

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

6,900

Open access books available

185,000

International authors and editors

200M

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

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Decision Support Systems Application to Business Processes at Enterprises in Russia

Konstantin Aksyonov, Eugene Bykov,
Leonid Dorosinskiy, Elena Smoliy, Olga Aksyonova,
Anna Antonova and Irina Spitsina
*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.

System analysis and synthesis constitute a management aid, targeting more effective system organization considering its limitations and objectives. In the limits of decision support process, analysis and synthesis methods are used for forecast and estimation of the consequences of taken decision. In such case available situation development scenarios are designed and analysed, scenario synthesis is performed, results are being evaluated in order to develop the policy for the most desired results. Computer aid for analysis and synthesis problems provides the most perspective and rational system development strategy making. Formalized analysis and synthesis problems solving might be interesting from the point of view of workflow automation for analysts and decision making people.

The following analysis and synthesis problems of business systems are discussed further: virtual enterprise establishment; business process re-engineering; business process benchmarking (improvement and enhancement). In Russia major research relates to business systems analysis. This concept includes business processes, management, production and manufacture, logistics, technological processes and decision making. The task of virtual enterprise establishment (Andreichikov & Andreichikova, 2004) relates to a class of complex systems structural synthesis tasks. The main objective of virtual enterprise establishment is cooperation of legally independent companies and individuals, manufacturing certain product or providing services in common business process. Main goal of business processes re-engineering is reorganization of material, financial and informational flows, targeting organizational structure simplification; resource use redistribution and minimization; client needs satisfaction time decrease; quality of client service improvement.

Business process re-engineering offers solution for the following tasks: existing business process and enterprise activity analysis, decomposition to sub-processes; analysis of bottlenecks in business processes structure, e.g. related to resources underload or overload,

queuing; business process re-organization (synthesis) for elimination of problems and reach of set effectiveness criteria; design and development of information system for business process support. Problem of business process structure bottlenecks analysis can be solved with aid of static or dynamic modeling methods. Static modeling implies definition and analysis of business process structure, as well as cost analysis of process functions, identification of the most demanding and unprofitable functions, or those ones with low coefficient of resource use. Dynamic simulation modeling allows implementation of multiple business process operations in continuous period of time, offering statistics gathering of process operation and identification of bottlenecks in their structure. Business process re-organization relying on static modeling is based on heuristic approach and demands high qualification and experience from analyst. Simulation modeling in business processes re-engineering allows automation of re-engineering rules and their application. Business process benchmarking closely relates to re-engineering. Main goal of benchmarking is business process re-organization in accordance with master model. Master model is based on a combination of best business processes of various enterprises, identified with comparative analysis.

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

Practical application of business process re-engineering approaches as management tools has its limitations: need for a highly qualified analyst, his deep understanding of problem domain; unavailability of synthesis of business process re-engineering decisions and check of computed solutions on real management object. Simulation and AI methods applied to automation of business process re-engineering allow to: 1. Reduce demand for analyst experience in system analysis and synthesis by using formalized expert knowledge and applying mathematical modelling methods; 2. Estimate computed re-engineering solutions and select the most effective solution on management object model.

Use of multi-agent approach on stage of business process model formalization is caused by the presence of decision making people in the system; their behaviour is motivated, they cooperate with each other, accumulate knowledge of problem domain and management tasks solution scenarios. Intelligent agent model provides design of decision making person model, which is the basis of information technology implementation for model analysis and decision making.

2. Analysis of business processes multi-agent dynamic models

Consider the problem of business process formalization model selection. The following models support agent representation of business processes: Gaia model, Bugaichenko's model, Masloboev's model, simulation model of intelligent agents interaction (SMIAI), Resource-Activity-Operation model (RAO), multi-agent resource conversion processes model (MRCP).

Gaia model of multi-agent system was offered by Michael Wooldridge and Nicholas Jennings (Wooldridge et al., 2000). It defines a system as an artificial structure that consists of heterogeneous agents, interacting with each other in order to achieve common goal that in turn consists of agents' sub-goals. Note that authors do not consider agents behaviour in situations with competitive goals.

The most abstract entity in concepts hierarchy of Gaia model is a system. Next hierarchy level contains roles (Fig. 1). Model design begins with analysis stage.

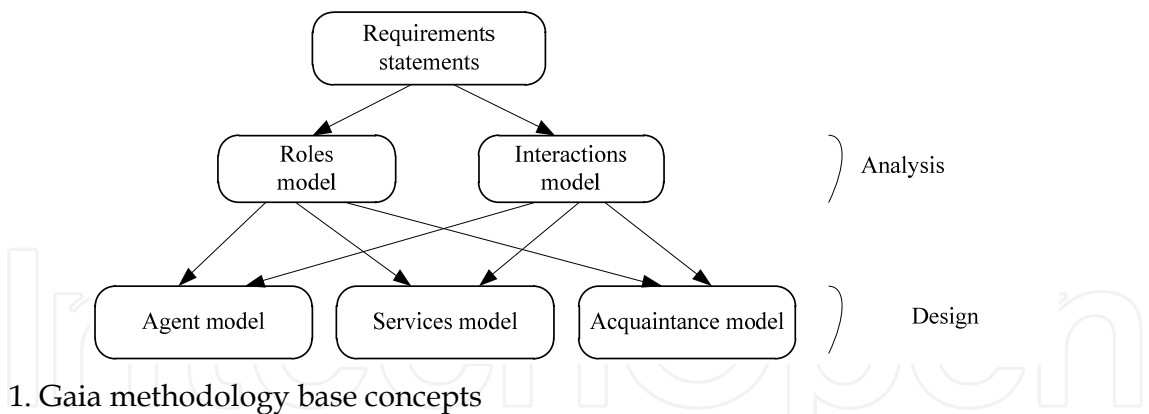


Fig. 1. Gaia methodology base concepts

A role has 4 attributes (Fig. 2) (Wooldridge et. Al, 2000): *Responsibility* determines functionality of role and can be divided into 2 kinds – *Liveliness properties* contain system reaction to external exposure (reactive component) and can be defined with temporal logic that has activity and role protocols as arguments; *Safety properties* define a set of activities that should not occur during role functioning. *Permissions* include rights associated with a role, for resources that are available. Resources in Gaia model may include information objects that may be used in ways of generation, reading and modification. *Activities* are computations to be carried out by the agents. These computations are carried out without interactions with other roles. Ways of interaction with other roles are defined by *Protocols*.

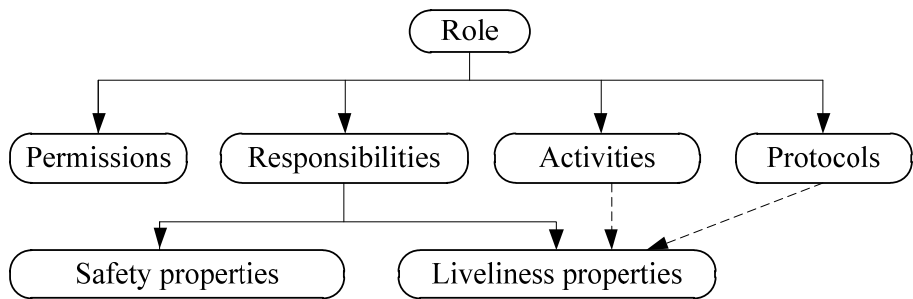


Fig. 2. Role attributes in Gaia model

Interactions model includes definition of the attributes of interaction protocols: interaction goal, interaction initiator role, interaction respondent role, interaction input and output resources and text definition of interaction process. Authors imply limitation of system structure being static in frames of roles interaction model definition.

Design phase starts with *agent model* definition. This includes system agents types definition and agent types correlation with defined roles. During simulation a defined number of agent instances, in accordance with instance generation rules, are dynamically created for each type.

Services model is provided for agent functionality definition, when each agent service is mapped to a role defined on role activity analysis phase. Service has information inputs and outputs (corresponding to attributes of roles interaction protocol), as well as start and finish conditions in form of role safety properties. Here service represents a resource convertor, active when input resources are present and start condition is fulfilled; it generates output resources when finish condition is fulfilled.

Acquaintance model represents a directed graph with agent types as graph nodes and communications between agent types as arcs.

Bugaichenko's model (Bugaichenko, 2007) defines intelligent agents with mental (Belief-Desire-Intention, BDI) architecture. Multi-agent system properties are defined with aid of developed formal logic – MASL – multi-agent systems logical specification method with time limitations; these systems are considered to be capable of experience accumulation and analysis.

Multi-agent system model is represented with a triplet:

$$MAS=(S, AG, env), \text{ where}$$

S – finite collection of external environment states;

$AG=\{ag_1, \dots, ag_n\}$ – finite collection of agents;

env – function, defining a possible reaction of external environment to activity of all system agents.

Each agent is represented with a collection:

$$ag=(S, A, env, see, Ib, bel, brf, ID, des, drf, filter, plan, prf), \text{ where}$$

A – nonempty finite collection of agent activities;

env – external environment behavior function, correlating a collection of next possible states of external environment to the current state of external environment and selected action of an agent;

see – correct perception of external environment states by an agent, setting collection P of equivalence classes on S ;

Ib – collection of agent beliefs, dependent on agent perception of external environment and his own activity;

bel – collection of current agent beliefs;

brf – beliefs update function;

ID – collection of agent desires; depends on goals (criteria functions);

des – collection of current agent desires;

drf – agent desires update function;

$filter$ – agent desires filtration function;

$plan$ – current agent plan, represented by finite state machine with input alphabet P , output alphabet A , states set I_{pln} , transition relation opl_n and initial state of $ipl_n, 0$;

prf – plan update function.

Bugaichenko's agent has mental (BDI) architecture, featuring three components: beliefs, desires and intentions (Bugaichenko, 2005).

Beliefs contain information about regularities and current state of external environment of an agent. This information may be erroneous or incomplete, thus it may be considered as belief, not as reliable knowledge. Note that prognosis function is also considered an agent belief.

Agent desires make up a collection of all agent goals. It is unlikely that an agent, limited with resources, will be able to achieve all his goals.

Intentions make up a collection of goals that an agent decided to achieve. Satisfiability of intentions (in case of planning agent) may be defined as possession of a plan, leading to goal achievement.

One of the most complex stages of decision support process is selection (planning) of activities for goals achievement. Bugaichenko offers agent plans representation in form of a network of interacting finite state machines (Bugaichenko, 2007).

Problems of accumulating experience by the system and agent self-prognosis of own activity within Bugaichenko model are also solved with use of symbolic data representation in form of resolving diagrams.

Multi-agent system properties are described with aid of developed formal logic MASL – method for logical specification of multi-agent systems with time limitations; these systems are considered to be capable of experience accumulation and analysis. MASL logic is the interconnection of capabilities of the following temporal logics: Propositional Dynamic Logic PDL, Real-Time Computation Tree Logic RTCTL and Alternating-Time Temporal Logic ATL (Bugaichenko, 2006). Adaptation of these logics for specification of properties of multi-agent system mathematical model allowed Bugaichenko formalize definition of cooperation and agent competitiveness, define nondeterministic behaviour of external environment, extend expressive power of specification language.

Key concept of agent interaction modeling is *coalition*. According to Bugaichenko, coalition C is a certain subset of system agents that act together in order to achieve personal and common goals. Agents of the coalition trust each other, exchange information and coordinate activities. At the same time they do not trust the agents for outside the coalition, and do not affect their behavior. So, formally, coalition may be considered a single intelligent agent, that may have sense of external environment and mental behavior defined as a combination of senses and mental states of coalition agents.

Masloboev's multi-agent model (Masloboev, 2008) was developed for information support of innovative activity in the region together with support and estimation of potentially effective innovative structures.

Innovative activity subjects are represented in form of software agents, operating and interacting with each other in common information environment (virtual business environment, VBE) for the benefit of their owners, forming an open multi-agent system with decentralized architecture.

According to Masloboev (Masloboev, 2008), VBE model has the form of

$$E_{VBE} = \{S, P, I, A, R, Atr\}, \text{ where}$$

S – collection of business process subjects;

P – collection of business processes;

I – set of relations on object models (business process subjects);

A – collection of agents;

$R = \{BI, BPL\}$ – collection of innovation resources, including business ideas (BI) and business plans (BPL);

Atr – collection of model objects attributes.

Agent model has the form of:

$$A = \{S, BI, ORG_A, C_A\}, \text{ where}$$

ORG_A – agent organizational structure,

C_A – agent inner structure.

Organizational structure is defined in form:

$$ORG_A = \{G, RL, CP, ACT, STR, L, ST, SL, T\}, \text{ where}$$

G – agent goals tree;

RL – collection of agent roles that it needs to operate in order to achieve the goals;

CP – collection of agent skills and capabilities that it needs to possess in order to operate the roles;

ACT – collection of actions;

STR – collection of agent behavior strategies for goal achievement;

L – collection of languages, including agents interaction language, local planning language, and execution level language, defined by used network services;

ST – set of agent states;

SL – collection of agent operation rules;

T – general transfer function on multitudes ST , SL and ACT (Masloboev, 2009).

Agent inner structure defines its functional design:

$$C_A = \{K_A, M_A, P_A, R_A, I_A, C_A\}, \text{ where}$$

K_A – mental subsystem;

M_A – modeling subsystem;

P_A – analysis and planning subsystem;

R_A – reactive subsystem;

I_A – coordination and interaction subsystem;

C_A – communication subsystem (Masloboev, 2009).

Masloboev's hybrid agent architecture is an extension of existing InteRRaP architecture with problem-oriented subsystem of continuous simulation modeling (system dynamics models complex), used by the agents for simulation of single innovation projects development scenarios, and behavior of companions and competitors on innovative services market. Thus, use of simulation apparatus lets the agent forecast the results of its activity (risks and economical effect of investments into innovations). Hybrid architecture consists of these elements:

- *Knowledge control subsystem*, based on descriptive logic, provides temporal reasoning and some other processing mechanisms. Three levels of knowledge play key role in agent architecture: problem domain knowledge, interaction knowledge (general declarative behavior rules together with rules for problem domain knowledge replenishment and update), and control knowledge (knowledge, applying interaction knowledge to problem domain knowledge for replenishing and update of operational memory).
- *Operational memory* is used as a temporary data storage for control subsystem knowledge, user or communication control data. Operates in form global message board.
- *Decision support system* is implemented on the basis of OLAP technology.
- *Communications control module* compiles and sends messages to other agents, also receives message delivery confirmations. ACL (agent communication language), based on theory of speech activity, is used for agent interaction.

Masloboev's model is designed for creation of virtual enterprise for innovation project implementation by selecting the most effective one (from the point of view of innovation project investments) and its correlation with the most advantageous business structures capable of project implementation. Innovation project is selected on the basis of effectiveness criteria mathematical models (Masloboev, 2008), including such project indicators as economical effectiveness, estimated project implementation time, partners reliability (competence).

Business structures within the model are implemented as agent coalitions with common goals of corresponding business ideas implementation. Business ideas are used for automatic generation of agent coalition on the basis of semantic matchmaking of corresponding business idea parameters.

Model of intelligent agent with simulation apparatus is implemented in multi-agent system of innovation activity information support, providing joint use of innovation portals information databases during automated generation of potentially effective business structures, targeting innovation projects implementation.

Simulation model of intelligent agents interaction (SMIAI) was developed by Rybina and Paronjanov with the purpose of formalizing communicative activity of intelligent agents by modelling separate communication components: interaction participants, communicative environment, problem region, interaction language, dialog scenarios (Rybina & Paronjanov, 2008a).

Model of intelligent agents interaction is defined in form (Rybina & Paronjanov, 2008a):

$$MI = \{SA, SE, SP, DI, L, SRA\}, \text{ where}$$

SA – collection of agents A_i ;

SE – collection of communicative environments E_k ;

SP – collection of problem regions, set by problem domains PR (collections of classes, their instances and relations between them), collection of current tasks ST (represented with five typical tasks: diagnostic, engineering, planning, control and study) and relations of compliance of elements PR with current tasks from collection ST ;

DI – collection of dialog scenarios between interaction participants within multi-agent system;

L – interaction language, represented with lexical, syntactical and semantical components;

SRA – collection of relations, indicating possibility of interaction between a pair of interaction participants A_j and A_i in environment E_k .

Intelligent agent model was received by adaptation of Cetnorovich's model K with a purpose of having intelligent agents as participants of interaction (Rybina & Paronjanov, 2008a):

$$A = \{M, Q, R, P, Eff, MI, Plan, Aim, Time\}, \text{ where}$$

M – collection of environment models, available to the agent;

Q – collection of agent goals;

R – collection of actions, acceptable for the agent;

P – collection of tasks decomposition options (partial plans library);

Eff – collection of acceptable impacts on the agent from environment;

MI – agents communication model;

Plan – function of agent activity plan generation based on current goal and environment model;

Aim – agent goal-setting mechanism based on external impacts;

Time – collection of activities reflection into their duration based on environment state.

Interacting agents must have knowledge of communication language and problem domain that have to be the same for each agent or at least have common areas (Rybina & Paronjanov, 2008b).

Interaction of intelligent agents in SMIAI model is performed in form of a dialog (polidialog), including global structure, dependent only on goals of interaction participants, thematical structure, dependent only on the the current problem, and local structure, used for representation of communication actions sequence on local level (Rybina & Paronjanov, 2008b). Authors offer use of painted Petri nets for local interaction steps definition (i.e. actions and reactions of partner agents on each step of interaction).

Model of dialog local structure has finite set of dialog states (P_i positions) and finite set of transfers t_j , each of which has a corresponding expression (transfer condition) in interaction language L . Model makes use of KIF and KQML languages for language interaction. The main concept here is speech acts theory, which is the basis of KQML language, as well as large amount of problem domains, defined on the basis of KIF language.

Graphical notation of local structure model for agents dialog is presented on Fig. 3.

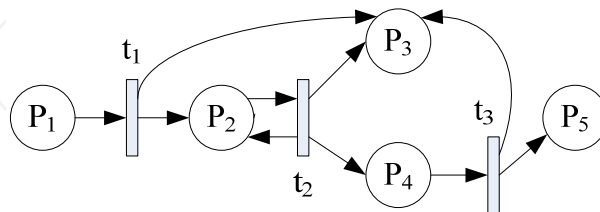


Fig. 3. Graphical notation of local structure model for agents dialog

Each dialog position is defined in form:

$$P = \left\{ agent_state \begin{bmatrix} execut_action \\ expec_reactions \\ end_of_dialog \end{bmatrix}, color_position, actions_collection \right\}$$

Each dialog transfer has a structure like:

$$t = \{ P_{i-1}, P_i, transfer_condition(painting_P_{i-1}), actions_with_colors \}, \text{ where}$$

$$actions_with_colors = \begin{bmatrix} Delk = \{ position_with_deleted_color; deleted_color \}, \\ Addk = \{ position_with_added_color; added_color \}. \end{bmatrix}$$

Sequece of agents communicative actions is defined on the Petri net by output track analysis, determination of passed positions and inclusion of those actions into final sequence that are defined in actions list for current position.

General model of intelligent agents interaction is implemented in SMIAI, developed with aid of Gensym Corp. G2 and Microsoft Visual Studio. System has been tested in problem domains of online billing, investment projects management, control of state for chemically-dangerous objects anf other (Rybina & Paronjanov, 2009).

Resource-Activity-Operation model (RAO) is used for definition of complex discrete systems (CDS) and activities within these systems in order to study static and dynamic features of events and activities (Emelyanov & Yasinovskiy, 1998). System distrecity is defined with two properties:

- CDS content may be defined with countable set of resources, each of which relates to certain type;
- CDS state changes occur in countable moments of time (events) and $C_{e_i}^+ = C_{e_{i+1}}^-$ (system state after e_i event is identical to system state before e_{i+1} event).

Conceptually CDS may be represented with a set of *resources*, that have certain parameters. Parameter structure is inherited from resource type. Resource state is defined with a vector of all its parameter values. System state is defined with all its resource parameter values.

There are two types of resources – permanent (always exist during model simulation) and temporary (dynamically created and destroyed during model simulation). By analogy with GPSS language, RAO resources may be called transacts, that are dynamically generated in certain model object, pass the other objects and are finally destroyed. Note that resource database in RAO is implemented on the basis of queueing system apparatus, having automatic support of system operation statistic gathering: average value of examined indicator, minimum and maximum values, standard deviation.

Resources perform specific *activities* by interacting with each other. Activities are defined with events of activity start and activity end, the event in general is considered a signal, transferring data (control) about certain CDS state for certain activity. All events are divided into regular and irregular. Regular events reflect logic of resource interaction between each other (interaction sequence). Irregular events define changes of the system, unpredictable in production model. Thus, CDS operation may be considered as timely sequence of activities and interrupting irregular events.

Activities are defined with *operations*, that are basically modified production rules that consider timely relations:

IF (condition) THEN1 (event1) WAIT (timely interval) THEN2 (event2).

Operation defines pre-conditions (operation resources state should fulfill those) and state change rules for resources in the beginning and end of corresponding activity. Operation with temporal interval of 0 in RAO model is considered a *decision point*, and is a usual production rule. RAO decision points are equivalent to reactive agents that store environment response data in knowledge base.

Intelligent modeling system RAO-simulator has been developed, its structure is shown on Fig. 4.

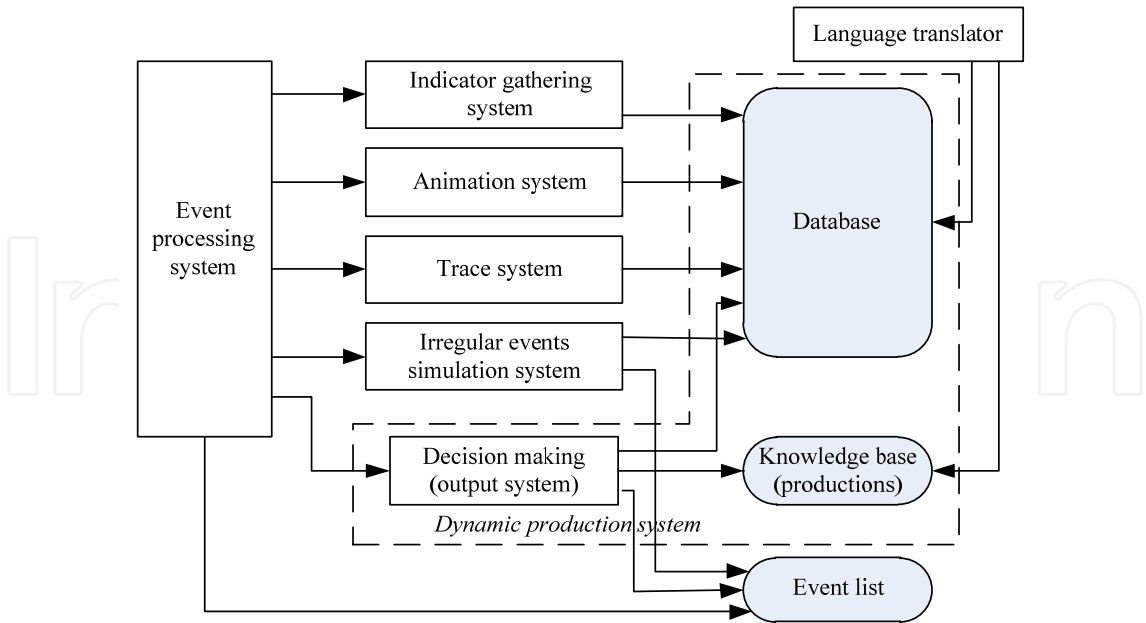


Fig. 4. Structure of RAO-simulator

Main elements of RAO-simulator include dynamic production system and events apparatus. During simulation system state changes in accordance with irregular event definition or activity that started or ended. Output system is called after each state change. It scans

knowledge base for all operations and checks pre-conditions for availability of operation start. If located, the start events are raised for corresponding actions. Trace system displays detailed information on events to dedicated file, which is further processed for process detailed analysis and information representation in convenient form.

Dynamic model of multi-agent resource conversion processes (MRCP) (Aksyonov & Goncharova, 2006) was developed on the basis of resource conversion process (RCP) model (Aksyonov, 2003) and targets modeling of business processes and decision support for management and control processes.

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

Key concept of the RCP model is a resource convertor that has input, launch condition, conversion, control, output. Launch condition defines a moment in time when a convertor starts activity on the basis of such factors as state of conversion process, state of input/output resources, control commands, tools for conversion process and other events in external environment. Conversion duration is defined immediately before conversion based on control command parameters and active resource limitations.

MRCP model may be considered an extension to base RCP model, adding functionality of intelligent agents.

The main objects of discrete Multi-agent RCP are: operations (*Op*), resources (*Res*), control commands (*U*), conversion devices (*Mech*), processes (*PR*), sources (*Sender*) and resource receivers (*Receiver*), junctions (*Junction*), parameters (*P*), agents (*Agent*) and coalitions (*C*). Process parameters are set by the object characteristics function. Relations between resources and conversion device are set by link object (*Relation*). The agents and coalitions existence resumes availability of the situations (*Situation*) and decisions (action plan) (*Decision*).

MRCP model has hierarchical structure, defined with system graphs of high-level integration.

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 behavior and capable of interaction with other objects in dynamic virtual environment. In every point of system time a modeled agent performs the following operations (Aksyonov & Goncharova, 2006): 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.

Coalition is generated consequently after several agents union. Agent coalition has the following structure:

$$C = \langle \text{Name}, \{A_1, \dots, A_m\}, G_C, KBC, M_{In}, M_{Out}, SPC, \text{Control_O} \rangle, \text{ where}$$

Name – coalition name;

$\{A_1, \dots, A_m\}$ – a collection of agents, forming a coalition;

G_C – coalition goal;

KB_C – coalition knowledge base;

M_{In} – a collection of incoming messages;

M_{Out} – a collection of outgoing messages;

SPC – a collection of behaviour scenarios acceptable within coalition;

Control_O – a collection of controlled objects of resource conversion process.

Fig. 5 shows an example of *C1* coalition formation after union of *A2* and *A3* agents. Here *C1* coalition controls agents *A2* and *A3*, but *A1* agent acts independently.

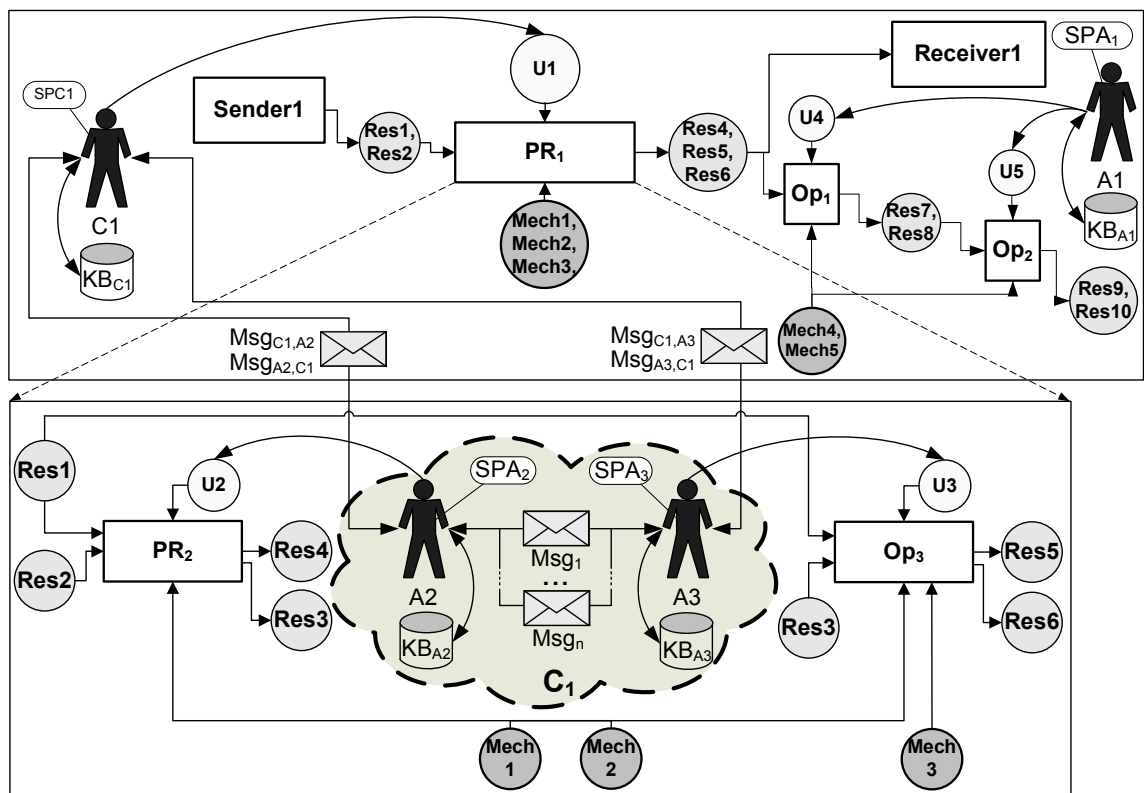


Fig. 5. Coalition formation

Simulation engine algorithm of agent-containing model consists of the following main stages: current point of system time identification $SysTime = \min T_j$; agent and coalition $j \in RULE$

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).

Each agent possesses its knowledge base, set of goals that are needed for behavior 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 (Fig. 6):

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

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 behavioral 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 problem domain.

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 behavior components are implemented in form of agent productional rules base and inference machine (simulation algorithm).
3. **Reactive behavior** 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.

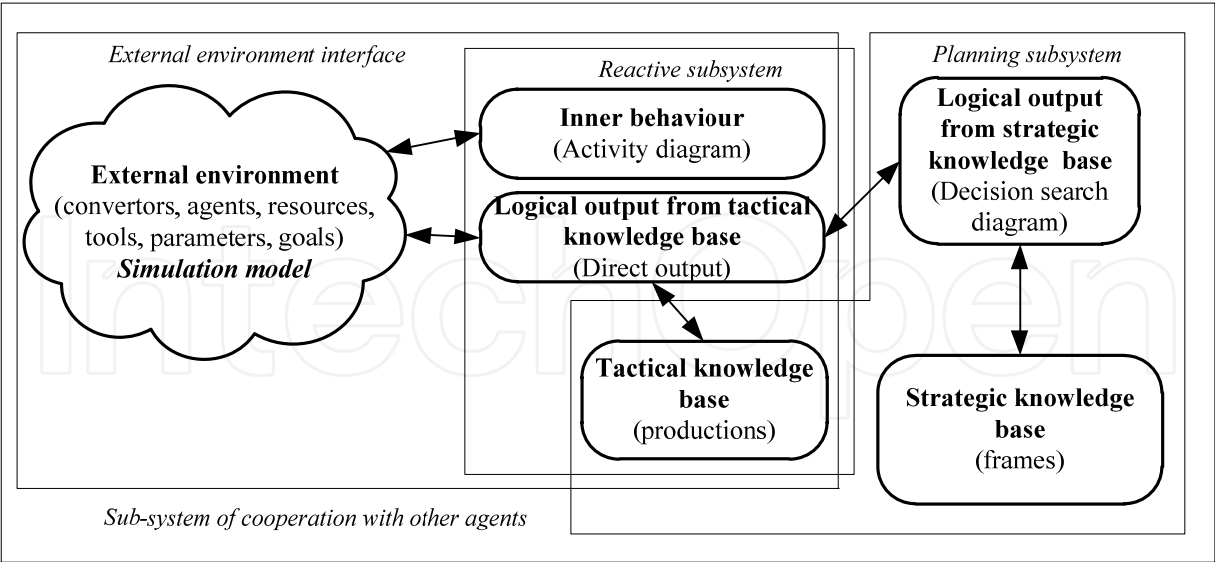


Fig. 6. MRCP agent hybrid architecture

Problem domain 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 (Fig. 7), 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.

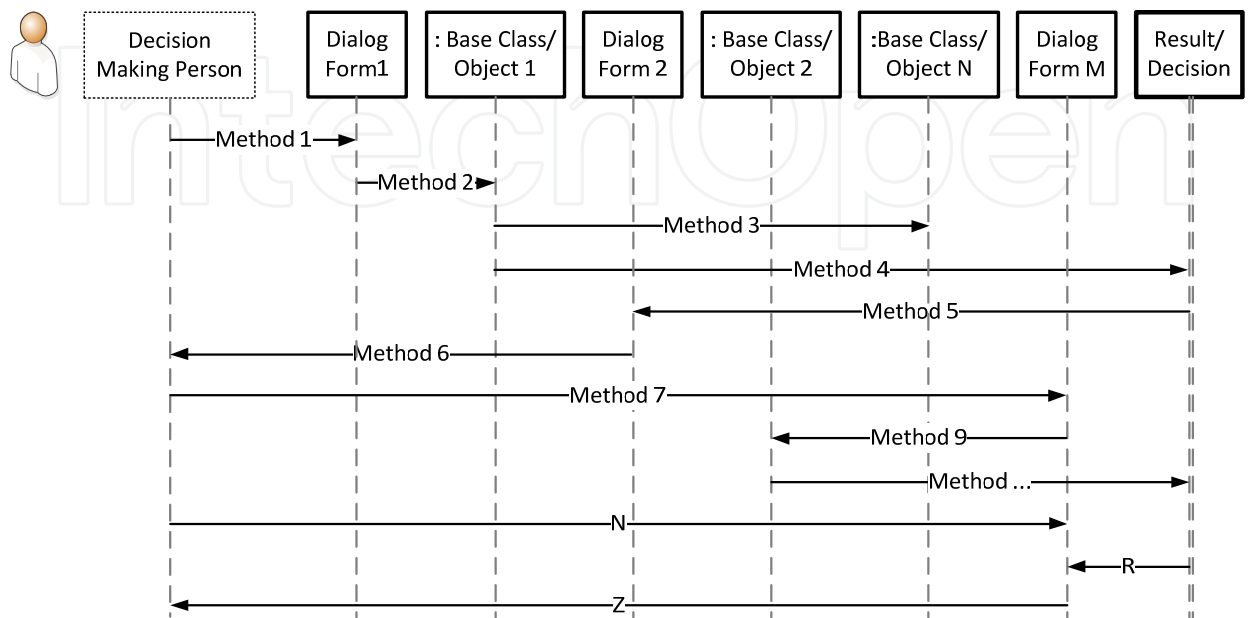


Fig. 7. General decision search diagram in decision support system BPsim.DSS

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 behavior 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. Comparison of described models in presented in Table 1. The following comparison criteria has been selected: resource convertor model (discrete business process model); queue system model – tool for analysis of discrete stochastic systems and business process; reactive agent and intelligent agent models (tool for decision making people definition). Full potential of intelligent agents is implemented in Bugaichenko, Maslovboev and MRCP models. At the same time Buhgaichenko’s agent is not equivalent to decision making person, but is a software entity that works independent from an expert (analyst). Masloboev uses a complex of continuous system dynamics models for agent actions forecast and has no support for discrete-event processes. SMIAI model is oriented on agent communication methods research and does not integrate with simulation modeling system. Gaia model has support for information resources convertor and intelligent agent, but has less functional capabilities, than RAO and MRCP. RAO model is similar to MRCP in areas of decision support and has powerful tools for business processes formalization. A serious disadvantage is missing support of intelligent agents. MRCP model includes a hybrid agent model (intelligent and reactive), model of resource convertor and queue system, allowing the analyst to analyze dynamic features of business processes. Hybrid agent model allows analysis of various decision making people behavior scenarios.

Comparison criteria	Bugaichen-ko's model	Masloboev's model	Rybina's model	Gaia model	RAO model	MRCP model
Resource convertor model						
input/output/start conditions/duration	○	○	○	● / ● / ● / ○	● / ● / ● / ●	● / ● / ● / ●
hierarchical model of convertor				○	○	●
operations timely interrupts				○	○	●
Queue system	○	○	○	○	●	●
Reactive agent model (knowledge representation form)	MASL temporal logic	Productions	Painted Petri nets	○	Productions	Productions
Intelligent agent model						
agent goals	●	●	●	●	○	●
actions forecast	●	●	○	○		●
action analysis and planning	●	●	●	●		●
agents self-study	●	●	●	○		●
planning, forecast, self-study implementation technology	Resolving diagrams -based expert system	System dynamics simulation	Expert system	Expert system		Frame-based expert system
message exchange language	TPA / Signals	TPA	TPA	Signals		Signals
agents cooperation model	●	●	●	●		●
Software implementation of the model	Beta	Proto-type	Proto-type	○	Beta	RTM

Table 1. Comparison of dynamic situation modeling models

Thus, MRCP model and BPsim products have the fullest functionality in area of business processes formalization.

3. Current state of dynamic situations modeling systems

Dynamic situations modeling systems area state analysis reveals unavailability of resource conversion processes oriented systems. Nearest functionality analogs (Table 2) include simulation and expert modeling tools, particularly real-time expert system G2 (G), multi-agent simulation system AnyLogic (L), business-processes modeling system ARIS (T), simulation system Arena (A) , simulation model of intelligent agents interaction SMIAI (S), Masloboev’s innovations support system (M), Resource-Activity-Operation model (R). The last three are unavailable for purchase, that is the reason for zero in retail price line.

Comparison criteria	T	G	L	A	S	M	R
Problem domain 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 modeling availability							
• “Agent” element	○	○	●	○	●	●	○
• Agents behavior models	○	○	●	○	●	●	○
• Agent’s knowledge base support	○	○	○	○	●	●	○
• Message exchange language	○	○	○	○	●	○	○
Discrete event Simulation modelling	●	●	●	●	●	○	●
Expert modeling	○	●	○	○	●	○	○
Situational modeling	○	●	○	○	●	○	○
Object-oriented approach							
• Use of UML language	●	○	○	○	○	○	○
• Object-oriented programming	○	●	●	○	●	○	●
• Object-oriented simulation	○	●	●	○	●	○	●
• Problem domain conceptual model and object-oriented simulation integration	○	○	○	○	○	○	●
Retail price, ths \$	50	70	8	4	0	0	0

Table 2. Modeling 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, except RAO system that makes use of internal programming language. 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, G2, SMIAI and RAO 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 modeling and expert systems are used in modeling, 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 modeling and multi-agent systems.

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.

4. 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 modeling 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 problem domain business process modeling and software design (www.bpsim.ru), offering a comprehensive solution for business modeling and techno-economic engineering problems, which in turn considerably simplifies and speeds analysts’ work:

- *BPsim.MAS* – multi-agent dynamic situations modeling 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 classes 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. Problem domain 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, problem domain 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 3.

BPsim.DSS agent model is represented with four levels in compliance with *InteRRaP* architecture general concept. External interface and reactive behavior 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). Cooperation level is based on both modules.

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 3. Methods used in *BPsim* products

An example, illustrating decision search diagram workflow, is presented on Fig. 8. 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.

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

1. Problem domain 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. 7), as well represents agent knowledge base, based on frame-concepts. So, agent knowledge base may be defined in two ways: productive and frame-concept – based. Here is an example of wizard implementation. This one focuses on analysis an re-engineering of business processes.

System analysis in area of business process management implies design of various system models (or views) by the analyst, each of which reflects certain aspects of its behavior. Re-engineering of these models targets the search of alternative ways of system processes development by means of consolidation of certain model sub-elements into a single process. There are three ways of achieving this. One is *parametric synthesis*, when initial model is converted into series of new models by modifying separate parameters. Another is *structural synthesis*, this is a way of forming new models by modifying model structure according certain rules. *Mixed synthesis* uses features of both options. An actual task is development of algorithm for model analysis and mixed synthesis.

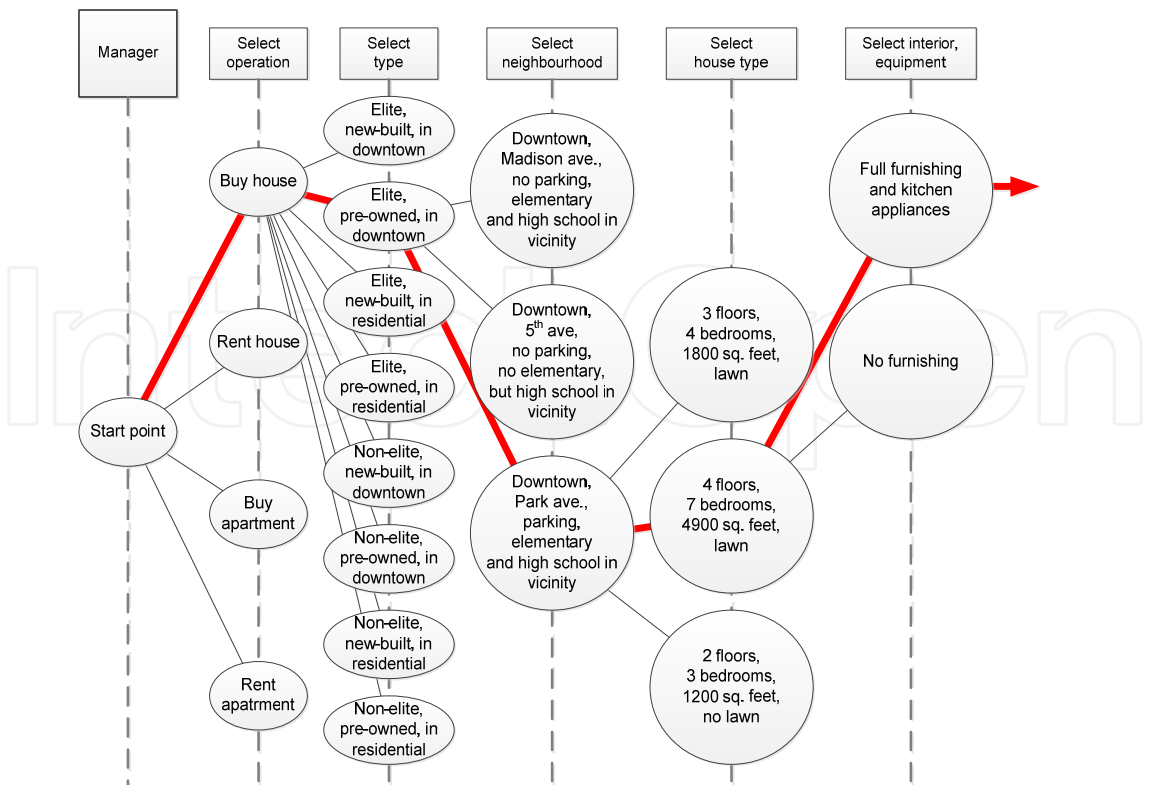


Fig. 8. Decision search tree for decision search diagram

Consider initial model M_0 of multi-agent resource conversion process. Carrying out of experiments gathers statistical data of operation execution, agent activity and tools utilization in operations. Analysis and mixed synthesis algorithm is shown on Fig. 9 in form of decision search graph. Graph nodes stand for: 0 – zero value, L – low value, A – average value, H – high value – of corresponding graph object (queue size, load or idling). Directed graph arcs connect fulfilled conditions of model analysis (nodes) into a general solution of synthesis operator application. For example, solution for application of operator “Delete parallel operation” is the following chain execution: model statistics contains zero values of average transact queue for operation, average operation load, operation idling caused by missing resources or occupied tools. Dashed arcs correspond to solutions with zero-length or short transact queue for operation. Solid arcs correspond to solutions with long transact queue for operation.

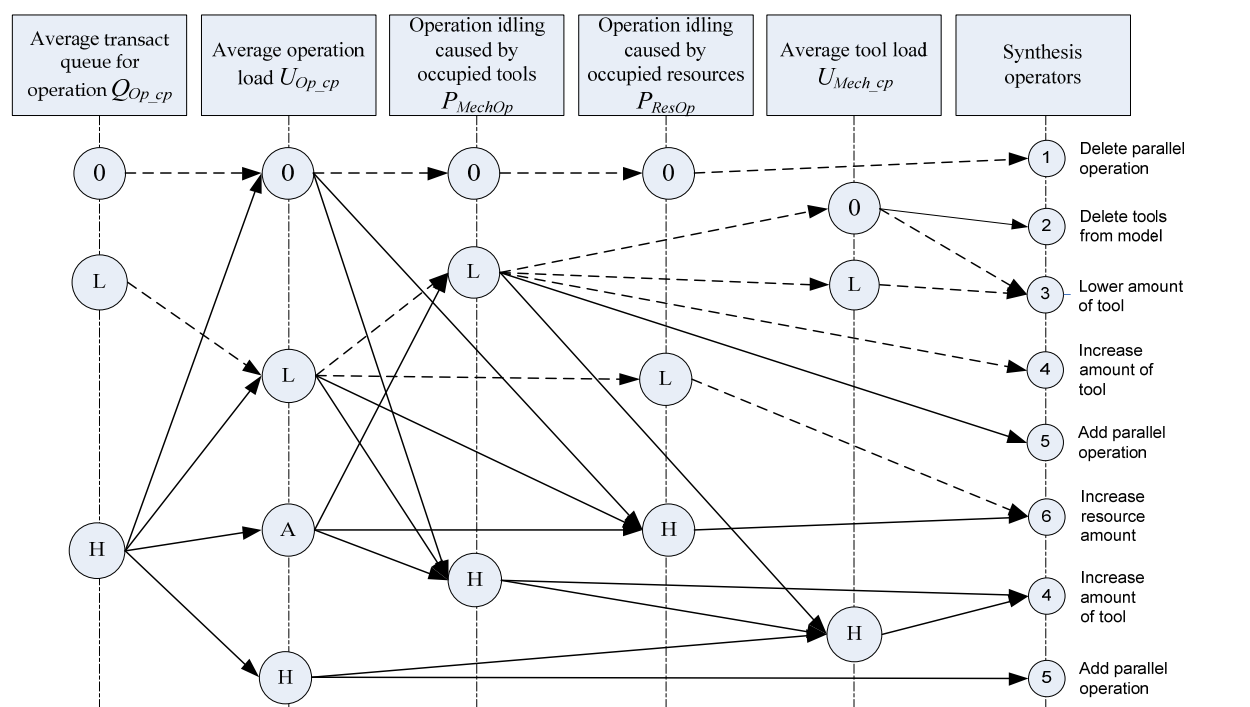


Fig. 9. Decision search graph for synthesis operators application, implemented in intelligent agent

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

5. BPsim.DSS system application

5.1 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

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 (i.e. search of decision in a force majeure situation that occurs under irresistible force or compulsion and may drastically change system certain parameters), 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.

BPsim line products were used for business processes analysis and requirements specification development for Common Information System (CIS) of Ural Federal University. University has complex organizational structure (faculties, departments, subsidiaries and representative offices), which means that success of University business processes optimization depends on quality of survey and enterprise model comprehensiveness. Survey revealed non-optimal implementation of certain processes at the University, e.g. movement of personnel (Fig. 10).

```
graph TD; S1[ ] --> S2[ ]; S2 --> S3[ ]; S3 --> S4[ ]; S4 --> S5[ ]; S5 --> S6[ ]; S6 --> S7[ ]; S7 --> S8[ ]; S8 --> S11[ ]; S8 --> S9[ ]; S9 --> S10[ ]; S10 --> S11[ ]; S11 --> S15[ ]; S11 --> S12[ ]; S12 --> S13[ ]; S13 --> S14[ ]; S14 --> S15[ ]; S15 --> S11; Agent[Agent] --> S9; style S7 stroke:#f00,stroke-width:4px; style S9 stroke:#f00,stroke-width:4px; style S10 stroke:#f00,stroke-width:4px; style S15 stroke:#f00,stroke-width:4px;
```

Generation of documents

1 Signing application

2 Application reasoning

3 Application approval

4 Passing application to asst. rector

5 Preparing statement

6 Edit statement

7 Statement approval

8 Departments approvals

9 Statement approval by asst. rector

10 Statement copy

11 Forming order

12 Order approval by asst. rector

13 Preparing statement

14 Send out statement

15 Register order in database

Agent

www.intechopen.com

Simulation model “as-is” of the process was designed in dynamic situations modeling system BPsim.MAS. Model data has been achieved from questioning employees of personnel office, IT department, and four selected dean offices. Model nodes represent document processing stages. Nodes 1-10 correspond to document processing within faculty, nodes 11-14 – processing of faculties weekly documents in personnel office. Use of intelligent agent for business process analysis and re-engineering allowed fixing certain problems.

1. Intelligent agent discovered two identical transacts within the model. Analyst decided that dean office employees prepared two similar documents on different stages – application and first statement. Recommendation – unification of these documents.
2. Intelligent agent discovered a feedback loop, which is a negative indicator for document flow business process. Analysis of highly-loaded personnel affairs-involved employees revealed that dean offices allowed generation of long document queues. Also asst. rector was required to approve 3 similar documents at different stages. Recommendation – process automation, use of electronic documents.
3. Analyst discovered that IT personnel assistance is required for database data modification. Recommendation – process automation, data changes should be carried out by information-specific personnel, IT dept. employees should not perform uncommon functions.

The following process updates were offered after analysis and re-engineering:

1. When employee signs an application, the dean office representative creates its electronic copy in common database;
2. First statement is not necessary, all information needed for the order is stored in common database, so, approvals need to be gathered only once;
3. All process participants work directly with electronic application, without requirement of sending documents to dean office;
4. Availability of storing files electronically;
5. Asst. rector approves only application and order;
6. Availability of tracing document route;
7. Employees of IT department no longer perform uncommon function of updating database;
8. In case of urgent order the corresponding information objects are generated by the dean office employee.

As a result, the “as-to-be” model no longer has nodes 7, 9, 10, 15 (Fig. 10). New business process definition became a basis for CIS module “Movement of personnel” requirements specification, which as well used diagrams, designed in BPsim.SD CASE tool. The deployment effectiveness was estimated after deployment and half-year use of the module based on simulation models “as-is” (old model) and “as-to-be” (new model). Results are presented in Table 4.

Thus, due to “Movement of personnel” 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.

Indicators	"AS-IS" model	"AS-TO-BE" model
Documents processed, per month	390	1264
Documents lost, per month	12	0
Dean office employees performance, documents per hour	0.4	0.5
Personnel office employees performance, documents per hour	2.4	7.9

Table 4. "Movement of personnel" deployment effect estimation

5.2 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 Subary 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.

5.3 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. 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 (Shvetsov, 2004). 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. 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.

BPsim.DSS was applied for MSN technical economical engineering in Ural region, covering metropolis Ekaterinburg, Russia, and satellites. Designed model is shown on Fig. 11.

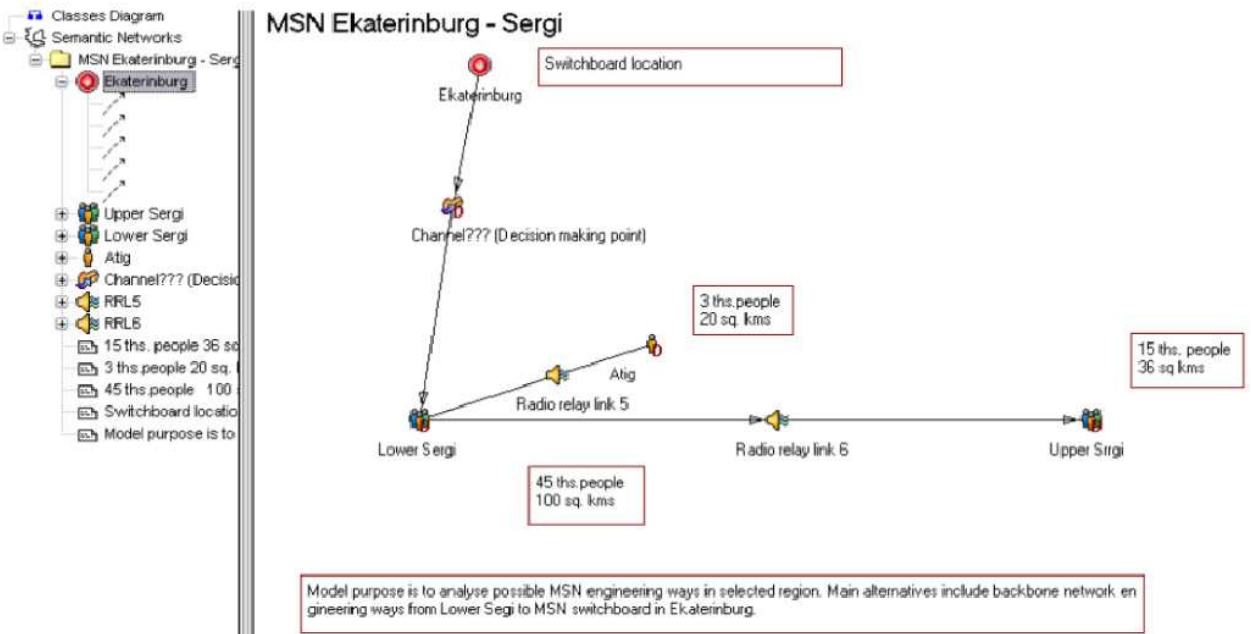


Fig. 11. MSN Ekaterinburg – Sergi model view

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. 12. 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.

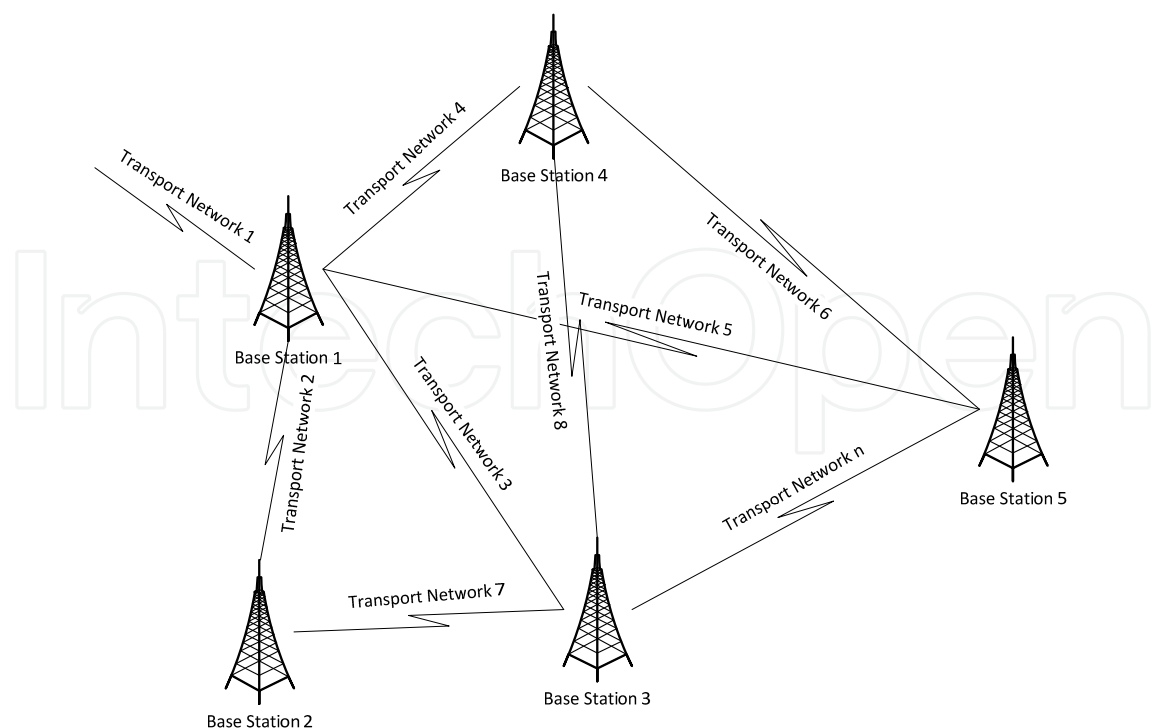


Fig. 12. Modelled MSN graphical model

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.

6. Conclusion

In this chapter we have presented the following keynote features. Some popular dynamic situations modelling systems including AnyLogic, ARIS, G2, Arena, RAO and models/prototypes of other systems 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 problem domain 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 problem domain 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:

- Problem domain 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 modeling 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 problem domain models design, data and knowledge input, logical output mechanism implementation.

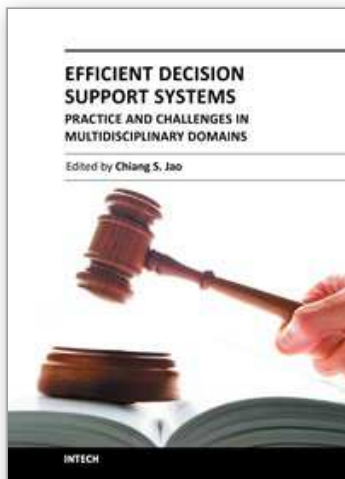
7. Acknowledgment

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

8. 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., Bykov E. A., Smolij 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. & Goncharova N.V. (2006). *Multi-agent resource conversion processes dynamic simulation*, Ekaterinburg, USTU
- Aksyonov, K.A. (2003). *Research and development of tools for discrete resource conversion processes simulation*, DPhil research paper. Ural State Technical University, Ekaterinburg, Russia
- Andreichikov, A.V. & O.N.Andreichikova (2004). *Intelligent information systems*, Moscow: Finance and statistics.
- Bugaichenko, D.Y. (2007). *Development and implementation of methods for formally logical specification of self-tuning multi-agent systems with timely limitations*. DPhil research paper. St. Petersburg, Russia.
- Bugaichenko, D.Y. (2006). Mathematical model and specification of intelligent agent systems / Bugaichenko. *System Programming*. No 2.
- Bugaichenko, D.Y. (2005). Abstract architecture of intelligent agent and methods of its implementation / Bugaichenko, Soloviev. *System Programming*. No 1.
- Emelyanov V.V., Yasinovskiy S.I. (1998). *Introduction to intelligent simulation of complex discrete systems and processes*. RAO language. Moscow, Russia
- Masloboev A.V. (2009). Hybrid architecture of intelligent agent with simulation apparatus / Masloboev, *MGTU bulletin*. Vol. 12, No 1. Murmansk State Technical University, Murmansk, Russia
- Masloboev A.V. (2008). Method of superposed generation and effectiveness estimation for regional innovation structures / Masloboev, *MGTU bulleting*. Vol. 11, No 2. Murmansk State Technical University, Murmansk, Russia
- Muller J.P. & M.Pischel (1993). *The Agent Architecture InteRRap: Concept and Application*, German Research Center for Artificial Intelligence (DFKI)
- Rybina G.V., Paronjanov S.S. (2009). SMIAI system and its application experience for building multi-agent systems / Rybina, Paronjanov. *Software products and systems*. No 4.
- Rybina G.V., Paronjanov S.S. (2008a). General model of intelligent agents interaction and methods of its implementation / *XI National Conference on Artificial Intelligence*. <http://www.raai.org>
- Rybina G.V., Paronjanov S.S. (2008b). Modeling intelligent agents interaction in multi-agent systems / Rybina, Paronjanov. *Artificial intelligence and decision making*. No 3.
- Shvetsov, A.N. (2004). *Corporate intellectual decision support systems design models and methods*, DPhil research paper, St.Petersburg, Russia
- Wooldridge M. (1995). *Intelligent Agent: Theory and Practice*, *Knowledge Engineering Review*, Vol. 10 (2).
- Wooldridge M., Jennings N., Kinny D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3. p. 285-312.



Efficient Decision Support Systems - Practice and Challenges in Multidisciplinary Domains

Edited by Prof. Chiang Jao

ISBN 978-953-307-441-2

Hard cover, 478 pages

Publisher InTech

Published online 06, September, 2011

Published in print edition September, 2011

This series is directed to diverse managerial professionals who are leading the transformation of individual domains by using expert information and domain knowledge to drive decision support systems (DSSs). The series offers a broad range of subjects addressed in specific areas such as health care, business management, banking, agriculture, environmental improvement, natural resource and spatial management, aviation administration, and hybrid applications of information technology aimed to interdisciplinary issues. This book series is composed of three volumes: Volume 1 consists of general concepts and methodology of DSSs; Volume 2 consists of applications of DSSs in the biomedical domain; Volume 3 consists of hybrid applications of DSSs in multidisciplinary domains. The book is shaped decision support strategies in the new infrastructure that assists the readers in full use of the creative technology to manipulate input data and to transform information into useful decisions for decision makers.

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 Smoliy, Olga Aksyonova, Anna Antonova and Irina Spitsina (2011). Decision Support Systems Application to Business Processes at Enterprises in Russia, Efficient Decision Support Systems - Practice and Challenges in Multidisciplinary Domains, Prof. Chiang Jao (Ed.), ISBN: 978-953-307-441-2, InTech, Available from: <http://www.intechopen.com/books/efficient-decision-support-systems-practice-and-challenges-in-multidisciplinary-domains/decision-support-systems-application-to-business-processes-at-enterprises-in-russia>

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

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](https://creativecommons.org/licenses/by-nc-sa/3.0/), 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