# AN AGENT-DIRECTED MULTISIMULATION FRAMEWORK FOR MANAGEMENT SIMULATION GAMES

Galina Merkuryeva[a], Jana Bikovska[b], Tuncer Ören [c]

[a), (b] Departmet of Modelling and Simulation
Riga Technical University
Kalku Street 1, LV-1658 Riga, Latvia

[c] The McLeod Modeling & Simulation Network (M&SNet) of SCS
School of Information Technology & Eng. (SITE)
University of Ottawa, Ottawa, Ontario, Canada

[a]Galina.Merkurjeva@rtu.lv, [b]Jana.Bikovska@rtu.lv, [c]Oren@site.uottawa.ca

## ABSTRACT

The paper presents an overview of agent-based multisimulation applications in management simulation games. The scenario-based game management concept is introduced in the paper. Agent-directed multi-simulation framework for management simulation games is developed. An illustrative example for a specific web-based logistic simulation game is provided.

Keywords: multisimulation, multi-models, agent-based simulation, agent-supported simulation

## 1. INTRODUCTION

Computer simulation games are widely used in management education to provide extensive experience under control conditions. In general, they are contrived situation which imbed players in a simulated business environment, where they make management-type decisions from time to time, and their choices at one time generally affect the environmental conditions under which the subsequent decisions must be made. In other words the main purpose of management simulation game is to simulate interactions between business actors, so they are well suited for using agent-directed approaches. Multisimulation is a novel simulation methodology allowing simultaneous explorations of different simulation scenarios which plays very important role within the structure of simulation games.

Agent-directed simulation that integrates agent and simulation paradigms has two distinct categories (Ören, 2000; Yilmaz, 2005; Yilmaz and Ören, 2006): (a) use of simulation for agents or agent simulation, and (b) use of agents for simulation (agent-supported simulation and agent-based simulation). Agent simulation, i.e., simulation of agent systems is used in several application areas: manufacturing systems, robotics, transportation, logistics, e-commerce, etc. Agent-supported simulation involves the use of intelligent agents to improve simulation and gaming infrastructures and environment. Agent-based simulation is the use of agent technology to generate or monitor model behavior in a simulation study. In this study, in addition of agent simulation, agent-supported and agent-based simulation is used for the development of management games.

## 2. AGENTS IN TUTORING SYSTEMS

Agent technology is widely used in different tutoring systems. For, example, in (Hospers et al., 2003) such an agent-based system for nurse education is discussed. Here authors describe the development of an effective teaching environment called Ines or Intelligent Nursing Education Software that uses agents to support learning and is able to provide sensitive feedback, demonstrations of practical exercises and/or explanations. Two architecture styles are used in Ines: (1) ”data abstraction and object-oriented organization,” and (2) ”event based, implicit invocation.” The first one means that single components are represented by objects that have their own attributes and methods. Objects could be identified as physical objects used by nurse in practice (needles, swaps, etc) and agents used in learning process. Objects are implemented by using several classes within Java. The second architecture style is used to implement agents’ behaviour in the tutoring system. Each agent is capable of transmitting messages to other agents, and receiving messages by observing all messages that are sent to that agent or that are broadcasted to all agents.

Ines architecture consists of four main parts: input from devices, a concrete user interface, an abstract user interface and the cognitive part which includes student, instructions, domain, and patient models. Two types of software agents, i.e. proactive and reactive agents, are used in the instruction model to provide teaching instructions to a nursing student. Each proactive agent observes the user interface and send an observation to a reactive agent. Reactive agents receive the observations and respond to them. The following are examples of proactive (a) and reactive (b) agents that define their function, i.e., (a) Time Agent, Error Agent, Task Observer and Sterile Agent, and (b) Feedback Agent, Interaction Observer Agent and Examination Agent. To implement agents, the agent platform developed by the Parlevink group is used.

Capuano et al. (2000) present an innovative architecture for Agent Based Intelligent Tutoring System (ABITS) by using the Multi-Agent paradigm. ABITS extends a traditional Course Management System (CMS) with a set of intelligent functions which allow student modeling and automatic curriculum generation and aimed to improve the learning effectiveness based on the adaptation of the teaching material to student skills and preferences. ABITS knowledge indexing is based metadata about the learning objects and their interrelations, and conceptual graph that is used to link concepts underlying the knowledge domain, i.e., prerequisite, sub-concept, and so on. To implement a multi-agent paradigm in ABITS, Java-based Agent framework JAFMAS is used. ABITS implements three main types of agents, one for each base function topology. Evaluation Agents function is evaluating and updating of the Cognitive state that contains the knowledge degree, reached by a particular student. Affective Agents perform evaluating and updating Learning Preferences that enclose information about the student perceptive capabilities. And Pedagogical Agents perform evaluating and updating Course Curriculum depending on learning goals, cognitive states and learning preferences. Additionally, to ensure communication between CMS and ABITS, Spooler Agent is embedded in the CMS module. When CMS requires an ABITS service, it requests this service to the Spooler Agent which requests services from Evaluation, Affective or Pedagogical agents as needed. The main innovative issues of ABITS system are based on adoption of software agents and distributed nature of the software architecture.

## 3. AGENTS IN MANAGEMENT GAMES

The literature review provides several examples of agents' functions in management games. For instance, in Van Luin et al. (2004), the classic supply chain simulation and management game called, as the Beer Game, is remodelled as a multi-agent based simulation using the Agent-Object-Relationship (AOR) modelling language. Authors extend and refine the classical discrete event simulation paradigm by enriching it with the basic concepts of the AOR metamodel. The resulting simulation method is called as Agent-Object-Relationship Simulation (AORS). In AOR modelling language, an entity is an agent, an event, an action, a claim, a commitment, or an ordinary object. Agents can communicate, perceive, act, make commitments and satisfy claims while objects are passive entities without such capabilities. The state of AORS system contains the simulated time, the environment state including the non-agentive environment (as a collection of objects) and the external states of all agents (e.g., their physical state, their geographic position, etc.), the internal agent states (e.g., representing perceptions, beliefs, memory, goals, etc.), and a list of future events. At the Beer Game four agents with similar behavior can be identified: the retailer, wholesaler, distributor and factory. Each agent receives orders from the downstream supply chain node and deliveries from its upstream node. Here, the incoming and outgoing orders are modelled as messages, the shipping of beer as a non-communicative action event and the reception of beer is modelled as non-action event. The reactions of agents to the incoming events are specified in the form of production type rules. For instance, the rule that is triggered periodically at the end of week (event) compares the current inventory with outstanding orders and either send the entire order or ship the maximum available quantity of beer. AORS model of the Beer Game have been used for testing different replenishment policies in supply chains, however authors did not consider how a supply chain agent could improve its performance by learning from the game past experience.

Overviews of agent based paradigms and enterprise simulation for e-learning and knowledge transmission is given by Remondino (2007, 2008). Author discusses the integration of cognitive and reactive agents to web based business games. Cognitive agents' behaviour is goal-directed and reason-based, so they may have different roles in the game, for instance, the agent can compute some strategies to be used to make profit in the simulation. That said, this agent could then be used both as a decision support system for human users – since it could foresee some results, based on its acquired experience – and as a virtual tutor, explaining the relations among certain variables (decisions) and the achieved results. Reinforcement learning algorithms are used to evaluate utility of action to take next. Reactive agents simply retrieve preset behaviours. From a structural point of view, the reactive agents can constitute some parts of the model itself, in order to make it self adaptive when facing unforeseen decisions and context (simulate customer or supplier, represent production plants, etc.). Evolutionary algorithms (for example, Genetic Algorithm) may be applied for reactive agents to solve action selection problem.

Dobson et al. (2004) investigated technologies, which may have significant impact on the business game-based training and as the most promising the agent technology is considered. Several possible functions of an agent in the business strategy game are proposed: (1) representing market entity, sometimes completely replacing a human player or working in the background as an individual assistant to the learner; (2) monitoring trainee and evaluating his/her actions online; (3) modeling consumers' behavior. Authors successfully implemented agent technology in strategy business game for the telecommunication industry. The developed game provides the assistance for learners via agents and the following advantages are found: agents demonstrate optimal behavior by making rational decisions that offers a range of new opportunities for competency development in the trainees; the ability of the rule-based systems to provide the logics leading to decisions made by the agents for the human player allow the trainee to learn from these explanations in the process of the game; if necessary, the agent-based game can be run at much faster pace that offers an opportunity to run multiple scenarios in just one class session and in such way to investigate the impact of various parameters on the business and industry as a whole. However, all suggestions made by authors have declarative nature so more information is needed on particular models and agent realization.

The development of a simulation game, REAL Business, using an agent-based approach is described by Bai et al. (2007). The proposed architecture extends the traditional ICAI (Intelligent Computer Aided Instruction) architecture, which utilize the techniques of expert systems and artificial intelligence to help a person learn interactively and consists of expert module, pedagogical module, and user model. REAL contains the following components: (1) reflective agent which stores specific information about individual learners (e.g., level of competence) to be used by the pedagogical agent, (2) expert agent which contains expert knowledge of a particular subject domain (e.g., in REAL Business, the domain knowledge is on probability and in REAL Plant - ecology), (3) pedagogical agent which compares the knowledge of an expert agent with the reflective agent and thus provides teaching strategies (for instance, for those who showed competence within the first business community, the agent may generate situations where some resources are limited), and (4) communication agent controls the human computer interaction, it observes the user's mouse click and provides users with help for using tools; clarifies learning goals; gives navigation orientations, etc. In REAL business, Students are supposed to help a store owner design business strategies to run an ice cream store. They taught a reflective agent through observing the event network, probabilities to multiple events, as well as data from the prior sales and design production rules. The outcome is observed in the simulation game by evaluating the agent's performance. Data are collected, reorganized, and displayed for students so they can justify and evaluate their predictions based on the simulation and data evaluation. To implement agents the rule-based approach is adopted and JESS (Java Expert System Shell) is chosen as an engine. REAL is embedded in a rule-based system. A rule-based system consists of facts (agent's goal, belief, desire, physical conditions, etc.), rules (procedural knowledge, instructing how an agent should act at certain conditions), and a reasoning engine that acts on them. In the conclusion, author indicates that, although, pedagogical, reflective, and expert agents are not new, their integration of all three agents within an educational simulation is unique and built framework for developing intelligent agents in simulation games.

Multisimulation is a novel simulation methodology which allows simultaneous explorations of different simulation models and/or scenarios. It was conceived by Ören (2001). Ören and Yilmaz –who also advocate agent-directed simulation (Ören, 2001; Yilmaz, 2005, Yilmaz and Ören, 2009)–contributed to multisimulation (Yilmaz et al., 2006, 2007; Ören and Longo, 2008). This article is the first one to use multi-simulation and agents in simulation games.

## 4. AGENT-DIRECTED FRAMEWORK FOR MANAGEMENT GAME

### 4.1. Scenario-based Game Management

The game simulation could be performed according to a predefined scenario chosen (or specially elaborated) by the game manager. Scenario plays very important role within the game structure and it has to be formalized for further agent implementation.

In general, a scenario can be conceived as a means of transforming the system's initial state to a final state, following main development trends, which are influenced by both internal events and external activities. It could be presented by the following 5-tuple:

$$<SI, T, E, A, SF>,$$

where:

- *SI* presents *initial state* of the system;
- *T* presents predefined system development *trends* (e.g., demand process);
- *E* presents predefined internal *events* (e.g., changes in demand process);
- *A* presents performed external *activities* (e.g., number of produced items);
- *SF* defines *results* or the system final state once the scenario has terminated.

The generic scenario structure is shown below (see Figure 1).

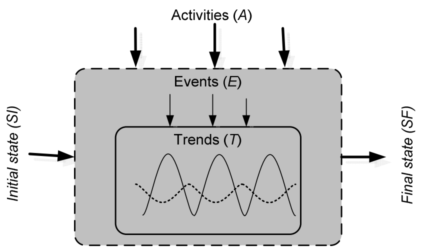


Figure 1: Generic Scenario Structure

Trends $T_i$ introduced to model system dynamics over time *t* are defined by different functions, e.g.

$$T_i = f_i(t), T_i \in T .$$

Patterns of these trends depend on the type of function and its parameters.

All internal and external influences depend on the current system state and they are aimed at achieving particular result of the system functioning. Any internal and external influences lead to short-term or long-term changes in the pattern of trends, i.e.

$$f_i^E : E \to T_i \text{ and } f_i^A : A \to T_i,$$

where $E = \{e_1, e_2, ..., e_n\}$ is a set of internal events; and $A = \{a_1, a_2, ..., a_n\}$ is a set of external activities.

System initial state is defined by the system structure as well as by value of the trends' functions at the moment when the system investigation starts:

$$SI = T_i(t_0),$$

where $t_0$ defines the initial time instant.

In terms of management simulation game, scenario specifies the initial state of game's business environment and determines the development of the game situation through time according to predefined trends as well as contains the list of important events taking place during simulation. Events are indicated by the game manager and can affect trends. Participants make their decisions (or in other words, take actions) that can affect both trends and events. Trends may be configured according to the teaching goals by changing appropriate parameters available in the game.

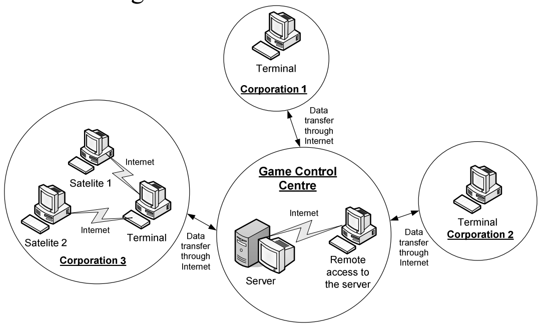

Terminal
Corporation 1
Data transfer through Internet
Game Control Centre
Internet
Satelite 1
Satelite 2
Terminal
Corporation 3
Data transfer through Internet
Server
Remote access to the server
Data transfer through Internet
Terminal
Corporation 2

Figure 2: Game Software Structure

The structure of the game software is given in Figure 2 and it can be applied to the example below. The game software consists of two parts: (1) *Game Control Centre*, which performs function of the game server, where business environment is simulated and (2) *Corporations* that ensure user interface for entering various decisions. Decisions, entered by participants are transferred through Internet to the Game Control Centre then are processed there, and their consequences in the form of different reports are transferred back to terminals. Corporations can be located at a far distance from the Game Control Centre; since all communications are performed via Internet. The game manager can view the data of all participants and apply necessary scenario alterations. Additionally, corporations can run satellite terminals for distributing decision authorities among corporation members as well as the game manager can have remote access to the game server from additional terminal.

The game is functioning in real-time mode. Decision-making and having results from the decisions made are provided in interactive real-time mode as they are in the real-world environment. Interactive mode means that decisions are made continuously when in the game model and game market situations occur which need to be reacting to by the participants. In the interactive model decisions are made as soon as they are needed or at least as soon as the decision-maker notices that the market situation needs actions from him.

## 4.2. Roles of Intelligent Agents

Intelligent agents can be introduced in the game in different ways (see Figure 3): (a) agents may perform tasks of the game manager related to the elaboration of the game scenario and control the game session, (b) play a participant role, (c) and perform a tutor function in the game.

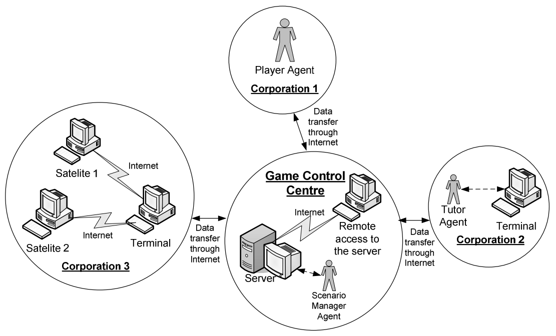


Figure 3: Agents in the Game Software Structure

## 4.3. Agent-supported Game Manager

An agent can perform tasks of the game manager (see Figure 4) related to the elaboration of the game scenario, i.e., to define and set initial conditions of the game according to a composed story or developed case study, monitoring the game session by checking trends and defining events, and making necessary changes in scenario during run-time by modifying trend functions, i.e.

```
set SI
for each j
   for each i
        define $T_i(t_j)$
        generate $e_i \in E$
        if $e_i \neq \emptyset$
           modify $T_i(t_{j+1})$
        end
   end
end
```

It is well-known from experience that the human factor significantly can create inconsistencies in the designed scenario which, with a high probability, may lead to serious mistakes in the game. Agents would have the capability to increase the consistency, completeness and other qualities of the game scenario and the reliability of its software. Production rules embedded in the game software can be used to set up parameters of the business environment according to defined policies. In order to generate production rules, inductive learning algorithms can be used. Monitoring of the game scenario is based on the dynamic comparison of parameters of the business environment and state of the companies in the game.

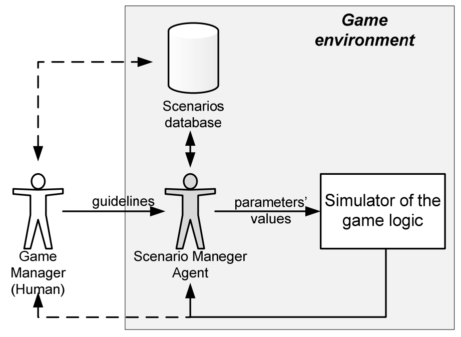


Figure 4: Scenario Manager Agent

### 4.4. Agent as a Virtual Player and/or a Tutor

If a business game of this type is used for individual training, the agent could substitute another competitive company and be introduced to simulate its behaviour, i.e. in the above algorithm, which describes behaviour of the Scenario Manager Agent, instead of event $e_i$, activity $a_i$ is introduced. The structural scheme of an agent-based simulator is given in Figure 5.

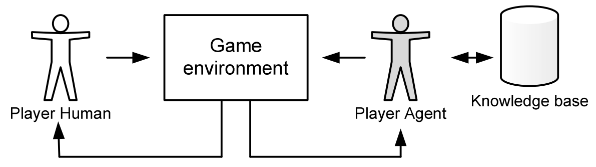


Figure 5: Virtual Player within the Game

When the agent plays a tutor role, it may define weak points in the players' activities, give necessary explanations in the form of causal relationships, and may suggest possible ways to solve the problems, if the participants need any support in their decision analysis. If the number of participants is large, it may not be so easy to manage the game due to its online mode and lack of time to make a deeper analysis of the current situation, i.e.

```
for each j
   for each i
        define $T_i(t_j)$
        generate $a_i \in E$ //player generates $a_i$
        evaluate $a_i$ //evaluate performance of
activity
        if performance $a_i$ is not satisfactory
           generate $a_i' \neq a_i$ //agent generates $a_i'$
           modify $T_i(t_{j+1})$
        else
           modify $T_i(t_{j+1})$
        end
     end
end
```

The corresponding scheme of Tutor Agent and Trainee interactions is presented in Figure 6.

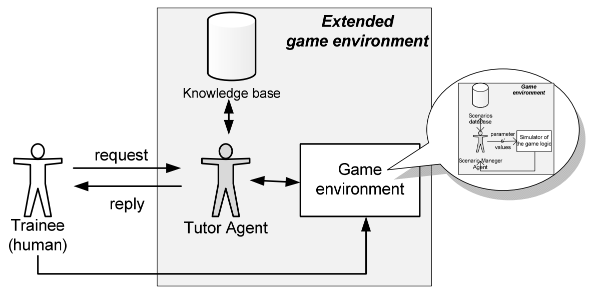


Figure 6: Tutor Agent and Trainee Interaction Scheme

## 5. MULTISIMULATION FOR GAME SCENARIOS

In general, multisimulation is a simulation, where at decision points, simulation updated may include decisions on branching of simulation studies as well as selection of models/submodels and associated parameters and experimental conditions to be analysed simultaneously and used in subsequent stage of simulation (Yilmaz et al., 2006). Therefore, from the above definition we can presume that multisimulation concept could be successfully applied for management of simulation games. Multisimulation methodology is introduced to enable exploratory simulation using various types of multimodels. A multimodel consists of several submodels where only one or some of them is/are active at a given time. The "multimodel" formalism introduced by Ören (1987, 1991) can be applied to specify scenarios.

### 5.1. Scenario Multimodel

Most of scenario management functions are described above. One of them is monitoring of the game session, and making necessary scenario alternations during run-time. Monitoring is aimed at managing key indicators of functioning of business environment. If any of indicators is out of critical value then the necessary changes should be introduced to the current scenario or, in other words, the alternative scenario should be applied that allow changing current situation according to the training goals. A multimodel is associated with a dynamic scenario that defines the sequence of scenarios in specific time moments (see, Figure 7), where *Si* is the initial scenario in the game, but *S1,..., Sn* – possible alternative scenarios, *S1., ...,S1.m* and *Sn.1,..., Sn.k* – respective subsequent scenarios.

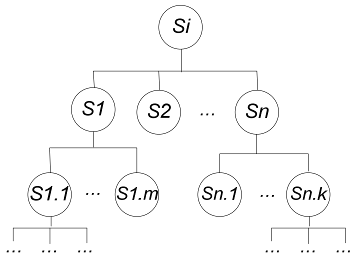


Figure 7: Scenario Multimodel

According to classification of multimodels introduced by Yilmaz and Ören (2004); Yilmaz et al. (2007), here single aspect multimodels (i.e., only one submodel is active at a given time) with static structure (i.e. model does not allow structural changes) and adaptive behaviour are introduced. In the last case constraint-driven multimodels are controlled by stationary or adaptive transitions policies that can have learning capabilities. Therefore, the task of alternative scenario choosing can be delegated to the agent.

## 5.2. Agent-Supported Scenario Multisimulation

Choosing the alternative scenario requires computational support to (1) **observe** simulation state, (2) **reason** to qualify scenario for update, and (3) **plan** for constraint-driven activation by exploring potential paths within the state space of the problem domain. Each sub-scenario is viewed as an operator that transforms the state of the simulation model. Agent paradigm provides the necessary computational infrastructure to attain these objectives.

The scenario manager agent decouples the simulator of the game logic from scenario models and it is aimed at performing dynamic scenario updates. The activation conditions of submodels are defined in terms of production system. A hypothetical submodel activation process is shown in Figure 8.

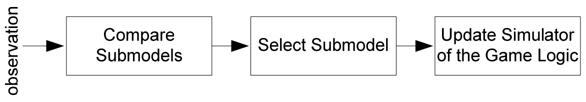


Figure 8: Submodel Activation Process

The input to the process refers to the observed condition used by the compare submodels component. Dynamic submodel replacement requires monitoring simulation conditions to determine if any of the potential state variables of interest are changed and it could be an indicator of a scenario update in the simulator. During the model comparison phase all submodels, the preconditions of which are implied by the current observed state are identified and their comparison is performed by the scenario manger agent. During the next phase only one submodel should be qualified for further exploration. Finally, simulator of the game logic is updated by the selected submodel.

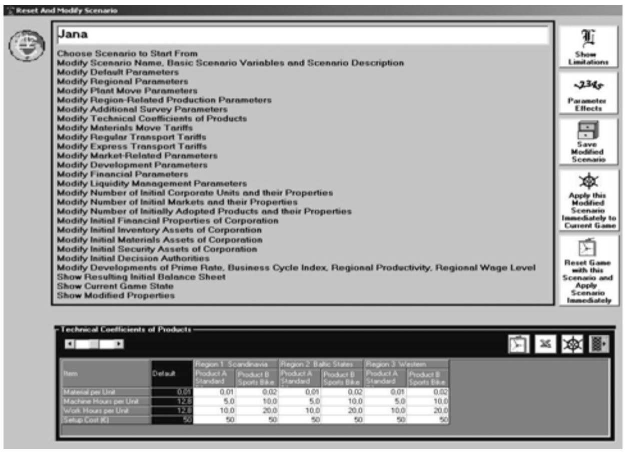
## 6. AN ILLUSTRATIVE EXAMPLE

As an example of a management simulation game for introducing agent-directed multisimulation technology, the International Logistics Management Game (ILMG) is selected (Grubbström et al. 2005). The game is aimed at providing virtual business environment for training decision-making skills in the area of international logistics, production management, finances, etc. The game can be classified as a total enterprise (or general management), interacting, computer-aided and internet-based, real-time processed business simulation. ILMG software functioning according the structure presented in Figure 2.

Different scenarios in the game are introduced to simulate real world situations and force trainees to acquire skills and experience in managing different functions of the company in various situations. During the game, participants operate as different corporations within international market, producing goods or services (i.e. making strategic and operational decisions) and competing each other in order to improve their corporation performance and to achieve the following goals: earn profit, capture substantial market share, and achieve high customer satisfaction.

The ILMG scenario is generated by the game manager according to the needs of a specific training goal and the participants' knowledge background. The business environment and the MIS reports are made as complex as in a real life or it could be simplified as necessary. Initial conditions of the game and corporations' state are set up by changing different parameters that can be divided to several groups (see, Figure 9): regional parameters (e.g., unit projecting cost, basic wage level), regional-related production parameters (e.g., productivity parameters, overheads parameters), technical coefficients of products (e.g., material per unit, setup cost), market related parameters (e.g., price effect parameters), financial parameters (e.g., prime rate, tax rate), etc.

Most of the parameters have to be defined before the game starts and they can't be changed afterwards. However some parameters (i.e., prime rate, business cycle index of the region, regional productivity index, and regional wage level) can be easily edited during the game that allows changing business environment to simulate different situations.


Reset And Modify Scenario
Show Limitations
Parameter Effects
Save Modified Scenario
Apply this Modified Scenario Immediately to Current Game
Reset Game with this Scenario and Apply Scenario Immediately

Jana

Choose Scenario to Start From
Modify Scenario Name, Basic Scenario Variables and Scenario Description
Modify Default Parameters
Modify Regional Parameters
Modify Plant Move Parameters
Modify Region-Related Production Parameters
Modify Additional Survey Parameters
Modify Technical Coefficients of Products
Modify Materials Move Tariffs
Modify Regular Transport Tariffs
Modify Express Transport Tariffs
Modify Market-Related Parameters
Modify Development Parameters
Modify Financial Parameters
Modify Liquidity Management Parameters
Modify Number of Initial Corporate Units and their Properties
Modify Number of Initial Markets and their Properties
Modify Number of Initially Adopted Products and their Properties
Modify Initial Financial Properties of Corporation
Modify Initial Inventory Assets of Corporation
Modify Initial Materials Assets of Corporation
Modify Initial Security Assets of Corporation
Modify Initial Decision Authorities
Modify Developments of Prime Rate, Business Cycle Index, Regional Productivity, Regional Wage Level
Show Resulting Initial Balance Sheet
Show Current Game State
Show Modified Properties

Technical Coefficients of Products

| Item | Default | Region 1 Scandinavia | | Region 2 Baltic States | | Region 3 Western | |
|---|---|---|---|---|---|---|---|
| | | Product A Standard | Product B Sports Bike | Product A Standard | Product B Sports Bike | Product A Standard | Product B Sports Bike |
| Material per Unit | 0.01 | 0.01 | 0.02 | 0.01 | 0.02 | 0.01 | 0.02 |
| Machine Hours per Unit | 12.8 | 5.0 | 10.0 | 5.0 | 10.0 | 5.0 | 10.0 |
| Work Hours per Unit | 12.8 | 10.0 | 20.0 | 10.0 | 20.0 | 10.0 | 20.0 |
| Setup Cost (K) | 50 | 50 | 50 | 50 | 50 | 50 | 50 |

Figure 9: Scenario Elaboration Screen

In ILMG, the game manager has to set over than a hundred different parameters for the simplest scenario (one region, one product, two bank accounts) (see Figure 10). To help perform this task special stand-alone application GameEditorPro was developed (see Figure 11). It generates a full set of the game scenario parameters according to guidelines provided by the game manager.

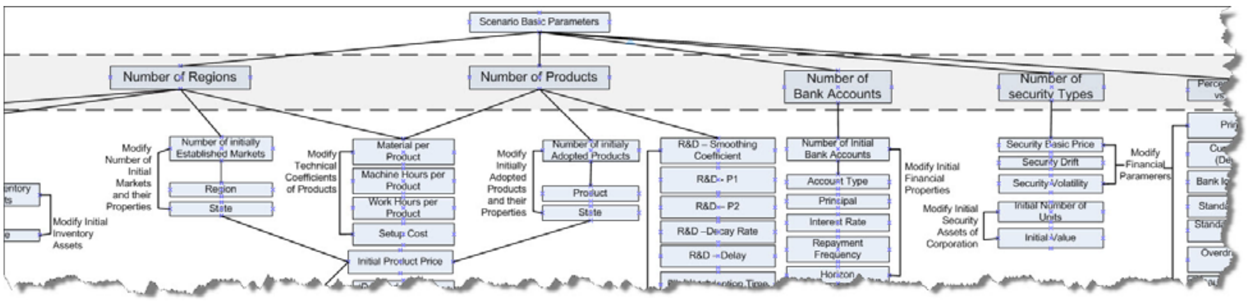
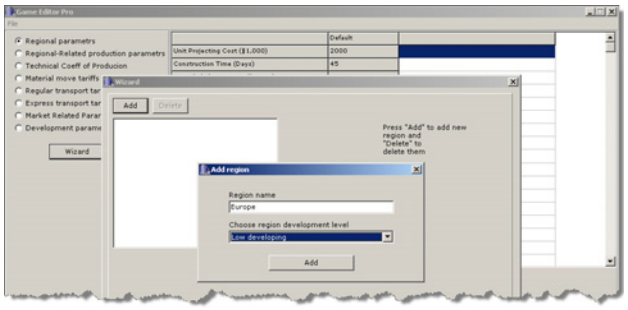

Scenario Basic Parameters
Number of Regions
Number of Products
Number of Bank Accounts
Number of security Types
Modify Number of Initial Markets and their Properties
Number of initially Established Markets
Region
State
Modify Initial Inventory Assets
Modify Technical Coefficients of Products
Material per Product
Machine Hours per Product
Work Hours per Product
Setup Cost
Initial Product Price
Modify Initially Adopted Products and their Properties
Number of initialy Adopted Products
Product
State
R&D – Smoothing Coefficient
R&D – P1
R&D – P2
R&D – Decay Rate
R&D – Delay
Number of Initial Bank Accounts
Account Type
Principal
Interest Rate
Repayment Frequency
Horizon
Modify Initial Financial Properties
Security Basic Price
Security Drift
Security Volatility
Modify Financial Paramerers
Modify Initial Security Assets of Corporation
Initial Number of Units
Initial Value

Figure 10: Examples of the ILMG parameters



| | Default | |
|---|---|---|
| Unit Projecting Cost ($1,000) | 2000 | |
| Construction Time (Days) | 45 | |

Region name: Europe

Choose region development level: Low developing

Figure 11: GameEditorPro Software Interface

The game manager decides only about number of regions and products as well as their characteristics; then the software generates all other parameters automatically. Regions have three possible characteristics: with (1) advanced, (2) less advanced and (3) developing economy. Products can be (1) hi-tech, (2) ”medium”-tech and (3) low-tech. The following algorithm is proposed for the automatic generation of parameters:

- the full range of parameters’ values is defined;
- each of defined range is divided in three equal subsets;
- each of parameters’ subset characterizes either one of region type: (1) advanced, (2) less advanced and (3) developing economy, or product type: be (1) hi-tech, (2) ”medium”-tech and (3) low-tech;
- within each subset values of parameters are generated randomly.

Further, the game is used to simulate generated scenario and to display the results. Here the procedure of dynamic scenario update is not yet explored.

## 7. CONCLUSIONS

An agent-directed multisimulation framework for scenario simulation in management simulation games is proposed in the paper. Three possible roles for agents in games are considered: scenario manager, virtual player, and virtual tutor. Though, the use of agents in simulation in general and in simulation games in particular is widespread, however combination of multisimulation with agent technology is a novel concept that seems to be very promising for use in simulation games, especially for scenario management in interacting, computer-aided and internet-based, real-time processed business simulations. The example of such game is considered in the final chapter of the paper.

## REFERENCES

Bai X. Black J.B., Vitale J., 2007. Learning with the Assistance of a Reflective Agent. URL: http://www.ilt.columbia.edu/projects/REAL_ABSHL2007_final.doc

Bikovska J., Merkuryeva G., Grubbström R.W., 2006. Enhancing Intelligence of Business Simulation Games. *20th European Conference on Modelling and Simulation.* Proceedings of the International Conference, May 28-31, 2006, Bonn, Sankt Augustin, Germany, p.641-646.

Capuano N., De Santo M., Marsella M., Molinara M., Salerno S., 2000. A Multi-Agent Architecture for Intel-ligent Tutoring. URL: http://www.capuano.biz/Papers/SSGRR_2000_P1.pdf

Dobson Mike, Kyrylov Vadim, Kyrylova Tetyana, 2004. Decision Training Using Agent-Based Business Strategy Games. *Proceedings of the Seventh IASTED International Conference: Computers and advanced Technology in Education.* August 16-18, 2004, Kauai, Hawaii, USA. Pages 66-71.

Grubbström R.W., Merkuryeva G., Bikovska J., Weber J. 2005. “ILMG: Learning Arrangements and Simulation Scenarios.” In *Proceedings of 19th European Conference on Modelling and Simulation.* Riga, Latvia, June 1-4, p. 715-720.

Grubbström, R.W., Bikovska, J., *International Logistics Management Game. Participants’ Manual*, Linköping 2003.

Hospers M., Kroezen E., Nijholt A., Op Den Akker R., Heylen D., 2003. An Agent-based Intelligent Tutoring System for Nurse Education. URL: http://wwwhome.cs.utwente.nl/~anijholt/artikelen/ctit18_2003.pdf

Ören, T.I., *Model Update: A Model Specification Formalism with a Generalized View of Discontinuity*, (Proceedings of the Summer Computer Simulation Conference, Montreal, Quebec, Canada, 1987 July 27-30), pp. 689-694.

Ören, T.I., *Dynamic Templates and Semantic Rules for Simulation Advisors and Certifiers,* In: P.A. Fishwick and R.B. Modjeski (eds). Knowledge-Based Simulation: Methodology and Application, (Springer-Verlag, New York, 1991), pp. 53-76.

Ören, T.I. (2000 - Invited Paper). Agent-directed Simulation - Challenges to meet Defense and Civilian Requirements. Proc. of the 2000 Winter Simulation Conference, J.A Joines et al., eds., Dec. 10-13, 2000, Orlando, Florida, pp. 1757-1762.

Ören T., 2001. Towards a Modelling Formalism for Conflict Management. In: *Discrete Event Modeling and Simulation: A Tapestry of Systems and AI-based Theories and*

*Methodologies. H.S. Sarjoughian and F.E. Cellier (eds.)*, Springer Verlag, New York, pp. 93-106.

Ören, T.I. and F. Longo (2008). Emergence, Anticipation and Multisimulation: Bases for Conflict Simulation. Proceedings of the EMSS 2008- 20th European Modeling and Simulation Symposium. (Part of the IMMM-5 - The 5th International Mediterranean and Latin American Modeling Multiconference), September 14-17, 2008, Campora San Giovanni, Italy, pp. 546-555.

Remondino M., 2007. An Overview of Agent Based Paradigms and Enterprise Simulation for E-Learning and Knowledge Transmission. *Proceedings of the I3M2007 Conference.* October 4-6, 2007. Bergeggy, Italy.

Remondino M., 2008**.** A Web Based Business Game Built on System Dynamics Using Cognitive Agents as Virtual Tutors. Proceedings of the Tenth International Conference on Computer Modeling and Simulation (uksim 2008). Pages 568-572.

Yilmaz, L. (ed.) (2005). Agent-Directed Simulation. SCS, San Diego, CA

Yilmaz L., A. Lim, S. Bowen, and T.I. Ören, 2007. Requirements and Design Principles for Multisimulation with Multiresolution, Multistage Multimodels._ *Proceedings of the 2007 Winter Simulation Conference*, pp. 823-832, December 9-12, Washington, D.C.

Yilmaz L., Ören T., 2004. Dynamic Model Updating in Simulation with Multimodels: A Taxonomy and a Generic Agent-Based Architecture, Proceedings of SCSC 2004 - Summer Computer Simulation Conference, July 25-29, 2004, San Jose, CA., pp. 3-8.

Yilmaz L., Ören T., 2006. Intelligent agents, simulation, and gaming. *Simulation & Gaming, Vol 37 No. 3, September 2006, p. 339-349.*

Yilmaz, L. and T.I. Ören (eds.) (2009-in Press)). Agent-Directed Simulation and Systems Engineering. Wiley Series in Systems Engineering and Management, Wiley-Berlin, Germany.

Yilmaz L., Ören T.I., Ghasem-Aghaee N., 2006 . Simulation-based problem-solving environment for conflict studies: Toward Exploratory Multisimulation with Dynamic Simulation Updating. *Simulation & Gaming, Vol 37 No. 4, September 2006, p. 534-556.* DOI (Digital Object Identifier): 10.1177/1046878106292537.

Van Luin J., Tulba F., Wagner G., 2004. Remodeling the Beer Game as an Agent-Object-Relationship Simulation, *Proceedings of Workshop 2004: Agent-Based Simulation 5*, Lisbon, Portugal, 3-5 May 2004, SCS European Publishing House, 2004.

## AUTHORS BIOGRAPHIES

**GALINA MERKURYEVA** is Professor at the Department of Modelling and Simulation at Riga Technical University, Latvia. Her research interests are in the fields of discrete-event simulation, simulation metamodelling and optimisation, simulation-based training, supply chain simulation and decision support.

**JANA BIKOVSKA** is PhD student and teaching assistant at the Department of Modelling and Simulation, Riga Technical University, Latvia. Her professional interests are in the field of supply chain management, business simulation games, scenario approach in complex systems with application to simulation games.

**TUNCER ÖREN** is a professor emeritus of Computer Science at the University of Ottawa. His current research activities include advanced M&S methodologies such as multimodels, multisimulationand emergence; agent-directed simulation; cognitive simulation; and reliability and quality assurance in M&S and user/system interfaces. He has also contributed in Ethics in simulation as the lead author of the Code of Professional Ethics for Simulationists, M&S Body of Knowledge, and multilingual M&S dictionaries. He is the founding director of the M&SNet of SCS. He received "Information Age Award" from the Turkish Ministry of Culture, Distinguished Service Award from SCS and plaques and certificates of appreciation from organizations including ACM, AECL, AFCEA, and NATO.