned}
 FC &= \langle FN, FT, AI, BSS, SLS \rangle \\
 SLS &= \langle CS, AS \rangle \\
 CS &= \{(CN_1, CR_1), (CN_2, CR_2), \dots, (CN_n, CR_n)\} \\
 AS &= \{(AN_1, VR_1, AV_1), (AN_2, VR_2, AV_2), \dots, (AN_m, VR_m, AV_m)\}
 \end{aligned} \tag{1}$$

- where  $FC$  - frame-concept,  $FN$  - frame name,  $FT$  - frame type,  $AI$  - application information,  $BSS$  - behaviour scenario structure,  $SLS$  - slots structure,  $CS$  - concepts structure,  $AS$  - attributes structure,  $CN_n$  - concept name,  $CR_n$  - conceptual relation,  $AN_m$  - attribute name,  $VR_m$  - attribute available values range,  $AV_m$  - attribute value.

Thus, frame-concept and conceptual graph-based approach to subject area definition allows frame-semantic knowledge representation model use.

Multi-agent resource conversion processes architecture design is described in next sections.

## 7. Multi-agent resource conversion process model

Multi-agent resource conversion process model was developed on the basis of several approaches integration: simulation and situational modelling, expert and multi-agent systems, object-oriented approach.

The resource conversion processes simulation core is built on the widespread mathematical schemas of dynamic processes description (Petri nets, queuing systems, and system dynamics models). However, it is difficult enough to present all the features of the RCP with the help of specified models (Aksyonov & Klebanov, 2008). This was the reason for further RCP model extension.

Simulation engine algorithm of agent-containing model (Fig. 6) consists of the following main stages: current point of system time identification  $SysTime = \min_{j \in RULE} T_j$ ; agent actions processing (state diagnosis, control commands generation); conversion rules queue generation; conversion rules execution and operation memory state (i.e. resources and mechanisms values) modification. Simulator makes use of expert system unit for situations diagnosis and control commands generation (Aksyonov et al., 2008a).

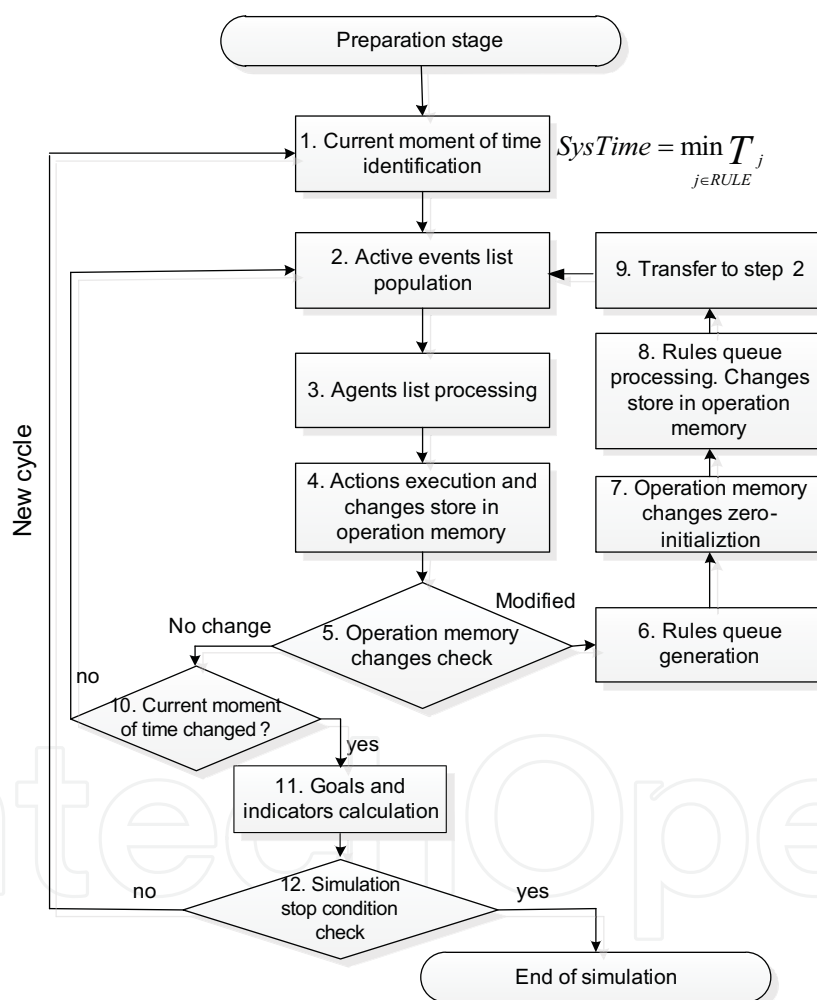


Fig. 6. Simulation algorithm

Each agent possesses its knowledge base, set of goals that are needed for behaviour configuration setting, and priority that defines agent order in control gaining queue. Generally in case of any corresponding to agent's activity situation an agent tries to find a decision (action scenario) in the knowledge base or work it out itself; makes a decision; controls goals achievement; delegates the goals to its own or another agent's RCP objects; exchanges messages with others.

Multi-agent resource conversion process agent may have hybrid nature and contain two components:

- Intelligent (production rules and/or frame-based expert system access)
- Reactive (agent activity is defined on UML activity diagram)

Right selection and implementation of multi-agent architecture is a key factor in multi-agent object oriented decision support system design.

## 8. Multi-agent RCP extension with InteRRaP architecture

Two main agent architecture classes are distinguished. They are:

1. **Deliberative** agent architecture (Wooldridge, 2005), based on artificial intelligence principles and methods, i.e. knowledge-based systems;
2. **Reactive** architecture, based on system reaction to external environment events.

All currently existing architectures cannot be defined as purely behavioural or purely knowledge-based. Any designed architecture is hybrid, offering features of both types.

Multi-agent resource conversion process architecture is based on InteRRaP (Muller & Pischel, 1993; Aksyonov et al., 2009a) architecture, as the most appropriate for subject area. InteRRaP architecture represents an aggregate of vertically ordered levels, relating to common management structure and using common knowledge base. Architecture consists of blocks: external environment interface, reactive sub-system, planning sub-system, cooperation with other agents, and hierarchical knowledge base. External environment interface defines agent capabilities in external environment objects and events perception, influenceability, and means of communication. Reactive sub-system utilizes agent capabilities in reactive behaviour, as well as partly utilizes agent knowledge of procedural kind. It is based on «behaviour fragment» concept as reaction draft in some standard situation. Planning sub-system contains planning mechanism that provides agent local plans capability (not related to co-operative behaviour). Cooperation sub-system is responsible for building co-operative behaviour plans, focusing on certain joint goals, or fulfilment of obligations for other agents, as well as implementation of agreements.

In accordance with InteRRaP architecture common concept, multi-agent RCP agent model is represented in four levels:

1. **External environment** model corresponds to the following MRCP elements: convertors, resources, tools, parameters, goals. External environment performs the following actions: generates tasks, transfers messages between agents, processes agent commands (performs resource conversion), alters current state of external environment (transfers situation  $S_n$  into state  $S_{n+1}$ ).
2. **External environment interface** and reactive behaviour components are implemented in form of agent production rules base and inference machine (simulation algorithm).
3. **Reactive behaviour** components performs the following actions: receives tasks from external environment, places tasks in goal stack, collates goal stack in accordance with adopted goal ranging strategy, selects top goal from stack, searches knowledge base. If appropriate rule is located, component transfers control to corresponding resource convertor from external environment. Otherwise, component queries local planning sub-system.
4. **Local planning** level purpose is effective search for solutions in complex situations (e.g. when goal achievement requires several steps or several ways for goal achievement are

available). Local planning component is based on frame expert system. Frame-concept and conceptual-graph based approach is utilized for knowledge formalization.

Subject area conceptual model and agent local planning knowledge base design is based on UML class diagram extension. Semantically this notion may be interpreted as definition of full decision search graph, containing all available goal achievement ways (pre-defined by experts). Current knowledge base inference machine is implemented in decision search diagram, based on UML sequence diagram. Each decision represents agent activity plan. Each plan consists of a set of rules from reactive component knowledge base. Based on located decision, current agent plan is updated. Examination of all available options, contained in knowledge base, generates agent plans library.

If an agent, when processing task or message received from external environment, is unable to locate appropriate rule in its knowledge base (e.g., select an option from several ones), the reactive behaviour component queries plans library, indicating goal (i.e. task to execute, or external environment state to bring into). Planning sub-system searches plans library, selects appropriate plan and places first rule of selected plan into reactive component goals stack.

## 9. Intelligent agent operation algorithm

Special-purpose object-oriented language RADL (Reticular Agent Definition Language) in form of **When-If-Then** construction implemented in agents and multi-agent systems engineering system *Agent Builder* (Reticular Systems, Inc.) (Andreichikov & Andreichikova, 2004) is used as a basis for agent behaviour rules. Mental model includes intentions, desires, obligations and capabilities as well as agent behaviour rules definition. Specific intelligent actions are calculated on the basis of this model. Rule constituents perform the following functions: **When**<...> contains new messages, received from other agents; **If**<...> compares current mental model with rule application conditions; **Then**<...> defines actions, associated with current events, mental model and environment state.

Considering that agent mental model is represented with goal-setting model and **When** function is immediately incorporated into algorithm, the following agent behaviour rules structure will be used in resource conversion processes subject area:

- **Name** <Rule Name>
  - **If** <Message Conditions, RCP Conditions, G\_Ag Conditions>
  - **Then** <G\_Ag Changes, Message Actions, Private Actions>
- where *Message Conditions* - message related conditions; *RCP Conditions* - resource conversion process related conditions; *G\_Ag Conditions* - goal related conditions; *G\_Ag Changes* - agent current goals modifying actions; *Message Actions* - message generation actions; *Private Actions* - convertors and resource related actions (activity plan), targeting set goals achievement.

Rules parts may be represented in form of first-order predicates. We assume that  $n$ -ary predicate on  $A$  set is a  $n$ -ary function, certain on  $A$  set, with the values from  $\{true, false\}$  set. An aggregate of sets with elements from  $A$  ( $a_1, a_2, \dots, a_n$ ), resulting in  $P(a_1, a_2, \dots, a_n) = true$  is a  $n$ -ary relation, corresponding to  $P$  predicate. On the contrary, any  $n$ -ary  $R$  relation on  $A$  set corresponds to  $P$  predicate  $P(x_1, x_2, \dots, x_n)$

$$P(a_1, a_2, \dots, a_n) = \begin{cases} true, & \text{if } (a_1, a_2, \dots, a_n) \in R, \\ false, & \text{if } (a_1, a_2, \dots, a_n) \notin R. \end{cases} \quad (2)$$

**Then** part includes actions and action plans, i.e. either a set of serial-parallel operations, bound to time, or a **Decision**.

Next section presents development principles and technical decisions of designed object-oriented multi-agent resource conversion processes based decision support system, relying on above-stated multi-agent resource conversion process model and multi-agent architecture.

## 10. Multi-agent systems simulation and engineering systems integration

Object-oriented decision support system *BPsim.DSS* ("Business Processes Simulation - Decision Support System") is implemented on basis of dynamic situations modelling system *BPsim.MAS* ("Multi-Agent Simulation"), software engineering system *BPsim.SD* ("Software Designer") and technical economical engineering system *BPsim.MSN* ("Multi-Service Network") integration.

The following program packages are being used during multi-agent resource conversion processes subject area business process modelling and software design ([www.bpsim.ru](http://www.bpsim.ru)), offering a comprehensive solution for business modelling and techno-economic engineering problems, which in turn considerably simplifies and speeds analysts' work:

- *BPsim.MAS* - multi-agent dynamic situations modelling system (Aksyonov et al., 2008a). *BPsim.MAS* offers the following functionality:
  - a. Multi-agent resource conversion process model design;
  - b. Dynamic simulation;
  - c. Experiment results analysis;
  - d. Model- and results-based reporting;
  - e. Experiment data export to Microsoft Office family products.
- *BPsim.SD* - Software Developer CASE tool  
*BPsim.SD* offers automation on the following phases of software development:
  - a. DFD diagrams design is not automated. As in every CASE tool a DFD diagram needs to be designed manually;
  - b. Use-case diagrams design is fully automated, use-case diagrams are achieved by a transition from a DFD diagram. This process lets us keep our business objects;
  - c. Classes diagram design is partially automated. The core classes' frames are generated automatically, that greatly simplifies work on the final class diagram. Benefit is estimated in 10-15%;
  - d. Sequence diagram design is semi-automatic.
  - e. Database structure generation is automated.

*BPsim.SD* offers an opportunity of forms design. This allows the end-user place the controls on the form as he wants them to be positioned. Some of the controls can be associated with data on the phase of GUI design before passing the project to the developers. After this phase a developer receives GUI forms in an appropriate format, i.e. the forms are saved in a software development file format (Aksyonov et al., 2008b).
- *BPsim.MSN* - techno-economic engineering intellectual system (Aksyonov et al., 2009b) automates the following functions:
  - a. Subject area conceptual model engineering;
  - b. Filling the knowledge base data;
  - c. Decision search diagrams design, setting up dialog-based expert system;
  - d. Decision search.



- *BPsim Wizard Technology* – a framework of intelligent software assistants for step-by-step model definition. A wizard is a dialog-based program assistant targeting information integration and conversion from one system (BPsim.DSS / BPsim.MAS / BPsim.SD) to another. BPsim Wizard Technology performs the following functions:
  - a. Transfers information between simulation, decision support and software engineering modules in the framework of a single complex problem;
  - b. Simplifies a non-programming user experience when getting started with BPsim products family;
  - c. Validates data on various stages of simulation model design, subject area conceptual model and information system engineering.

Various tools and methods use on all stages of organizational technical systems analysis and synthesis and their support by BPsim products is presented in Table 2.

No	Stage	Tool	Support in BPsim
1.	Processes definition	IDEF0 notation	SD, MSN
		DFD notation	SD, MSN
		UML use-case diagram	SD, MSN
		Multi-agent resource conversion processes notation	MAS
2.	Software engineering	DFD use-case, class, sequence diagrams	SD, MSN
3.	Knowledge formalization	Semantic networks	MSN
		Frames	MSN
		Production rules	MAS
4.	Decision support	Simulation	MAS
		Multi-agent simulation	MAS
		Situational control	MAS, MSN
		Dialog-based expert systems	MSN

Table 2. Methods used in BPsim products

Recently a model and algorithm that are based on transition from multi-agent resource conversion process model to information system model and represented with frame-based semantic network have been developed. A choice of frame-based semantic network as a knowledge representation model facilitates further technical implementation of transition algorithms, since the frame concept can easily be combined with object-oriented design. The transition is implemented on the basis of a dialog-based expert system.

A data processing function, unidirectional or bidirectional, confirms to any converter while automating enterprise processes. The functions include generation, reception, transmission, modification, deletion, etc.

Information system engineering problem domain conceptual model allows the demonstration of the structure of the information system and interconnections between its components. First level of semantic network comprises nodes corresponding to information system database and software. Further these nodes are decomposed up to user interface variables and objects and database tables and stored procedures.



Much more information of real life is required for information system engineering. This must be taken into consideration in data analysis, when data is structured according to certain rules. Each node of resource conversion process semantic network can be represented with corresponding class that possesses its properties and methods for the engineering method implementation.

Fig. 7 demonstrates transition algorithm from resource conversion process problem domain “Resource” object to problem domain elements presented in a form of decisions search visual diagram, built on sequence diagram with integrated dialog-based expert systems apparatus.

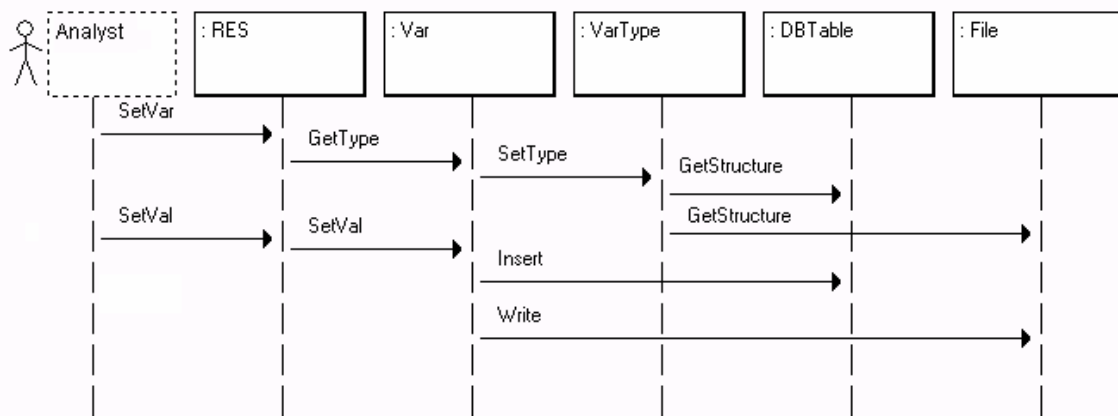


Fig. 7. “Resource design” decision search diagram

BPsim.DSS agent model is represented with four levels in compliance with InteRRaP architecture general concept. External interface and reactive behaviour components together with external environment model are implemented in BPsim.MAS tool. Local planning component is based on BPsim.MSN expert system module. Expert system shell visual output mechanism builder is based on decision search diagrams (UML sequence diagram extension) and presented on Fig. 8. Cooperation level is based on both modules.

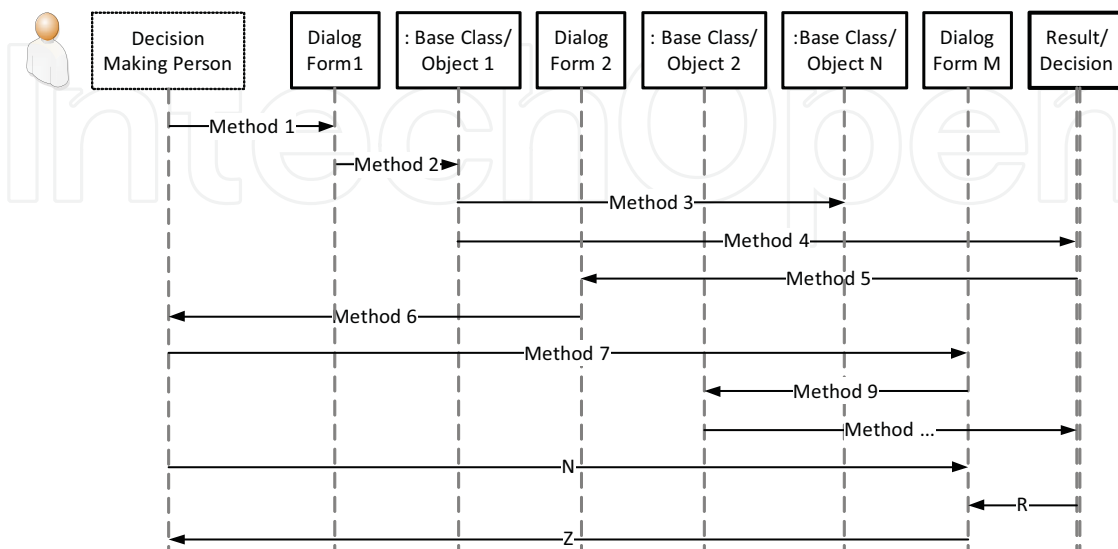


Fig. 8. General decision search diagram in decision support system BPsim.DSS

An example, illustrating decision search diagram workflow, is presented on Fig. 9. For simplification the dialog form classes are not shown on the diagram. The example illustrates work of expert system for real estate agency. The figure shows available house/apartment search in the database on the basis of user set criteria. The search is run in form of decision search diagram.

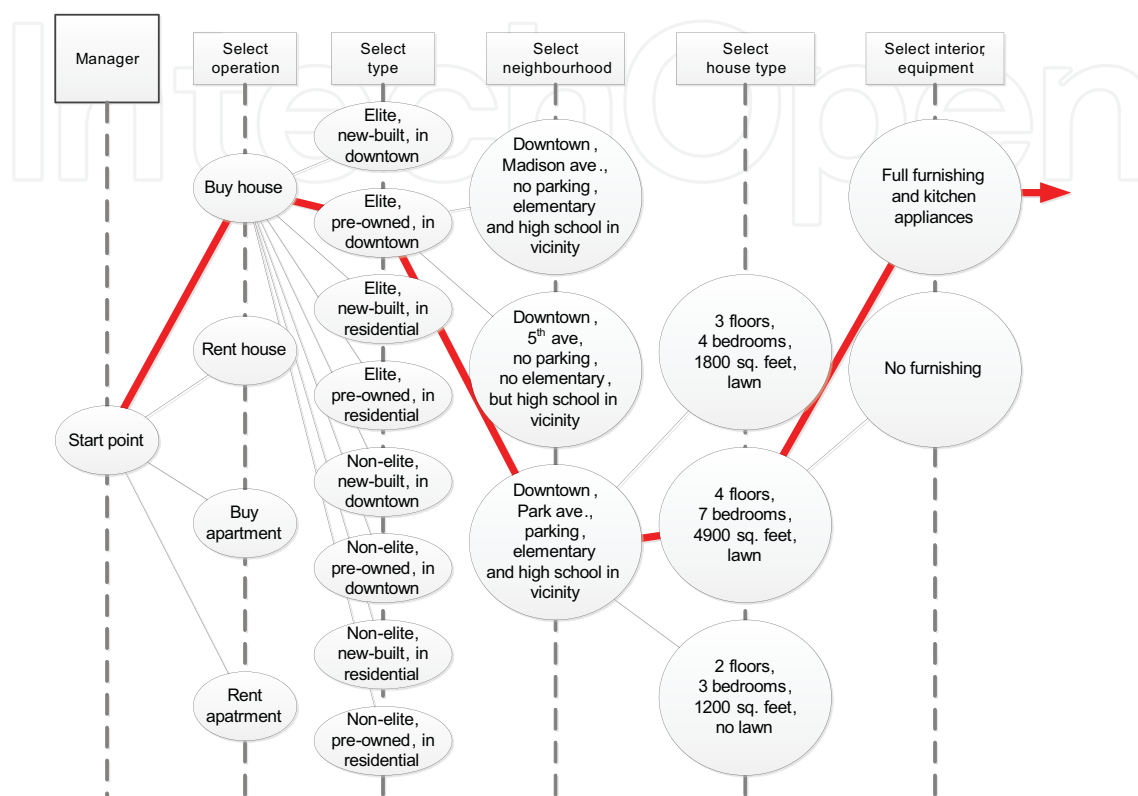


Fig. 9. Decision search tree for decision search diagram

Scheme, presented on Fig. 10, shows interaction of separate units during agent activity within BPsim.MAS and BPsim.MSN integrated scope. Basic principles and separate agent activity stages were mentioned above in section, devoted to InterRaP architecture conceptual model.

Object-oriented decision support system BPsim.DSS allows the following features implementation:

1. Subject area conceptual model definition
2. Multi-agent resource conversion process dynamic model design
3. Dynamic simulation
4. Experiment results analysis
5. Reporting on models and experiment results
6. Data export to MS Excel and MS Project

Decision support system visual output mechanism builder, based on decision search diagrams (Fig. 8), as well represents agent knowledge base, based on frame-concepts. So, agent knowledge base may be defined in two ways: productive (see Fig. 12 later on) and frame-concept - based (see Fig. 15 later on).

There are several examples demonstrating BPsim.DSS system application. They are presented in the next section.

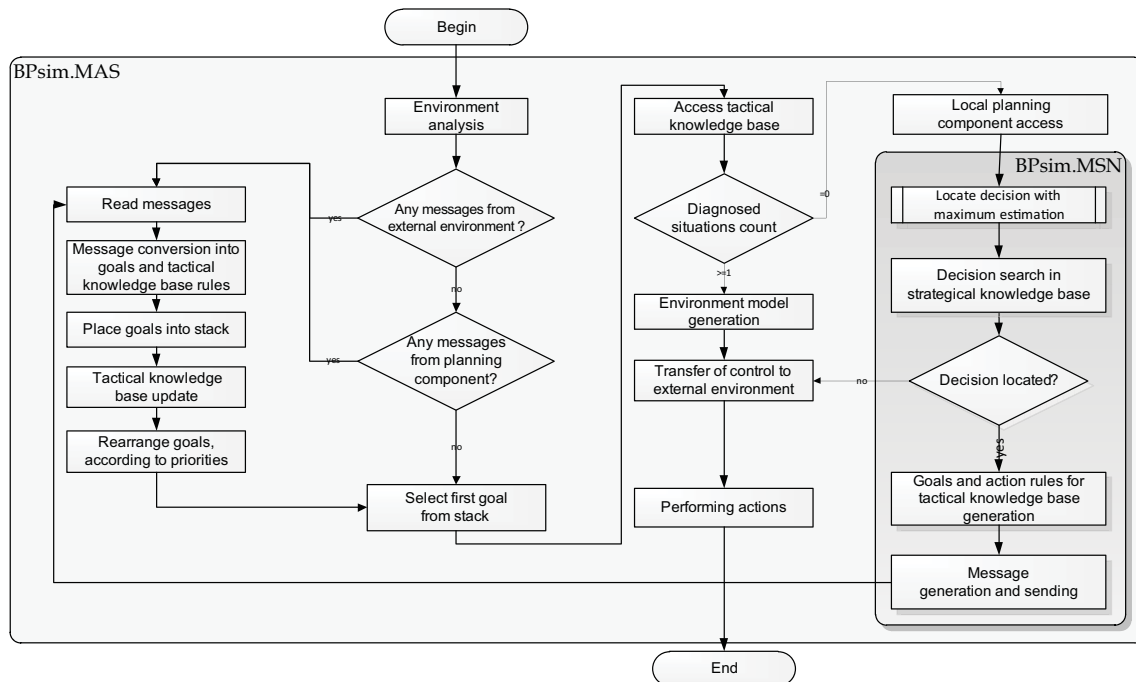


Fig. 10. Intellectual agent activity algorithm

## 11. BPsim.DSS system application

### 11.1 BPsim.DSS application to IT projects management

Decision support system BPsim.DSS was used on various stages of Ural State Technical University Common Information System (CIS) development and deployment, starting with educational process analysis stage, performing re-engineering, and ending with separate CIS units deployment efficiency estimation.

Model of an agent (decision making person), controlling software development process in Ural State Technical University, was developed in decision support system BPsim.DSS. Model consists of simulation model "Educational process software development" and decision support models, including the main model "CIS implementation options selection". Model knowledge base contains information on networking, hardware and software, information systems, IT-projects, teams of IT-specialists.

Expert system module is used for project alternatives and effective alternative search algorithms knowledge base development. Simulation model is used for separate project stages monitoring, detection of errors and conflicts, occurred on initial planning stage, solution of vis major, that happens during development project control and CIS deployment. Simulation model is based on Spiral model of software lifecycle and is designed in BPsim.DSS.

Currently AS-TO-BE model data is implemented in CIS program modules and deployed in Ural State Technical University. Due to "Contingent traffic" process improvement and automation dean's office employees work efficiency was raised by 27%, student desk employees work efficiency was raised by 229%. Deployment economical effect is estimated by about 25 thousand euro per calendar year. Economical effect is achieved in shortening and automation of unnecessary document processing stages, information double input prevention and employee load decrease.

Table 2 presents efficiency of BPsim use compared to the use of an average CASE tool. Data is based on the CIS development, during which the following diagrams were designed in order to build the final product: 25 DFD diagrams, 14 use-case diagrams, 1 classes diagram and 18 sequence diagrams. Table makes use of experimental statistical data, acquired during development of other projects, which include average times of a certain diagram type design time. Thus, an average time of a single use-case diagram design is 40 minutes, for classes diagram it is also 40 minutes and 90 minutes for a single sequence diagram.

Stage	Average CASE tool	BPsim	Efficiency	Number of diagrams	Benefit, min
<b>DFD diagrams design</b>	Varies, manual	Varies, manual	-	25	-
<b>Use-case diagrams design</b>	40 mins per diagram, manual	5 mins per diagram, automated	35 mins per diagram	14	490
<b>Classes diagram design</b>	40 mins, manual	6 mins, base classes automation	34 mins	1	34
<b>Sequence diagrams design</b>	90 mins, manual	50 mins, semi-automatic	40 mins per diagram	18	720
<b>Database structure generation</b>	5 mins, automated	5 mins, automated	-	N/A	-
<b>GUI design</b>	N/A	Forms designer	+	N/A	N/A

Table 3. Estimation of BPsim efficiency

### 11.2 BPsim.DSS application for industrial enterprise marketing strategy development on the basis of competing agents model

BPsim.DSS was used for multi-agent dynamic model development of Urals Industrial Group, CJSC (further referenced as UIG), plastic window frames construction, installation and maintenance enterprise. The main reason for modelling is UIG behaviour algorithm and pricing strategy development, targeting share of the market growth and transition to higher technological level, increasing enterprise competitiveness. One more reason for simulation is search for optimal market share (or production volume). The general UIG business process is presented on Fig. 11.

Model makes use of the following parameters:

- enterprises (share of product market; sales volume; premises price per square meter; processes timing data);
- competitive environment (number of competitors on market, share of the market, strife intensity, competitors prices, reaction on time and price, estimated competitiveness rate, elasticity of demand on price, demand seasonality, market capacity).

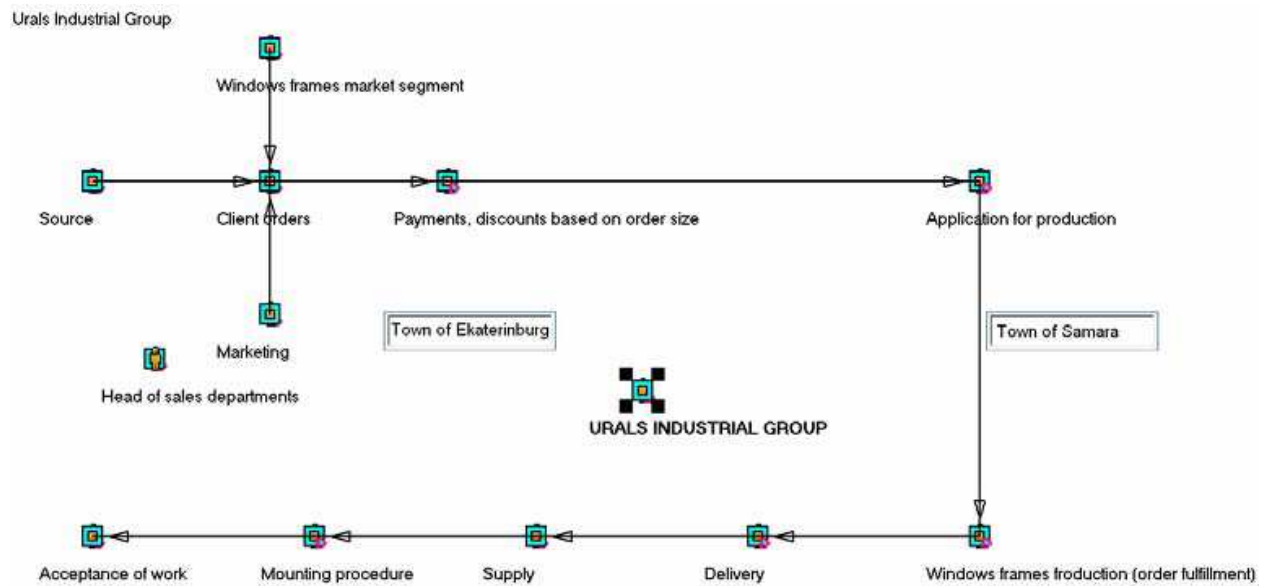


Fig. 11. UIG general business process, presented in BPsim.MAS

The model considers manufacture, sale, installation, servicing processes of the enterprise. Fragment of the model together with single agent’s knowledge base in If-Then form is presented on Fig. 12.

Nr	Situation Name	IF	THEN	Description
2	7/Sharp change in price, during	((t>0)&&(t\$6)=0)&&Rt	((Res60.=Res58)&&(Res58.=R	if moment of time
9	6/Sharp change in price, during	((Res183=6)&&(t>0)	((Res60.=Res58)&&(Res58.=R	price decrease at initi
1	Price decrease in first	(t=0)	((Res60.=Res58)	
2	1) price change	((Res183=1)&&(t>0)	((Res60.=Res58)&&(Res58.=R	if moment of time
3	2) price change - sharp, during	((Res183=2)&&(t>0)	((Res60.=Res58)&&(Res58.=R	if moment of time
4	3) price change - sharp, during	((Res183=3)&&(t>0)	((Res60.=Res58)&&(Res58.=R	if moment of time
5	4.1) price change - increase	((Res183=4)&&(t>0)	((Res60.=Res58)&&(Res58.=R	if moment of time
6	4.2) price change - increase	((Res183=4)&&(t>0)	((Res60.=Res58)&&(Res58.=R	if moment of time
7	4.3) price change - increase	((Res183=4)&&(t>0)	((Res161:=1)	
8	5) price change - sharp, during	((Res183=5)&&(t>0)	((Res60.=Res58)&&(Res58.=R	if moment of time

Fig. 12. UIG frames manufacturing process and agent’s knowledge base

Deployment in Urals Industrial Group focused on effective pricing strategy search, considering passive and active competitors behaviour. A number of experiments considering various agents-competitors behaviour sets (active/passive) were run. Fig. 13 presents the output data, which are various strategies, resulting in two small competitors' displacement from the market.

After a series of experiments a pricing policy, resulting in share of the market growth from 6.6% to 20-22%, was determined. Limiting to current problem the optimal values of processes characteristics were calculated. The projected saving rate from the modelling results implementation is estimated by €1.9 million per year. In addition, optimal values for the number of distribution points and mounting units depending on seasonal demand and applied to the current pricing strategy were calculated in the framework of the current project.

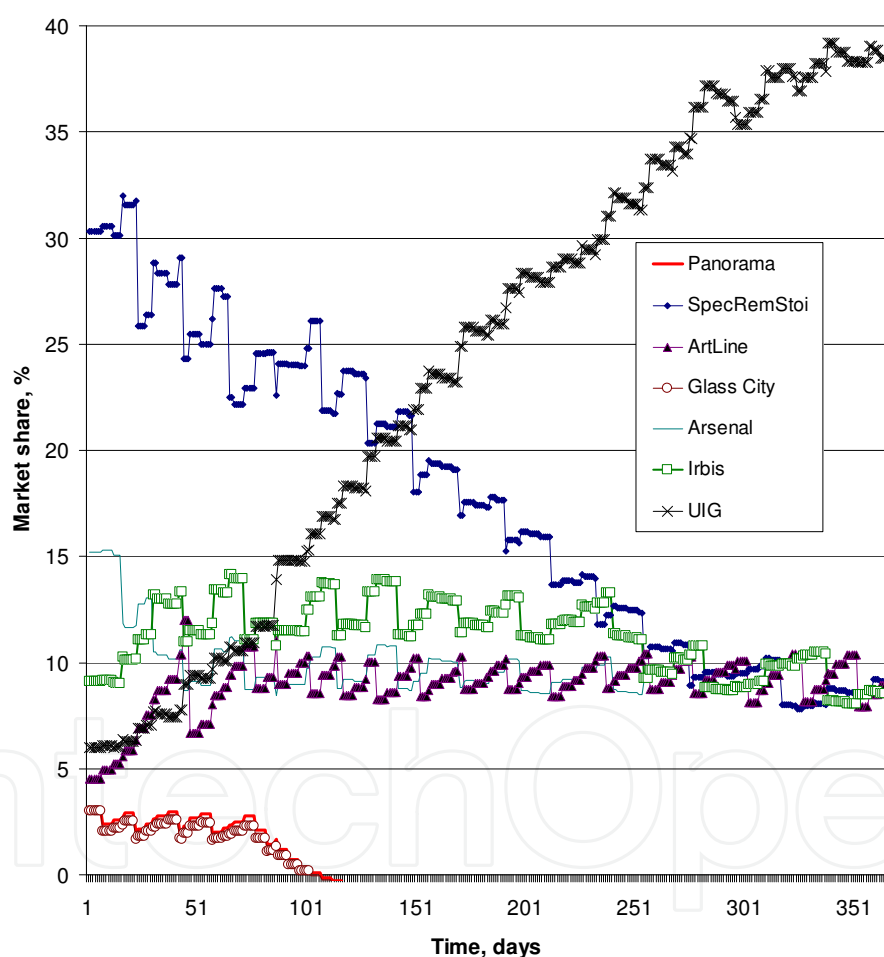


Fig. 13. Competitors displacement from market (competitors' passive behaviour)

### 11.3 BPsim.DSS application to multi-service telecommunication networks technical economical engineering

Another application of BPsim.DSS included multi-service telecommunication network models design and telecommunication services area business processes dynamic simulation (Fig. 14).



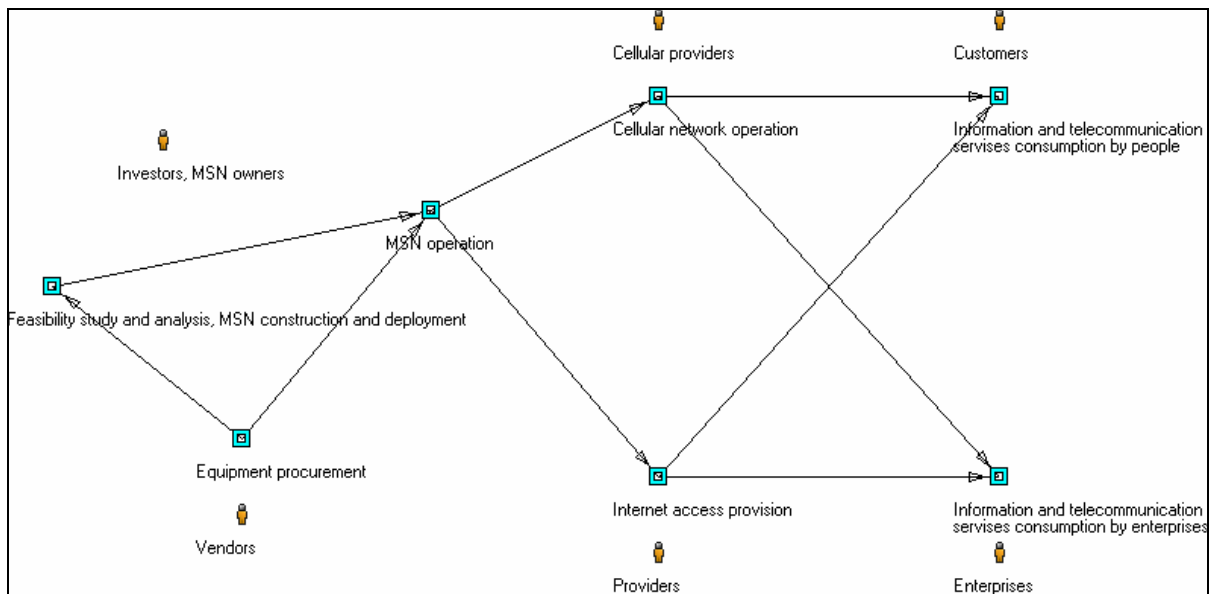


Fig. 14. Telecommunication technologies-based services market main processes and players

Currently leading Russian region cellular carriers engineers polling revealed, that carriers' development departments use their own experimental knowledge base when engineering data-communication networks, while data-communication implementation engineering solutions are foisted by hardware vendors. No operator either makes use of data-communication networks automated design aids, or models various designed/existing network behaviour situations when developing new regions, introducing new services or modifying data-communication network topology.

Development of automated design and modelling methods and aids requires large quantity of primary data for qualitative MSN technical and economical engineering, which includes: telecommunication hardware and technologies types and parameters; engineers, economists, project managers, marketers, and lawyers' level of knowledge.

Decision support systems fit most for MSN technical and economical engineering problem solution. Decision support systems can make use of simulation, expert and situational modelling (Aksyonov & Goncharova, 2006). Decision support systems development and deployment within cellular communication operators is a pressing and needed problem.

The following mathematical methods are used in MSN and business processes modelling, analysis and synthesis tasks: teletraffic theory may be used on all MSN levels except services level; simulation, situational and expert modelling methods are used for business processes analysis and synthesis tasks. Expert and situational modelling methods, neural networks, multi-agent and evolutionary modelling methods can be used in RCP formalization.

Multi-agent resource conversion processes theory is applied for MSN definition from decision support point of view.

Frame-concept and conceptual graphs based approach, offered by A. N. Shvetsov and implemented in form of «Frame systems constructor» expert system shell (FSC), is used as a means of knowledge formalization. A frame-based semantic network, representing feasible relations between frame-concepts, is defined in form of extended UML classes diagram, at the stage of system analysis.

UML sequence diagram is used for visual FSC output mechanism builder implementation (Fig. 15). This approach allows visual (in form of flowchart) problem solution flow

definition, when solution turns into a sequence of procedure (method/daemon) calls from one frame to another. Hereby, this approach allowed visual object-oriented ontology and knowledge-based output mechanism constructor implementation in form of decision search diagrams. Fig. 16 illustrates a fragment of decision search process in form of decision search diagram, with a fragment of MSN simulation model on background.

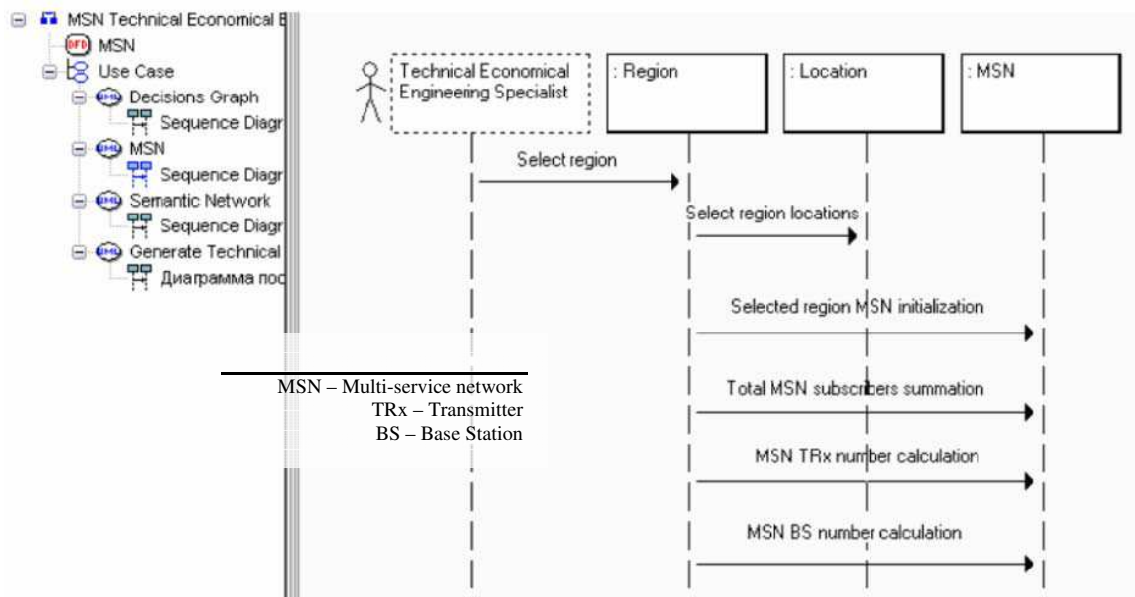


Fig. 15. Visual FSC output mechanism builder

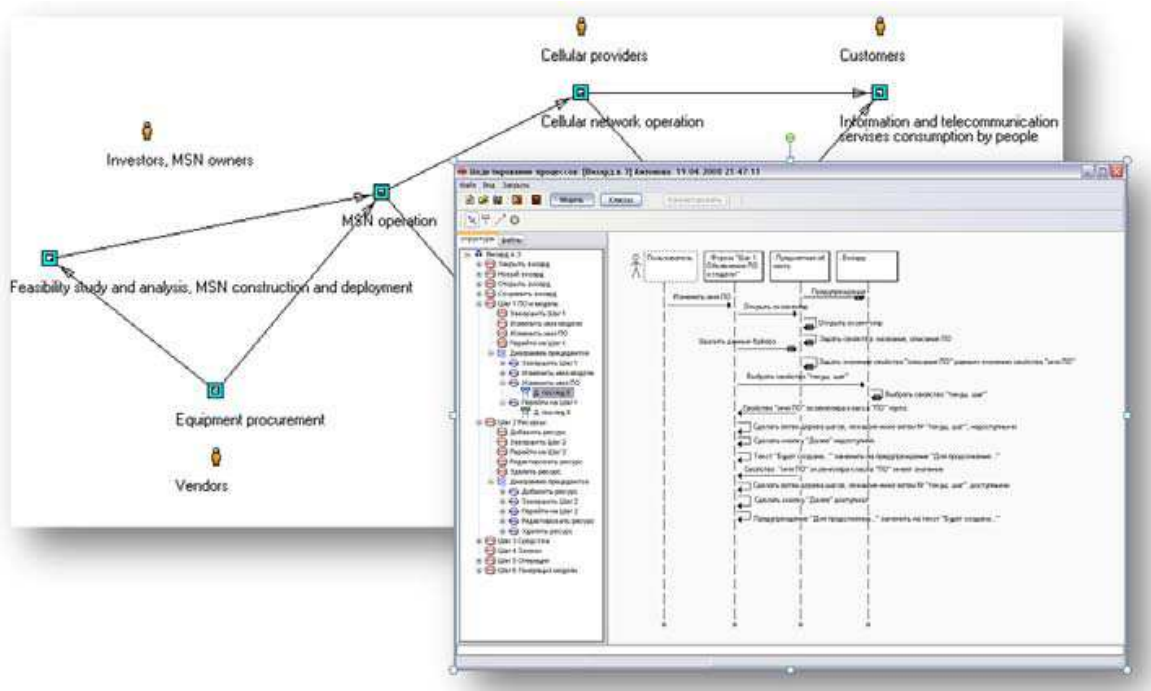


Fig. 16. MSN model and decision search process

BPsim.DSS was applied for MSN technical economical engineering in Ural region, covering metropolis Ekaterinburg, Russia, and satellites. Designed model is shown on Fig. 17.

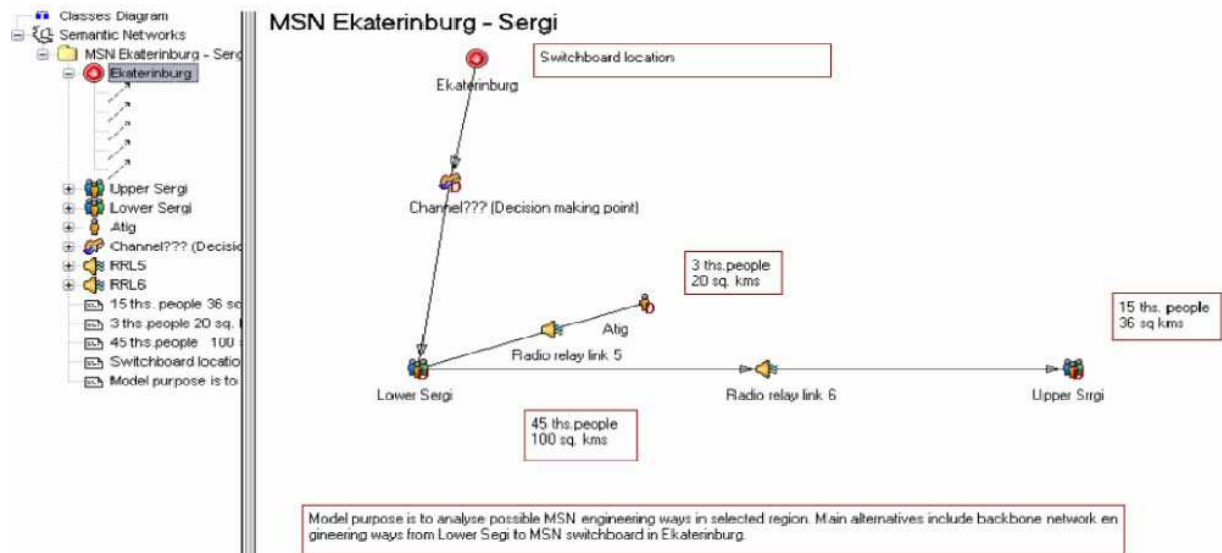


Fig. 17. MSN Ekaterinburg – Sergi model view

This constructor, provided that being filled with MSN subject area knowledge and technical and economical engineering rules, represents an intelligent MSN automated engineering system.

Graphical implementation of the model is presented on Fig. 18. Model allows switching on and off Base Stations (Access network elements) and Transport Networks, as well as changing elements parameters and allowing to select from options of renting or constructing a specific element.

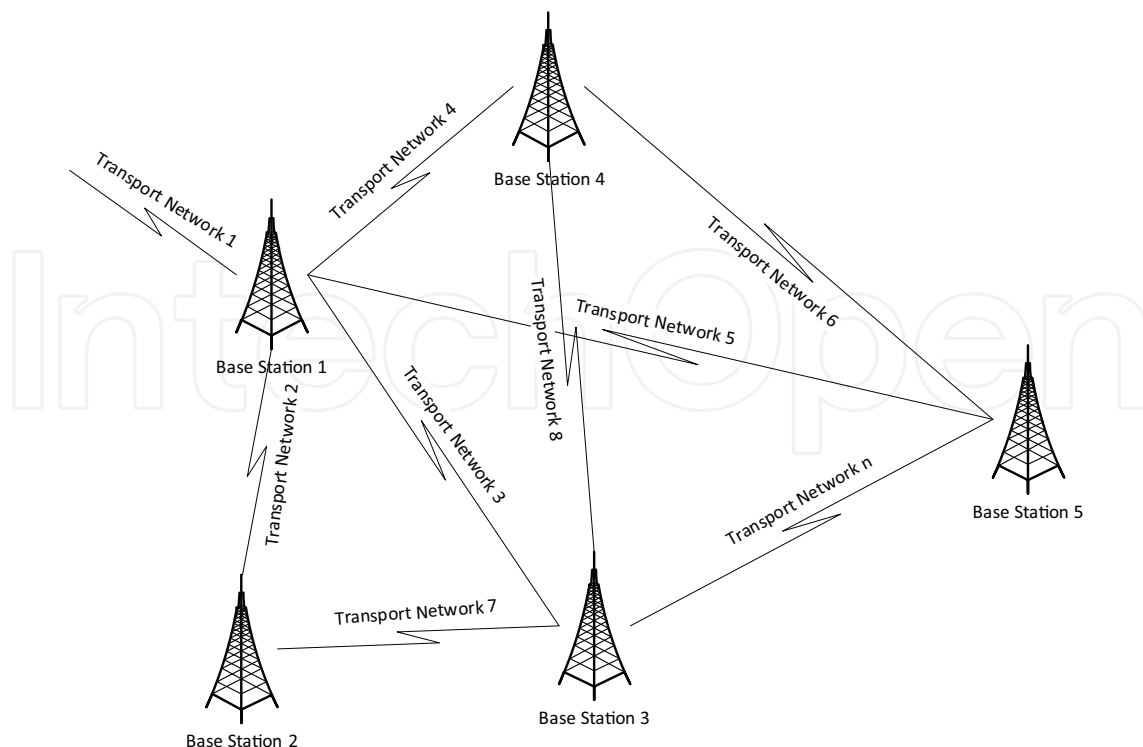


Fig. 18. Modelled MSN graphical model

The model is designed with a main purpose of MSN technical economical engineering with a centre in the metropolis and covering surrounding towns. Main goal is to estimate available MSN deployment options for provision of cellular and data transfer services. Synthesized model allows estimation of main investment indicators (IRR, EBI, Payback Period), that are required for substantiated decision making in MSN engineering. Live, one of the experiments that performed best, was implemented and performance indicators were measured after a certain while of performance. The real indicators were close to ones estimated with aid of decision support system BPsim.DSS. Transition from engineering to simulation modelling is implemented by semantic match making between FSC and multi-agent RCP elements. BPsim.DSS system from the object-structure analysis (OSA) (Gavrilova, 1989) point of view is presented in Table 4.

Stratum	Knowledge type	Stratum levels [BPsim.DSS functionality]
s_1	WHAT FOR-knowledge	Strategic analysis: system purpose and functions [Mission, vision, strategies, goals, indicators]
s_2	WHO-knowledge	Organizational analysis: system project team [Experts, analysts, decision-making people (agents)]
s_3	WHAT-knowledge	Conceptual analysis: main concepts, conceptual structure [FSC (hardware, technologies, MSN, services, processes, etc.)]
s_4	HOW-knowledge	Functional analysis: hypotheses and decision-making models [Agents' behavioural models (scenarios)]
s_5	WHERE-knowledge	Tridimensional analysis: environment, hardware, telecommunications [Geographic information system (regional geographic characteristics)]
s_6	WHEN-knowledge	Time analysis: time parameters and limitations [Simulation modelling (limitations in payback period, MSN deployment period, etc.)]
s_7	WHY-knowledge	Causal analysis: system explanation system engineering [Expert systems, knowledge and agent-rules bases, output mechanism]
s_8	HOW MUCH-knowledge	Economic analysis: resources, expenses, profits, payback [Solution (MSN technical and economical project)]

Table 4. Engineering and simulation system BPsim.DSS from OSA point of view

#### 11.4 BPsim.DSS application to subaru auto dealer logistical processes simulation

Finally, BPsim.DSS was applied to Subaru auto dealer sale process. Simulation result analysis helped this process be optimized, i.e. certain initial parameters being modified, resulting in effective logistics and warehouse processes. Initial data for simulation included sales statistics for each model, average retail pricing and dealer price markup, together with sales statistics depending on initial car location (at warehouse on location, at official Subaru representative's warehouse in Moscow, at Japanese warehouse ready for delivery),

including number of contracts and average delivery time from the order date. The main purpose was to estimate the necessary number of cars of each model at the warehouses on location and in Moscow, in order to achieve sales results of 20 to 40 cars per month.

Another model for Subaru auto included simulation of car repair process. The model considered main repair process stages, resulting in effective search of repair strategy.

Model was designed to examine, analyse and improve repair department activity of two dealers in Siberian region of Russia, and was based on the statistical data from the dealers. The model can be used by other enterprises, provided that it is adapted accordingly.

## 12. Conclusion

In this chapter we have presented the following keynote features.

Some popular dynamic situations modelling systems including AnyLogic, ARIS, G2, Arena were compared. This comparison revealed the necessity of a new system development, for it to be focused on multi-agent resource conversion processes. Among the disadvantages of the named systems we can name an incomplete set of features for dynamic situations modelling system; no support for subject area conceptual model engineering and multi-agent models, containing intelligent agents, design; incomplete multi-agent resource conversion processes problem orientation; programming user orientation; high retail price.

Multi-agent resource conversion process situational mathematical model requirements were designed. The model must provide the following functions: dynamic resource conversion processes modelling; definition of intelligent agent communities, controlling the resource conversion process; situational approach application.

System development required multi-agent resource conversion process model definition. The following features of the model were designed:

- Multi-agent resource conversion process main objects;
- Graphical notation;
- System graphs apparatus was applied to hierarchical process structure definition;
- Frame-semantic representation, based on frame-concepts and semantic graphs, was selected for knowledge representation model, which allowed subject area conceptual model definition;
- InteRRaP hybrid architecture was selected as a basis of multi-agent system;
- Multi-agent resource conversion process output mechanism, rule types, intelligent agent activity algorithm and situational simulation modelling algorithm were designed.

Based on the model and multi-agent resource conversion process system analysis, a software family of BPsim products was developed. It offers the full list of functional features, required from a problem-oriented dynamic simulation modelling systems and implements the following specific features:

- Subject area conceptual model definition;
- Multi-agent models definition, including both reactive and intelligent agents;
- Multi-agent resource conversion processes problem orientation;
- Balanced scorecard methodology integration;
- Significantly lower retail price.

Simulation, expert, situational and multi-agent modelling integration with object-oriented approach allowed implementation of new object-oriented multi-agent resource conversion processes simulation and decision support method, reflected in development of object-



oriented decision support system BPsim.DSS, deployed at companies in Ural region of Russia.

The mathematical model is based on discrete resource conversion process model. Within its framework the problem of transition between the knowledge representation, conceptual model and their technical implementation on relational database level, was solved. This approach allows Transact-SQL language to be used for subject area models design, data and knowledge input, logical output mechanism implementation.

### 13. Acknowledgment

This work was supported by the State Contract No. 02.740.11.0512.

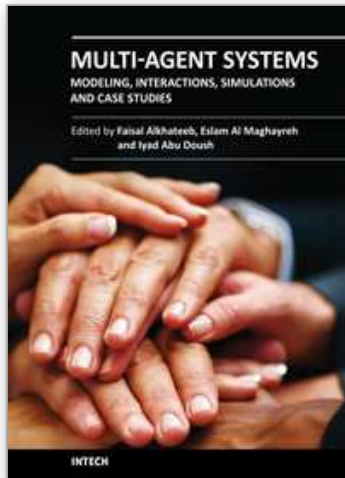
### 14. References

- Aksyonov K.A., Sholina I.I. & Sufrygina E.M. (2009a). Multi-agent resource conversion process object-oriented simulation and decision support system development and application, *Scientific and technical bulletin*, Vol. 3 (80), Informatics. Telecommunication. Control, St.Petersburg, pp.87-96.
- Aksyonov K.A., Smoliy E.F., Popov M.V. & Dorosinskiy L.G. (2009b). Development of decision support system «BPsim3»: Multi-service telecommunication networks design and modelling application, *Proceedings of 10th International PhD Workshop on Systems and Control*, September 22-26, 2009, Hluboka nad Vltavou, Czech Republic, pp. 112-117
- Aksyonov K.A., Bykov E. A., Smoliy E. F. & Khrenov A. A. (2008a). Industrial enterprises business processes simulateon with BPsim.MAS, *Proceedings of the 2008 Winter Simulation Conference*, Pages 1669-1677
- Aksyonov K.A., Spitsina I.A. & Goncharova N.V. (2008b). Enterprise information systems engineering method based on semantic models of multi-agent resource conversion processes and software, *Proceedings of 2008 IADIS International Conference on Intelligent Systems and Agents, part of Multi Conference on Computer Science and Information Systems (MCCSIS)*, July 22-27, 2008, Amsterdam, Netherlands, pp. 225-227.
- Aksyonov K.A. & Klebanov B.I. (2008). *Resource conversion processes simulation*, USTU, Ekaterinburg
- Aksyonov, K.A. & Goncharova N.V. (2006). *Multi-agent resource conversion processes dynamic simulation*, Ekaterinburg, USTU
- Andreichikov A.V. & O.N.Andreichikova (2004). *Intelligent information systems*, Moscow: Finance and statistics.
- Avramchuk E.F., Vavilov A.A. & S.V.Emelianov (1988). *Technology of system simulation*, Moscow: Machine construction industry; Berlin: Techniques
- Bertalanffy L. (1950). An Outline of General Systems Theory, *British Journal of the Philosophy of Science*, Vol. 1, pp. 134-164
- Bouling K.L. (1956). General Systems Theory, *The Skeleton of Science. Management Science*, № 2, pp. 197-208
- Buch G., Rambo D. & A. Jacobson (1993). *UML Language. User Manual*, Moscow: DMK
- Chastikov A. P., Gavrilova T. A. & Belov D. L. (2003). *Expert systems development. CLIPS environment*, St. Petersburg



- Courtois P. (1985). On Time and Space Decomposition of Complex Structures, *Communications of the ACM*, Vol. 28, № 6., pp. 596-610
- Gavrilova T.A. (1989). *Training expert system developers team*, Kishinev, pp. 59-62
- Gig J. van (1981). *Applied General Systems Theory*, Mir, Moscow
- Jackson P. (1998). *Introduction to expert systems. 3rd Edition*. Addison Wesley
- Mesarovich M. & Takahara Ya. (1978). *Common system theory: mathematical basics*, Moscow, Mir
- Minsky M. (1975). *A framework for representing knowledge in the psychology of computer vision*, P.H.Winston (ed.), McGraw-Hill
- Muller J.P. & M.Pischel (1993). *The Agent Architecture InteRRaP: Concept and Application*, German Research Centre for Artificial Intelligence (DFKI)
- Shvetsov, A.N. (2004). *Corporate intellectual decision support systems design models and methods*, DPhil research paper, St.Petersburg, Russia
- Sovetov B.Y. & Yakovlev S.A. (2001). *Systems modelling*, Moscow, High School
- Sowa J.F. (2000). *Knowledge representation: Logical, Philosophical, and Computational Foundations*, Pacific Grove, CA: Brooks/Cole Publishing Co.
- Wooldridge M. (1995). Intelligent Agent: Theory and Practice, *Knowledge Engineering Review*, Vol. 10 (2).

IntechOpen



## **Multi-Agent Systems - Modeling, Interactions, Simulations and Case Studies**

Edited by Dr. Faisal Alkhateeb

ISBN 978-953-307-176-3

Hard cover, 502 pages

**Publisher** InTech

**Published online** 01, April, 2011

**Published in print edition** April, 2011

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Agent systems are open and extensible systems that allow for the deployment of autonomous and proactive software components. Multi-agent systems have been brought up and used in several application domains.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Konstantin Aksyonov, Eugene Bykov, Leonid Dorosinskiy, Elena Smolij and Olga Aksyonova (2011). Decision Support based on Multi-Agent Simulation Algorithms with Resource Conversion Processes Apparatus Application, Multi-Agent Systems - Modeling, Interactions, Simulations and Case Studies, Dr. Faisal Alkhateeb (Ed.), ISBN: 978-953-307-176-3, InTech, Available from: <http://www.intechopen.com/books/multi-agent-systems-modeling-interactions-simulations-and-case-studies/decision-support-based-on-multi-agent-simulation-algorithms-with-resource-conversion-processes-appar>

# **INTECH**

open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen