

**INDUSTRIAL ENTERPRISES BUSINESS PROCESSES SIMULATION WITH BPSIM.MAS**

Konstantin A. Aksyonov  
 Eugene A. Bykov

Dept. of Automated Control Systems  
 Urals State Technical University – UPI  
 Mira st., 19  
 Ekaterinburg, 620000, RUSSIAN FEDERATION

Elena F. Smoliy  
 Alexey A. Khrenov

NPP “Business Support Systems”  
 Mira st., 32  
 Ekaterinburg, 620000, RUSSIAN FEDERATION

**ABSTRACT**

Necessity to mind a huge amount of factors while developing modern enterprises models dictates new requirements for modeling software, which needs to process all data to achieve precise results, make use of all possible means for that, such as distributed calculations and introduce original approaches where possible to avoid extra time waste for multiple simulation experiments. The article focuses on the software apparatus used in distributed multi-agent resource conversion processes based tool BPsim.MAS, pointing out its advantages and describing used technologies. Second part describes BPsim.MAS deployment in Urals Industrial Group, CJSC, which allowed income and market share growth. Various pricing strategies are discussed and active/passive competitors’ behavior is considered.

**1 INTRODUCTION**

One of the prospective directions of decision support systems (based on simulation and modeling (SM) system and expert system (ES)) tool development is the problem-orientation, which allows reducing the requirements to end-user knowledge level, especially in the programming area. The paper presents results of development and deployment of the multi agent resources conversion processes (MRCP, multi-agent RCP) theory and SMES tool “BPsim.MAS”, designed on the basis of that theory.

Table 1 shows the comparison of popular simulation modeling tools, such as ARIS ToolSet (T), G2 (G), AnyLogic (A) and BPsim (B), a basis for BPsim.MAS.

As we can see, all current systems lack support of some features that might be useful in effective simulation. For example, 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.

Regarding the disadvantages a new tool was designed. BPsim.MAS is based on the proved efficient apparatus of

resource conversion processes, fully implementing multi-agent approach, utilizing analytical modeling for certain subject areas and distributed calculations for complicated processes modeling as well as introducing other convenient features. BPsim.MAS is deployed on enterprises in the Urals Region, the achieved results are presented later in this paper.

Table 1: Modeling tools comparison

Comparison criteria	T	G	A	B
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 modeling availability				
· “Agent” element	○	○	●	○
· Agents behavior models	○	○	●	○
· Agent’s knowledge base support	○	○	○	○
· Message exchange language	○	○	○	○
Simulation modeling	●	●	●	●
Expert modeling	○	●	○	○
Situational modeling	○	●	○	○

**2 MATHEMATICAL MODEL OF THE MRCP**

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 goals of RCP subject area are: design of new RCP and existing RCP perfection, resources and conversion device state forecast, process time and cost characteristics es-

timation, resources costs and mechanisms usage time estimation.

A comparison of visual problem-oriented decision support system (DSS) based on SM and ES (SMES) is already presented in this paper (see Table 1). The main disadvantages of visual problem-oriented multi-agent SMES tools such as AnyLogic, ARIS, G2/ReThink, in the area of RCP, are the complexity of the RCP definition and experiments implementation, weak resources and mechanisms conflicts modeling tools, absence of intelligent agent library, no Russian language support. In this paper we will present mathematical model of the RCP and SMES system "BPsim.MAS", which is substantially free from the above mentioned disadvantages.

Creation of the RCP mathematical apparatus is based on the widespread mathematical schemas of dynamic processes description, such as Petri nets (Avramchuk, Vavilov and Emelianov 1988), queuing systems (Avramchuk, Vavilov and Emelianov 1988; Pritsker 1984) and system dynamics models (Avramchuk, Vavilov and Emelianov 1988; Pritsker 1984; Forrester 1961). However, it is difficult enough to present all the features of the RCP with the help of specified models.

The main objects of discrete Multi-agent RCP are presented on Figure 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*).

The RCP elements take part in message exchange and perform their conversion functions on the base of their behavioral models (state graphs) following the incoming

messages. The frames are selected as an agent's knowledge representation language.

A *k*-operation (*Op<sub>k</sub>*) can be represented by the following structure:

$$Op_k = \langle f, in, out, u, h_{Op}, c_a, mech, Status_{Op}, prior_{Op} \rangle \quad (1)$$

In (1) *f* is a function implemented by an operation. It is possible to forecast an output (*out*) on an input (*in*) with the help of *f*; *in* = {*in<sub>1</sub>*, ..., *in<sub>n</sub>*} is a set of inputs, characterized by the type (material, financial, information, power, labor) and quantity; *out* = {*out<sub>1</sub>*, ..., *out<sub>m</sub>*} is a set of outputs, *out* = *f*(*in*), characterized by the type and quantity; *u* = {*u<sub>1</sub>*, ... *u<sub>y</sub>*} is a set of control commands; *h<sub>Op</sub>* = {*h<sub>Op</sub><sup>1</sup>*, ..., *h<sub>Op</sub><sup>k</sup>*} are the characteristics of the operation; *c<sub>a</sub>* is a start condition of the operation; *mech* = {*mech<sub>1</sub>*, ..., *mech<sub>q</sub>*} are the conversion devices, that are characterized by the type and quantity; *Status<sub>Op</sub>* = {*wait*, *active*, *lock*} is the operation state, defined as a finite set of states: *wait* is waiting, *active* is execution, *lock* is interruption; *prior<sub>Op</sub>* is the operation priority.

The start condition (*C<sub>a</sub>*) is defined in (2):

$$C_a(t) = C_a^{in}(t) \wedge C_a^{out}(t) \wedge C_a^u(t) \wedge C_a^{mech}(t) \wedge C_a^{status}(t) \wedge C_a^{time}(t) \quad (2)$$

In (2) *C<sub>a</sub><sup>in</sup>* is a condition of necessary input resources; *C<sub>a</sub><sup>out</sup>* is a condition of the output registration limitations; *C<sub>a</sub><sup>u</sup>* - a condition of determinant control commands availability; *C<sub>a</sub><sup>mech</sup>* is a condition of necessary conversion devices availability; *C<sub>a</sub><sup>status</sup>* is a condition of fulfillment availability; *C<sub>a</sub><sup>time</sup>* is a start on time condition.

The transition of the operation in the "execution" state is attended by execution of input resources capture *A<sub>in</sub><sup>RES</sup>* and conversion devices *A<sub>in</sub><sup>MECH</sup>*. Being in the "execution" state, the operation can jump into the "interruption" state. The operation can be interrupted to provide the execution of another operation. Passing into the "interruption" state the operation stores the moment of stopping and releases entrapped conversion devices *A<sub>Lock</sub><sup>MECH</sup>*. The presence of free conversion devices *C<sub>a</sub><sup>mech</sup>(t)* is checked during the "interruption" state. The operation will be in the "interruption"

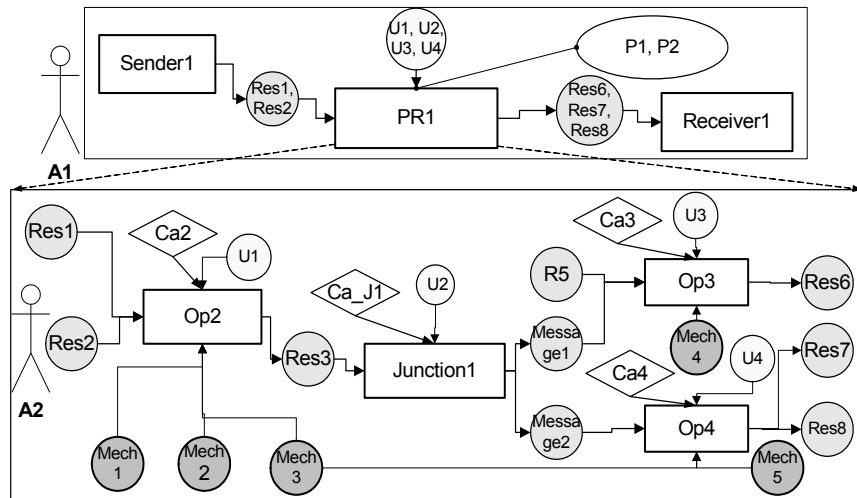


Figure 1: Hierarchical Multi Agent RCP: PR1 process works out in constituents Op2, Op3, Op4, Junction1 and corresponding resources and tools; agents A1 and A2 control different RCP levels

state until necessary conversion devices are released. In case of  $C_a^{mech}(t)=true$  the operation jumps into the "execution" state the conversion devices  $A_{UnLock}^{MECH}$  are captured and the execution is prolonged.

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: environment (current system state) analysis; state diagnosis; knowledge base access; decision-making. Thus the functions of analysis, situations structuring and abstraction, as well as resource conversion process control commands generation are performed by agents (Figure 2).

Simulation engine algorithm of agent-containing model consists of the following main stages: current point of system time calculation; agent activity processing (state diagnosis, control commands generation); conversion rules queue formation; conversion rules execution and work memory state, i.e. resources and mechanisms values, modification. Simulation engine accesses the expert system unit, i.e. agent knowledge base, in order to diagnose current state and generate control commands.

Each agent possesses its

- i. knowledge base,
- ii. set of goals that are needed for behavior configuration setting,
- iii. 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.

The operation is in the "execution" state while  $t < t_{End}^k$ , where  $t_{End}^k$  is the termination of  $k$ -operation moment. The operation jumps into the "idle" state when the condition  $t = t_{End}^k$  is met. This jump is attended by the creation of output resources  $A_{out}^{RES}$  operations and the release of seized conversion devices  $A_{out}^{MECH}$ .

The system graphs apparatus was applied for the representation of an RCP hierarchical organization and integral characteristics calculation (Avramchuk, Vavilov and Emelianov 1988):

$$\vec{PR}_{L=i} = \langle \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}_{L=i}; \{PR_{L=j}^p; p_i = 1, \dots, n_{L=j}^p\}_{j=2, \dots, i}; \{Relation_{AB}^{mk}\}_{L=i} \rangle. \quad (3)$$

In (3) the graph of  $i$ -level integration will be formatted as a result of step-by-step integration of the graphs

$\vec{PR}_1, \vec{PR}_2, \dots, \vec{PR}_{i-1}$  with  $j$ -level integration processes (sub-processes) set formation  $\{PR_{L=j}^p; p=1, \dots, n_{L=j}^p\}$  on each  $j$ -stage,  $L$  - level of integration. Units of the RCP set  $\{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}_{L=1} \subset \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}_{L=1} \subset \dots \subset \{Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m\}$  and resource ratio set  $\{Relation_{AB}^{mk}\}_{L=1} \subset \{Relation_{AB}^{mk}\}_{L=1} \subset \dots \subset \{Relation_{AB}^{mk}\}$  of the system graph  $\vec{PR}_{L=1}$  are RCP units and units resource ratio, also  $Sender^m \cup Op^m \cup Receiver^m \cup Junction^m \cup Agent^m$  units and  $Relation_{AB}^{mk}$  resource ratio of the first level of integrations system graph

$\vec{PR}_{L=j}$  which doesn't include step-by-step integration in one process  $PR_{L=j}^p$  (Figure 3).

Each top of the system graph RCP is characterized by attributes (metrics) set  $h_1, \dots, h_z$ . All sets of system graph tops attributes specify attributive set. The calculation of integral metrics of processes (system tops)  $h_1, \dots, h_z$  on an arbitrary  $i$ -level ( $i > 0$ ) is set above the tops of the  $(i-1)$ -level of integration.

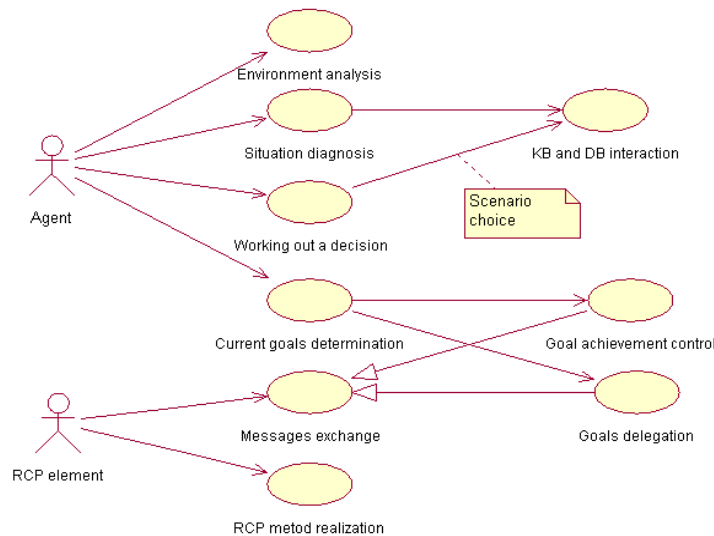


Figure 2: Use-case diagram determines relations between agents and RCP elements

The possibility of the use of typical process description models for the creation of the mathematical model of the RCP is examined; the typical process description models are: the augmented Petri networks; the queuing systems. It's shown that given models don't allow making an adequate representation of the RCP. There are some disadvantages revealed for the Petri nets: absence of timing; absence of the concurrent activities conflicts; the lack of the division of mark types (resources types); models of real processes described in the terms of Petri nets are bulky and badly readable. Owing to the fact that the change of the  $N_E$  – schemas has only two positions, there is no ability of process interrupt modeling. The conceptual apparatus of the Q-schemas are not corresponding with the problem area of the RCP, Q-schemas are oriented on the modeling instruments activity, and in the RCP there are modeling of the consecution and parameters of the conversion processes.

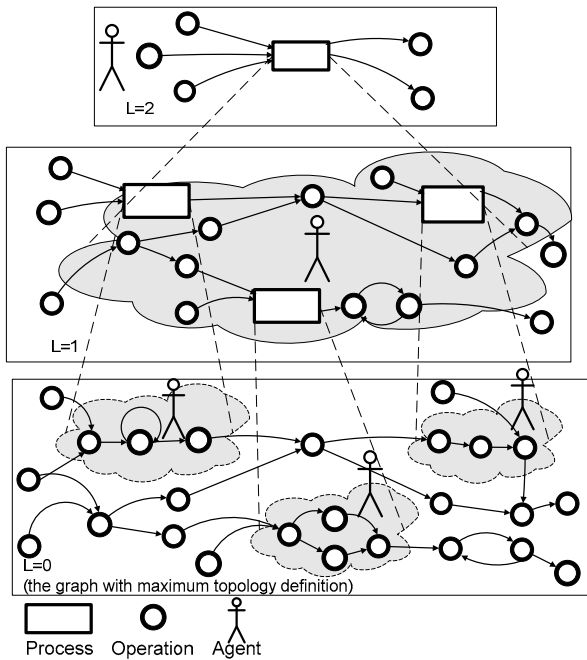


Figure 3: System graph of multi-agent RCP

The apparatus production rules are used for building a kernel of simulation system (Jackson 1998, Newell 1973). The structure of the production rules of the RCP system is defined as:

$$PS = \langle Rps, Bps, Ips \rangle \quad (4)$$

In (4)  $Rps = \{RES(t)\} \cup \{MECH(t)\}$  is the current state of resources and conversion devices (operative storage);  $Bps$  is a set of resources transformation rules (knowledge base);  $Ips$  is an inference engine.

The interpreter can be represented by:

$$Ips = \langle V\_ips, S\_ips, K\_ips, W\_ips \rangle, \quad (5)$$

In (5)  $V\_ips$  is the choice process from  $Bps$  and from  $Rps$  subset of the active rules  $Bps\_v$  and active data  $Rps\_v$ , which will be used in the next interpreter cycle;

$S\_ips$  - the rules and data comparison process;

$K\_ips$  - conflicts resolving process (or planning process), which is defining, what comparisons will be fulfilled;

$W\_ips$  - selected compared rule execution process. The results are the data modification in  $Rps$ .

The structure of a transformation rule which corresponds to the structure of the operation  $Op_k$  is defined:

$$RULE_k = \langle Ca(t), A_{IN}(t_{Ca}), A_{Lock}^{MECH}(t_{Lock}), A_{UnLock}^{MECH}(t_{UnLock}), A_{OUT}(t_{End}), Status^{RULE}, time_{RULE}, prior, kind\_prior, break\_off \rangle, \quad (6)$$

$$ALock(t_{Lock}) = A_{Lock}^{MECH}(t_{Lock}), A_{Lock}^{RES}(t_{Lock}) \quad (7)$$

$$A_{UnLock}(t_{UnLock}) = A_{UnLock}^{RES}(t_{UnLock}), A_{UnLock}^{MECH}(t_{UnLock}) \quad (8)$$

In equations (6), (7), (8)  $A_{IN}(t_{Ca}) = A_{in}^{RES}(t_{Ca})$ ,  $A_{in}^{MECH}(t_{Ca})$ ;  $A_{OUT}(t_{End}) = A_{out}^{RES}(t_{End})$ ,  $A_{out}^{MECH}(t_{End})$ ;  $Status^{RULE} = \{wait, active, lock, done\}$  are the states of a rule, where *done* - the rule is fulfilled;  $time_{RULE}$  is the duration of conversion process; *kind\_prior* is the type of priority (relative, absolute);  $break\_off = \{true, false\}$  is the tag of the interruption prohibit, if there is "true" – the rule cannot be interrupted;  $A_{Lock}^{RES}$  is the resources expenditure, including the time resource, necessary for stopping the operation  $Op_i$ ;  $A_{UnLock}^{RES}$  is the resources expenditure, including the time resource, necessary for resuming the operation  $Op_i$ . The rule status transition graph is presented on (Figure 4).

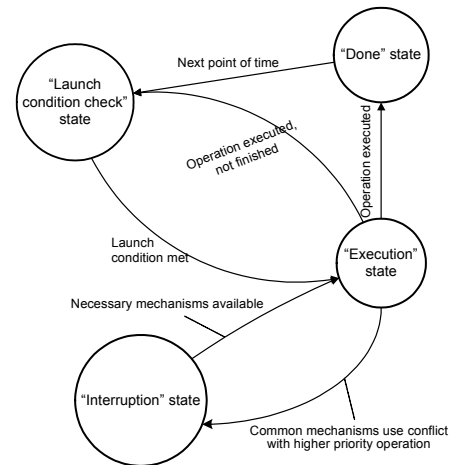


Figure 4: The rule status transition graph

The value  $Ca(t)$  is checked during «Launch condition check» status. If  $Ca(t) = true$ , then the rule in the same instant ( $t=t_{Ca}$ ) passes in a "activity" status  $Status^{RULE} = "active"$ . Operations  $Active_{IN}(t_{Ca})$  fulfilled in this status are: capture of the necessary input resources and mechanisms, calculation of costs and temporary metrics of a rule.

The rule transition in the "interruption" status can be caused by necessity of resources, which are essential for

the higher priority rule releasing. In this status the mechanisms  $Active_{Lock}^{MECH}$  are released and the stop moment is stored.

Free resources availability is checked during the "interruption" status  $C_a^{mech}(t)$ . The rule is in the interruption status until the necessary mechanisms will be released. In case of successful check, the rule passes in a "fulfillment" status:  $Status^{RULE} = "active"$ . The transition in this status is accompanied by the mechanisms capture operation  $Active_{UnLock}^{MECH}$ .

At the time  $t=t_{End}$  the rule is finished (transition to the "fulfilled" status), the operations  $Active_{OUT}$  are fulfilled: the output resources are created, the captured resources are released, the values of output parameters are calculated and the message of the process completion is generated.

It is accepted, that the conflicts can arise at the process execution in the following cases: on input resources; on the mechanisms. The conflicts resolving between different types of the rules ("operation", "junction", "source", "receiver") are realized on the rules execution planning mechanism, which are imbedded in the inference engine.

Inside the "operation" rules the order of execution is regulated on the priority - the maximum priority rule will be treated earlier. For each "operation" rule the priority is set at the model construction.

Each rule is characterized by the priority class: relative or absolute, and also by the possibility of the interruptions prohibition. The scale of priorities for the rules with relative and absolute priority is common. The priority varies from 0 up to M (0 - best priority).

In existing simulation modeling systems the mechanism of the inference engine will be realized as follows (Pritsker 1984): the rules are activated by the inference engine; thus the truth of a condition (IF) is checked; if it is true, the output machine fulfils operations were in the conclusion (THEN).

The operation algorithm of the inference engine consists of the following main stages: 1) definition of a current instant  $SysTime = \min T_j, j \in \{RULE\}$  (standard algorithm of the discrete-event simulation (DES)); 2) agent's actions processing (current situation diagnosis, executive instruction working-out); 3) queuing of transformation rules; 4) execution of transformation rules and operative storage state transition. Imitator applies to the expert system module for the current situation diagnosis and executive instruction working-out.

The main disadvantage of the direct and return output, that is usual for a static expert system, is the unpredictability of the execution time. For dynamic systems direct and return output with an exhaustive search possible to execution rules is the inadmissible luxury.

With the purpose of calculation minimization it is offered to use the inference engine algorithm, in which the model time registration is organized with the use of the centralized events calendar.

The main methods of speeding up include: event method of promoting on time, lists of different types of the rules, bringing in additional attribute to structure of the working storage units. The use of these methods in the evaporator machine-building production simulation allows reducing the time of realization of one machine experiment from 7 hours 53 minutes to 48 minutes. The machine-building process consists of 420 technological operations. That confirms an Ulrich hypothesis about the registration of events allocation in time experimentally. He has achieved high saving of a machining time; using that in digital logical networks leads to the only 1 % of units is simultaneously active (Avramchuk, Vavilov and Emelianov 1988). In the evaporator task only 3 % of the average were active. Comparison of algorithms speeds up exhaustive search (A\_p) and it is optimized (A\_opt) (Figure 5).

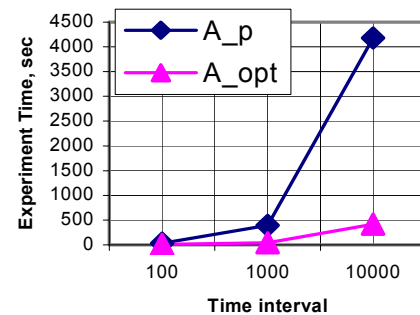


Figure 5: Comparison of algorithms

### 3 MULTI-AGENT SMES SYSTEM "BPSIM.MAS"

The problem-oriented SMES system is developed on the basis of surveyed model. SMES package of the RCP "BPsim.MAS" is worked out on the basis of the following means: Borland Delphi 7 and database control system MS SQL Server. SMES system "BPsim.MAS" provides execution of the following functions: the creation of dynamic model RCP (Aksyonov, Klebanov and Khrenov 2003; Aksyonov et al. 2005; Aksyonov et al. 2006); simulation; analysis of the simulation experiment results; obtaining reports on results; export of experimental results in MS Excel and MS Project formats.

Phases of work with BPsim.MAS are presented on Figure 6. The process consists of the following phases: subject area conceptual model design; MRCP situational model design and experiments carrying out. The system is based on the problem-oriented simulation system BPsim (Aksyonov, 2003). Such approach to dynamic situations modeling systems (DSMS) is discussed in (Philippovich, 2003) where a DSMS is based (extends) a different problem-oriented simulation system.

The system utilizes a frame-based approach (Shvetsov, 2004) for the purpose of subject area conceptual model design. This approach relies on frame-like structures and

conceptual graphs constructions overlapping (Sowa, 1976; Sowa, 1984; Sowa, 2000). Differentiation of active and passive frames and object behavior consideration are among the advantages of such approach.

Multi-agent RCP situational model design process consists of the phases, presented on Figure 6.

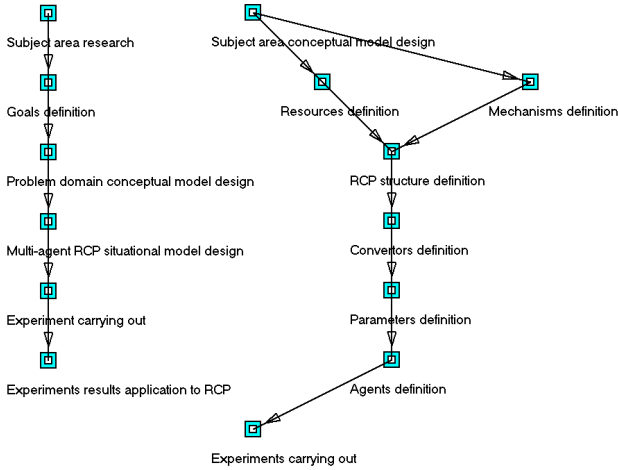


Figure 6: Work with BPsim.MAS, phases (left) and MRCP situational model design stages (right)

#### 4 “BPSIM.MAS” DEPLOYMENT

BPsim.MAS was used for multi-agent dynamic model development of Urals Industrial Group, CJSC (further referenced as UIG). The main reason for modeling is UIG behavior algorithm and pricing strategy development, targeting share of the market growth and transition to higher technological level, increasing enterprise competitiveness. Fragment of the model is presented on Figure 7.

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). Model describes the following enterprise processes: production process; sales process; products installation to end-user; after-sales service.

A number of experiments targeting the search of effective pricing strategy and considering various agents- competitors behavior sets (active/passive) were run. Figure 11 presents the output data, which are various strategies, resulting in two small competitors displacement from the market.

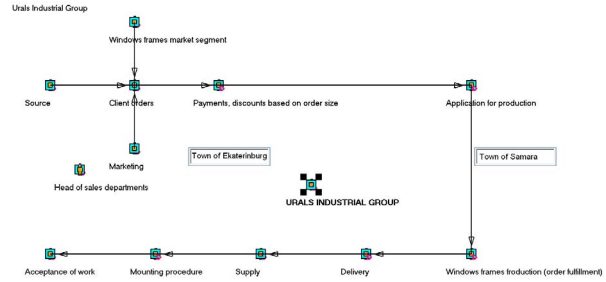


Figure 7: UIG, CJSC main process, presented in BPsim.MAS

If an existing model does not implement agent-based approach, one can easily add agents to it. After the agent properties are set we can switch on or off its participation in simulation. Figure 8 demonstrates agent properties window. Goal setting and its achievement control is presented on Figure 9.

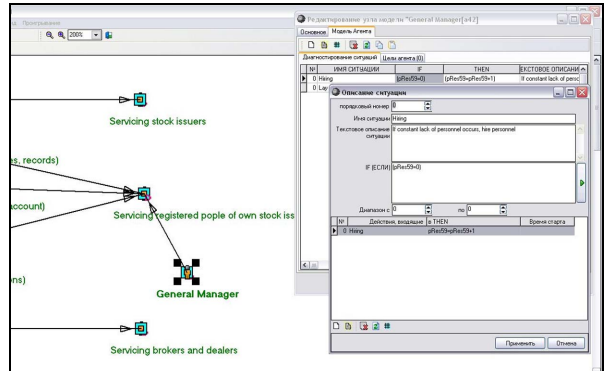


Figure 8: Defining agent settings

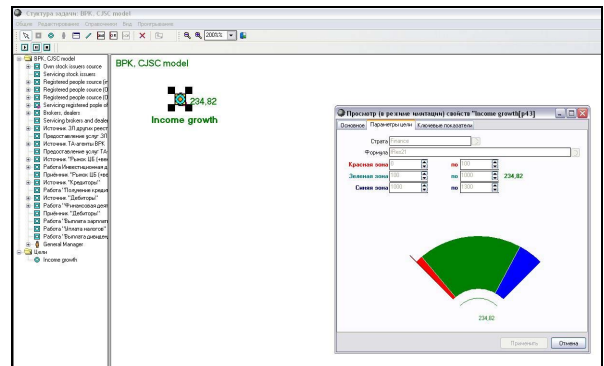


Figure 9: Goal achievement progress

Agents do have equivalent decision-making individuals’ models (head of sales, production, engineering departments, etc.) or competitive enterprises models. Figure 10 presents a knowledge base fragment of a certain agent, “Art-Line” company, which is a competitive company for the Urals Industrial Group. The knowledge base meets the



BPsim.MAS format and is filled within this product. Unfortunately it is impossible to translate the whole knowledge base into English, so only the captions were translated. An operator (BPsim.MAS end user) is able to define agent's action in every particular situation. In order to do this he needs to fill the required data including IFs and THENs. The table on Figure 10 has several other fields. № is the a sequence number of a particular situation. SITUATION NAME corresponds to the common name of the situation, e.g. "Reaction to price changes". IF is defined with system language and corresponds to the launch condition of the situation, e.g. when competitive price is lower than 1000 units, which might look like 'fRes95>1000' in system language. THEN looks much alike with IF, is defined in system language and corresponds to situation action, e.g. lowering own price or 'fRes99:=fRes95-5' in system language. DESCRIPTION field is used to fill additional remarks on the situation, e.g. "The reaction to price changes lowers own price by 5 units". The given situation example is very simple, the language allows much more complicated definitions which can be seen on Figure 10, language independent fields IF and THEN.

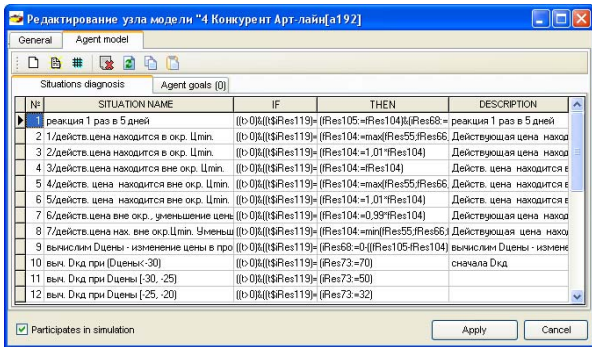


Figure 10: Competitive agent's knowledge base

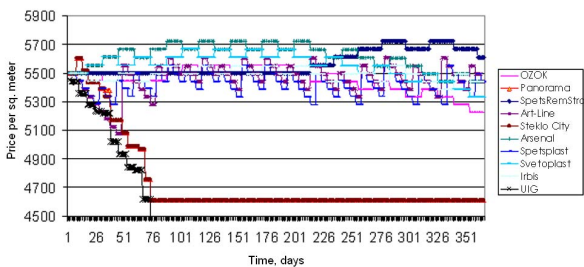


Figure 11: Pricing strategies resulting in competitors displacement from market

A number of experiments targeting the search of effective pricing strategy (among those presented on Figure 11) and considering various agents- competitors behavior sets (active/passive) were run. Figure 12 presents the output da-

ta, which are various strategies, resulting in two small competitors' displacement from the market.

Figures 13 and 14 present the dependencies of market share and price with an active competitors' behavior model and abrupt UIG price changes, resulting in enterprise's segment of market growth, made possible due to its leading role in price dictation together with unavailability of larger competitors to change prices rapidly, having longer price reaction time.

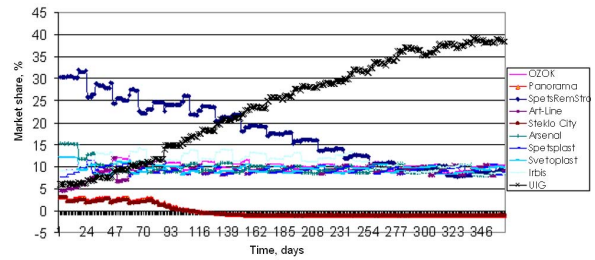


Figure 12: Competitors displacement from market (competitors' passive behavior)

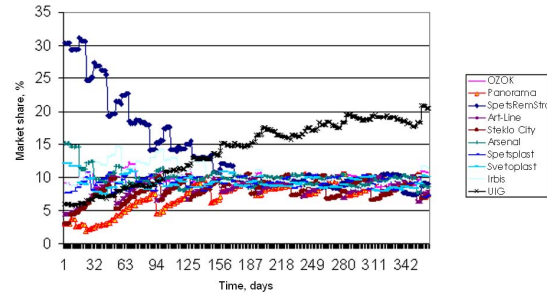


Figure 13: Market share changes: abrupt price changes together with active competitors' behavior

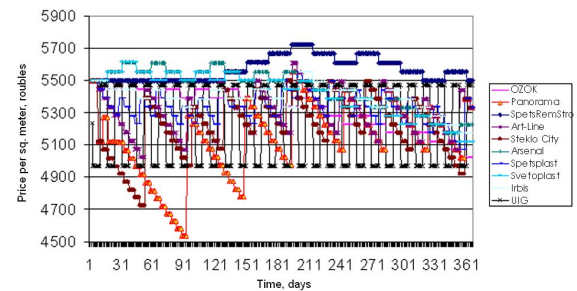


Figure 14: Competitors' reaction to abrupt price changes

Smooth price changes together with active competitors' behavior model do not result in noticeable segment of market growth.

Figure 15 presents UIG income variation depending on various pricing strategies. Rows on Figure 15 represent

the following strategies, according to row number in the legend:

1. Smooth price decrease from 5500 RUR in 50 RUR intervals per iteration for a period of one year. After reaching 4500 RUR price increases to 5500 RUR with competitors' passive behavior.
2. Abrupt price decrease from 5500 to 4500 RUR in the first quarter with competitors' passive behavior.
3. Soft price decrease from 5500 to 4500 RUR in the first half-year with competitors' passive behavior.
4. Soft price decrease from 5500 to 4500 RUR in the first half-year with competitors' active behavior.
5. Abrupt price decrease from 5500 to 4600 RUR in the first quarter with competitors' active behavior.
6. Periodical price decrease from 5100 to 4600 RUR (intermittent pricing strategy) taking into consideration competitors' activity with competitors' active behavior.
7. Periodical price decrease from 5470 to 4970 RUR taking into consideration competitors' activity with competitors' active behavior.
8. Abrupt price decrease from 5500 to 4280 RUR in the first two months with competitors' active behavior.
9. Abrupt price decrease from 5500 to 4200 RUR in the first two months with competitors' active behavior.
10. Soft price decrease from 5500 to 4160 RUR in the first seven months with competitors' active behavior.
11. Soft price decrease from 5500 to 4180 RUR in the first five months with competitors' active behavior.
12. Abrupt price decrease from 5500 to 4300 RUR in the first two months with competitors' active behavior.
13. Abrupt price decrease from 5500 to 4300 RUR in the first two months with competitors' passive behavior.
14. Soft price decrease from 5500 to 4160 RUR in the first half-year with competitors' passive behavior.

Rows 6 and 7 are based on abrupt price changes and active competitors' behavior model. Price and market share variation for row 7 were presented earlier on Figures 13-14. Rows 2 and 5 show reasonable income rate, but these experiments were carried out with passive competitors' behavior model.

Figure 16 presents UIG market share variation depending on various pricing strategies. Rows 6 and 7 based on abrupt price changes and active competitors' behavior model correspond to annual income of 1.9 and 2.6 million dollars together with market share growth to 22 and 20 percent respectively. Experiments 2 and 5 show reasonable results in market share but consider passive competitors' behavior which cannot be relied on. Experiment 8 does not meet the income requirements. The difference between experiments is caused by the variation of initial values, model characteristics, etc. Graphics are achieved with the use of BPsim.MAS output data that is collected from experiments.

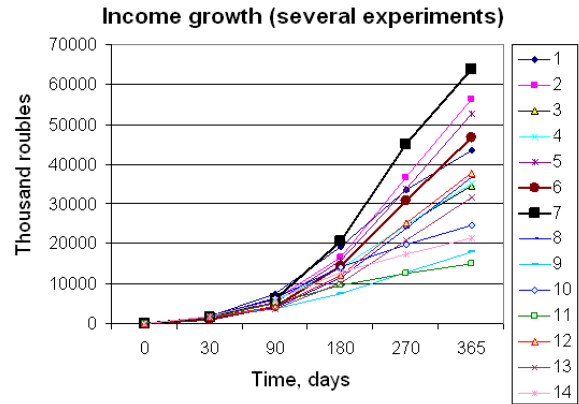


Figure 15: Income depending on various pricing strategies

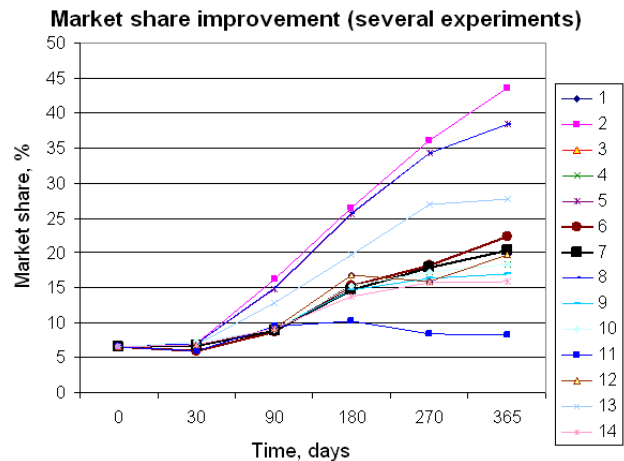


Figure 16: Market share depending on various pricing strategies

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 modeling 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.

## 5 SUMMARY

The designed dynamic situations multi-agent RCP modeling system BPsim.MAS is based on discrete-event simulation modeling and is integrated with an expert system. Multi-agent approach to resource conversion processes dynamic model increases its intellectuality with expert, situ-



ational and simulation modeling integration, as well as allows new problem classes to be solved. Such problems include management processes modeling and resolving of resource limitations based conflicts emerging between decision-making people in multi-agent environment.

#### ACKNOWLEDGEMENTS

The research is carried out with “Start” program support under state contract #5058P/7296 between small scientific and technical sphere enterprises extension work fund and “NPP Business Support Systems Ltd” as well as within the bounds of the President of Russian Federation grant #MK-2208.2007.9.

#### REFERENCES

- Aksyonov, K., B. Klebanov, and A.Khrenov. 2003. Computer-aided design system of simulation business process model, *Proceedings of the 4th IMACS Symposium on Mathematical Modeling*, ARGESIM Report no. 24. – Austria, Vienna University of Technology.
- Aksyonov, K.A., 2003. *BPsim simulation system user manual*. Ekaterinburg, 58 p.
- Aksyonov, K.A., E.F. Smoliy, N.V. Goncharova, A.A. Khrenov, and A.A. Baronikhina. 2005. Development of Resource Conversion Processes Model and Simulation System // *Proceedings of the EUROCON 2005*. – Serbia & Montenegro, Belgrad.
- Aksyonov, K., E.Smoliy, N.Goncharova, A.Khrenov, and A.Baronikhina. 2006. Development of Multi Agent Resource Conversion Processes Model and Simulation System, *Computational Science, ICCS 2006: 6th International Conference, Proceedings, Part III*. Lecture Notes in Computer Science, Volume 3993. – UK, Reading.
- Avramchuk, E.F., A.A.Vavilov, and S.V.Emelianov. 1988. *Technology of system simulation*, M.: machine construction industry; Berlin: Techniques, 520 p.
- Forrester, J. 1961. *Industrial Dynamics*, Cambridge, MA: MIT Press.
- Jackson, P. 1998. *Introduction to Expert Systems*, West Group, Rochester, NY, Addison-Wesley.
- Newell, A. 1973. *Production systems: models of control structures // Visual information processing*, New York: Academic Press.
- Philippovich, A. Yu., 2003. *Situational, simulation and expert modeling integration in printing industry*. Moscow, p. 310.
- Pritsker, A. A. B., 1984. *Introduction to simulation and SLAM II*. System Publishing Corporation, West Lafayette.
- Shvetsov A.N., 2004. *Corporate intellectual decision support systems design models and methods*. DPhil research paper. St.Petersburg, Russia, p.461.
- Sowa, J. F., 1976. *Conceptual graphs for a database interface*. IBM Journal of Research and Development, № 20:4. P. 336-357.
- Sowa, J. F., 1984. *Conceptual Structures : Information Processing in Mind and Machine*. Reading, MA : Addison - Wesley. 481 p.
- Sowa, J. F., 2000. *Knowledge Representation : Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA : Brooks / Cole Publishing Co. 594 p.

#### AUTHOR BIOGRAPHIES

**KONSTANTIN A. AKSYONOV** (1977) received the computer science engineer degree from the Ural State Technical University (USTU – UPI) in 1999. He has received his Ph.D. from the USTU – UPI in 2003. Aksyonov is the assistant professor of the Computer-based system department of the USTU – UPI. He is a deputy dean in the sphere of the scientific research. The orientations of the research study: simulation modeling of the processes and systems, information technologies, business-processes automation, information systems designing, strategic control. A scholar of the President of Russian Federation in 2001. The winner of the contest “The future of the city Ekaterinburg – in the view of the young scientists”, 2003. The winner of the contest for the best research study work USTU – UPI, 2004. Has got a Urals Region Governor Prize for the young scientists in 2005 and a grant of the President of the Russian Federation for the young scientists in 2007. Konstantin A. Aksyonov is a IEEE Member. Contact: [<wiper99@mail.ru>](mailto:wiper99@mail.ru).

**ELENA F. SMOLIY** received the computer science engineer degree from the USTU – UPI in 2001. The orientations of the research study: simulation modeling of the processes and systems, information systems designing. A prize-winner of the region contest for the best research study work in 2001.

**ALEXEY A. KHRENOV** received the computer science engineer degree from the USTU – UPI in 1999. The orientations of the research study: simulation modeling of the processes and systems, information technologies, information protection.

**EUGENE A. BYKOV** received the computer science engineer degree from the USTU – UPI in 2006. Currently he is a graduate student of USTU-UPI. Main subject of research is computer simulation of multi-agent RCP. He is a IEEE member, System, Man and Cybernetics society member. Contact: [<omediuz@gmail.com>](mailto:omediuz@gmail.com).