A MIXED REALITY FRAMEWORK FOR VISUALIZATION AND EXECUTION OF DEVS-BASED SIMULATION MODELS

Arnis Lektauers
Riga Technical University
1 Kalku street, Riga, LV-1658, Latvia
E-mail: arnis@itl.rtu.lv

KEYWORDS

ABSTRACT

A mixed reality (MR) represents an environment composed both by real and virtual objects. MR applications are more and more used in different scientific and practical areas, for instance, in medicine, architecture, entertainment, manufacturing, etc. This paper describes development of mixed reality framework for visualization and execution of discrete event simulation models based on the Discrete Event System Specification (DEVS) formalism. The mixed reality framework provides a 3D augmented/virtual environment representing and executing simulation models that encapsulate the static and dynamic behavior of the simulated system. Basing on the DEVS formalism, there is also presented a new methodological concept, called Visual-DEVS (VDEVS).

INTRODUCTION

Simulation is an important tool in the analysis and design of complex systems. Various types of simulation techniques and tools have been developed, in order to deal with complex systems. In order to help better to understand the behavior of complex systems, visualization tools have a crucial role that facilitates problem solving and decision-making. Combining real and virtual visualization objects in a mixed reality (MR) environment allows model users better to interact with the objects in the simulation environment. Today there exist several works, dealing with support of virtual, augmented and mixed reality for simulation and modeling (Fischer et al. 2003; Rehn et al. 2004).

Formalism, called Discrete Event Specification (DEVS) (Zeigler 2003) is gaining popularity in recent years, enabling to construct discrete-event hierarchical modular simulation models. In this work there is introduced a visual extension for DEVS that enables 2D and 3D mixed reality simulation visualization. 3D mixed reality simulation visualization framework is developed basing on the presented VDEVS formalism.

Mixed Reality

The concept *Mixed Reality* is based on the classification presented by (Milgram and Kishino 1994) and (Ohta and Tamura 1999). Mixed reality refers to the combination of computer graphics and real-world objects. This encompasses both augmented reality and augmented virtuality that places computer graphic objects into the real world placing real-world objects into virtual environments.

A wide range of mixed reality applications have been proposed, ranging from industrial to medical and entertainment applications. This framework is primarily concerned with one particular type of application that is known as monitor-based (non-immersive) or "window-on-the-world" (WOW) (Milgram and Kishino 1994) augmented reality. In this paradigm the data is passed from the video camera to the computer, modified and displayed in the monitor in real-time.

DEVS Formalism

Discrete Event System Specification (DEVS) formalism (Zeigler 2003) is a general formalism for modeling discrete event system. A basic DEVS model has such a structure:

$$AM = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

Where

X is the set of input values,

S is a set of states,

Y is the set of output values,

$\delta_{int} : S \rightarrow S$ is the internal transition function,

$\delta_{ext} : Q \times X \rightarrow S$ is the external transition function, where

$Q = \left\{ (s,e) \middle| s \in S, 0 \le e \le ta(s) \right\}$ is the total state set,

e is the time elapsed since last transaction,

$\lambda : S \rightarrow Y$ is the output function,

$ta : S \rightarrow R_{0,\infty}^{+}$ is the time advance function.

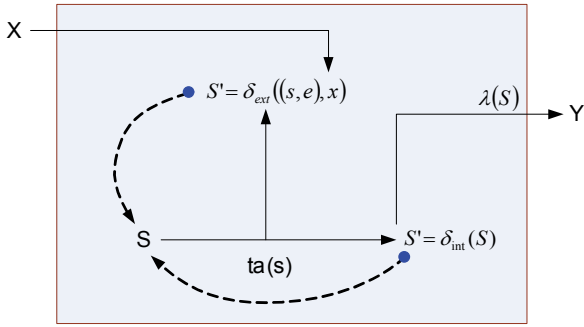The interpretation of these elements is illustrated in Figure 1.

Figure 1: Elements of DEVS Structure (Zeigler 2003)

VDEVS

In order to enable linking the DEVS formalism and visualization concepts into unified visual simulation framework, a Visual-DEVS (VDEVS) methodology is proposed. VDEVS extends the DEVS formalism supporting 2D/3D visualization of simulation models. A VDEVS model can be described as:

$$VM = \langle X, S, Y, \delta_{int}, \delta_{ext}, f, \lambda, ta \rangle$$

$$S = \langle S^{disc}, S^{cont} \rangle \text{ is the sequential state set,}$$

S^{disc} is the discrete state set,

$$S^{cont} = \langle T, G, V \rangle \quad \text{is the geometric}$$

visualization set,

$$T = \begin{bmatrix} R & P \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{are the local}$$

coordinates,

R is the local orientation (3x3 matrix),
P is the local position (3x1 vector),

G is the geometry data,

$$V = \langle Color, Texture \rangle \quad \text{describes the}$$

visualization attributes,

$$f : Q \rightarrow S^{cont} \text{ is the rate of change function.}$$

In order to model the continuous change required by animation purposes, the sequential state set S is decomposed into the discrete state and continuous state (Figure 2).
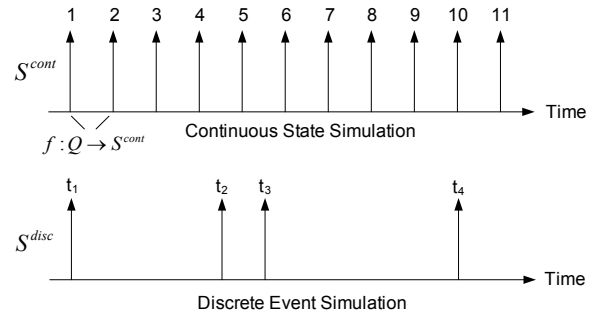

Figure 2: Continuous State and Discrete Event Simulation

SYSTEM ARCHITECTURE

The proposed framework implements the VDEVS formalism and consists of several components supporting simulation design, execution and control, as well as 2D/3D visualization and mixed reality interaction capabilities. The architecture of the given system is made by four base components – modeling module, visualization module, DEVS simulation module and mixed reality module. In Figure 3 is
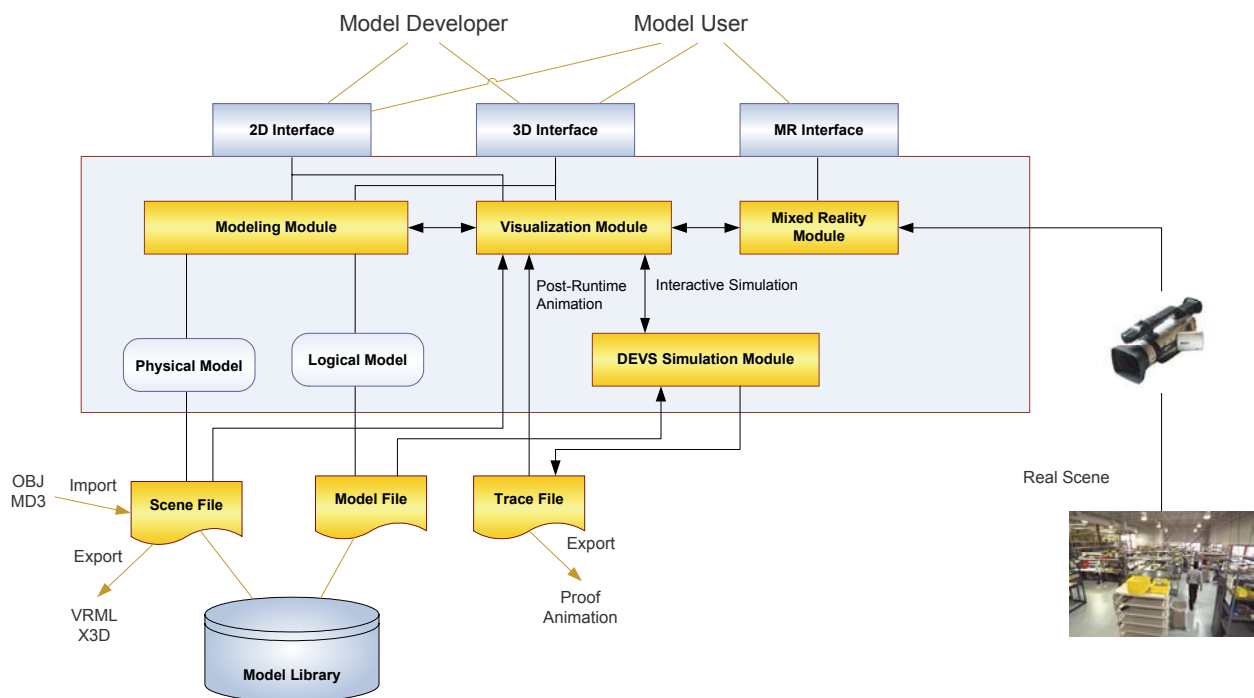

Figure 3: Architecture of Mixed Reality DEVS-based Simulation Environment

illustrated the architectural structure of the proposed MR simulation environment.

Modeling Module

The modeling module is provided for building of simulation models in virtual 3D modeling environment, including static and dynamic images, as well as 3D geometric objects. Simulation models are based on the physical and logical description of the simulated system. The physical model defines the layout and geometry of virtual scene as well as the visual attributes of virtual objects. It is possible to make the physical model with the tools that are built in the virtual modeling environment or to import from external 3D modeling and CAD software, using widely used 3D file formats, such as MD3, OBJ. The created physical model is saved in scene file or exported, for example, in X3D or VRLM formats for use in other 3D software tools.

The logical model contains all the necessary information about the dynamical behavior of the simulated system. For visual process-oriented as well as DEVS based simulation modeling purposes there are suitable 2D/3D simplified hierarchical activity cycle diagrams (SH-ACDS) (Odhabi et al. 1998; Lektauers 2003). To each functional simulation object corresponds a determined 3D graphical object. With the help of 3D computer graphics it is possible to improve the symbolism and esthetical aspects of the model. By graphical linking of the chosen model objects, there is created an simulation model diagram (Figure 4) that defines the model element couplings and execution logic. Every model element has certain subordinated metadata that contains run time parameters for the given element, for example, processing duration, generation frequency etc.
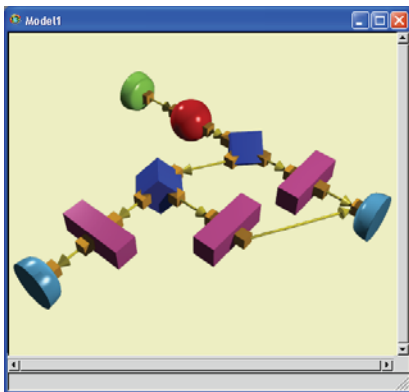

Figure 4: An Example of 3D SH-ACDS Simulation Model Diagram

DEVS Simulation Module

DEVS simulation module is the main component of discrete event modeling system that manages the execution time processing of simulation model and the generation and acquisition of simulation results. This simulation module implements the above described VDEVS formalism and allows to combine the processes of simulation and visualization into a unified visual simulation process. Still the functions of simulation modeling and animation are separated, therefore the simulation model can be activated independently from other system modules, but at the same time during its operating it can use visualization model, in order to provide interactive runtime animation. For execution of simulation animation there is necessary a synchronization between simulation model and visualization module. This task is directed and controlled by the given simulation system, implementing VDEVS formalism.

Model file is used as a source of input-information that contains the data about logical activities of simulation model. The results of simulation modeling execution can be saved in Trace file that allows post-runtime activating of animation without need to activate simulation process.

Visualization Engine

Visualization module is used for generating of 3D objects in mixed reality environment (for example, assembly parts, vehicles, user interface elements), executing all the necessary tasks of computer graphics rendering and animation. Visualization model is used in all simulation modeling phases, beginning with building of simulation model, ending with representing and analysis of modeling results. One of this model's tasks is to provide realistic synthetic object illustration, so that this object can be better virtually integrated in the flow of real video frames.

Rendering of real objects in a virtual environment allows getting more united and more realistic view at simulated system activities. Visualization model includes scene graph structure concept that is well known in computer graphics. Scene graph $G = (V, E)$ is oriented graph that contains a set of graphics nodes $u \in V$ and links $e(u, v) \in E$. Scene graph contains information about the parameters and properties of graphical objects – geometry, spatial position and orientation in 3D environment, colors and texture parameters.

For the visualization system there are established high graphical rendering performance requirements because practical 3D graphic scenes contain objects with big amount of polygons that makes the rendering process more difficult. In order to provide acceptable speed of graphic depiction, there can be used level-of-detail (LOD) criteria, allowing reducing the total amount of polygons.

Visual modeling deals with static and dynamic objects (Chawla and Banerjee 2001). Static objects remain in immutable state during all the modeling phases, but dynamic objects can change their spatial or visual state during user interaction and animation execution. These dynamic objects can move and rotate or they can contain movable elements.
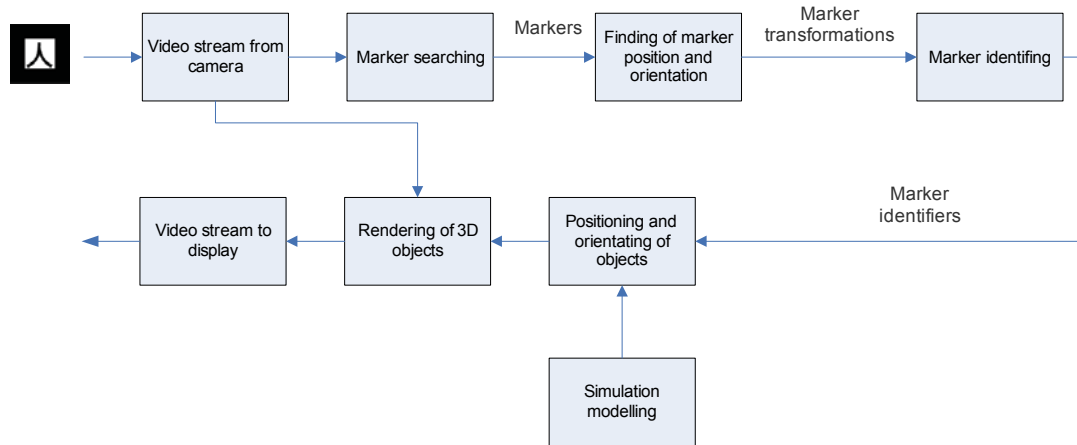
Figure 5: Image Acquisition and Processing using Mixed Reality Module

Mixed Reality Module

The main task of the mixed reality module is to provide the independence of visualization module from different possible real world data acquisition modes. The first working phase of MR module is to obtain the real world images, using the computer-attached video camera (Figure 5). The obtained frames are sent to visualization module where they are depicted in the background of virtual environment. The mixed reality interface module creates a list of detected objects that contains the identifiers as well as data of the spatial position and orientation of the given objects. This set of detected objects is used in visualization module, in order to make depiction of virtual 3D objects. Basing on the detected object identifier there is depicted a correspondent 3D object, using according transformation matrix. So that visual objects are depicted in accordance with markers, then, by manipulating with marked physical objects, the user can simultaneously manipulate with virtual objects

without a necessity for special input devices.

IMPLEMENTATION DETAILS

The mixed reality simulation program prototype SharpSim is developed in the C#/.NET 2.0 programming environment, including the following packages: SharpDEVS, Sharp3D, SharpODE, ShARpToolkit (Figure 6). Although there exist many software libraries for visualization and VR/AR purposes (VRJuggler, OpenSceneGraph etc.), C#/.NET software system was chosen for development of such software prototype in order to check the possibilities of this progressive software development environment. In the future there is a plan to port this software to the open source C# programming environment Mono, allowing to work in Linux operating system environment. Creation of the 3D computer graphics objects is achieved with the Sharp3D package, which is built atop the industry standard OpenGL graphics library. Video
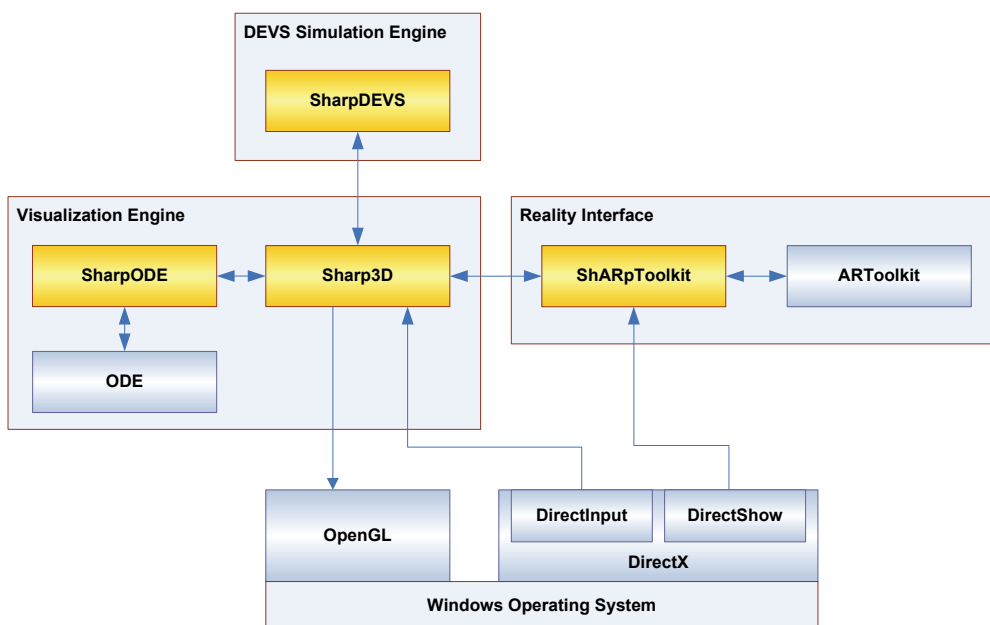

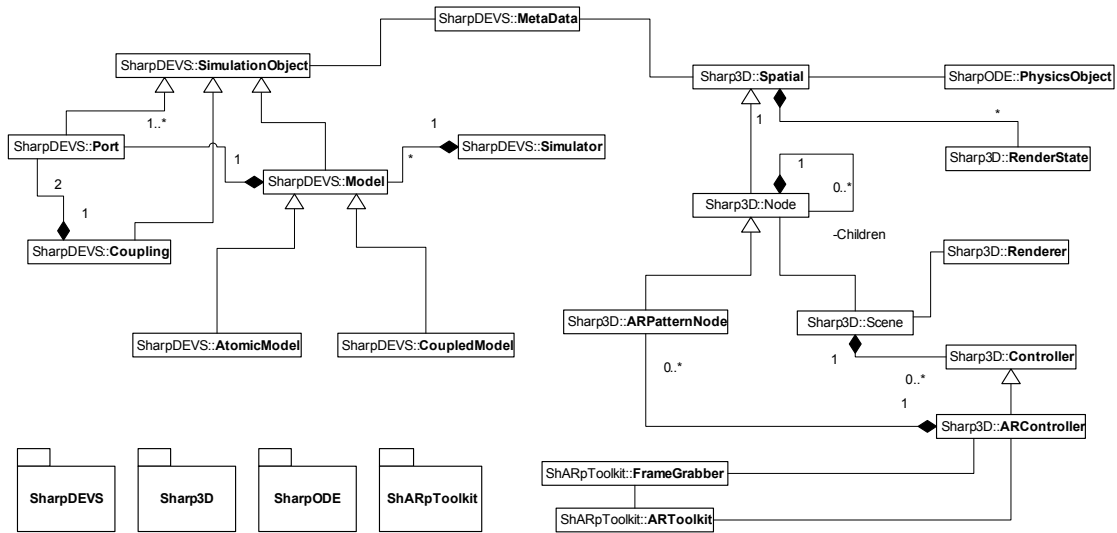Figure 6: Architecture of SharpSim Simulation Environment

Figure 7: Generalized SharpSim Class Structure Diagram

data processing of captured frames is based on Microsoft DirectX 9.0 framework by using DirectShow capabilities. DirectInput system is used to react on the changes in the input of the mouse and the keyboard. In Figure 7 there is shown the general structure of SharpSim classes.

The given software prototype uses augmented reality library ARToolkit (Billinghurst et al. 2001) as a tool for real world object detection, using fiducial markers that are placed near the detectable objects. The data from video camera is passed to the computer, modified and displayed in the monitor in real-time by visual combining of captured video frames and 3D graphical objects. The obtained information about the detected objects and their position in 3D space is then used by scene graph module Sharp3D. ShARrpToolkit package serves as ARToolkit interface for Sharp3D visualization module that by using of C#/.NET resources allows implementing of ARToolkit functions and related data exchange. In ShARrpToolkit module there is provided also video frame obtaining in real-time from the connected video camera, using Microsoft DirectShow technology. The obtained video frames are sent to ARToolkit library for marker detection and to Sharp3D, where by combining with virtual objects they are depicted as mixed reality scenes.

An open source Open Dynamics Engine (ODE) is used for simulating of physical object dynamics. SharpODE module serves as integration layer between ODE and Sharp3D by wrapping in C# methods the ODE function calls.

PRACTICAL EXAMPLE

For testing and verifying purposes of the proposed framework there is chosen a practical warehouse storage system, where the acquired video frames are used as background for simulation of warehouse working processes. But the fiducial markers that are placed in the warehouse storage room allow mixing the given real images with virtual graphical objects.

In the Figure 8 an example model of warehouse storage system is shown. It contains source, queue, sink and two activity elements. The source element Source1 generates flow of incoming goods. Received goods are processed by queue, resource and operator elements.
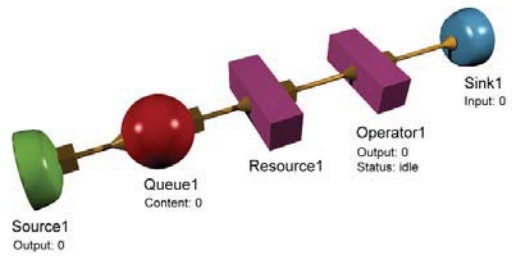

Figure 8: Simulation Model Diagram of Warehouse Storage System

In Figure 9 is shown a captured image of warehouse storage room containing two fiducial markers.

Figure 9: Captured Video Image of Warehouse Storage Room containing Fiducial Markers

In Figure 10 is illustrated a combined image, where virtual 3D objects are merged in to the real video captured image.

Figure 10: Mixed Reality Scene of Warehouse Storage Room

In Figure 11 is shown another image of the mixed reality scene representing working operations in warehouse storage room.

Figure 11: Example of the Combined Image with Virtual 3D Objects

CONCLUSIONS AND FUTURE RESEARCH

Mixed reality is a popular topic for researchers, a lot of applications and hardware for Mixed Reality can be found. But there are not much studies of AR/MR usefulness in simulation modeling. This article offers a framework for developing of DEVS-based discrete event simulation models in virtual 3D environment allowing the user to use a mixed reality interface to build complex simulation models.

The log-term vision of this work is to develop a full-scale mixed reality simulation environment for discrete event simulation of logistics systems. Potential benefits of achieving this vision would include understanding and applying the relationship between simulation effectiveness, visualization quality and user interaction.

REFERENCES

Billinghurst, M.; K. Hirkazu; I. Poupyrev. 2001. "The MagicBook: Moving Seamlessly between Reality and Virtuality". In *IEEE Computer Graphics and Applications*, (May-June). 2-4.

Chawla, R.; A. Banerjee. 2001. "A Virtual Environment for Simulating Manufacturing Operations in 3D". In *Proceedings of the 33th conference on Winter Simulation* (Arlington, VA, Dec.9-12). 991-997.

Fischer, M.; M. Grafe; C. Matysczok; B. Mueck; and M. Schoo. 2003. "Virtual & Augmented Reality Support for Discrete Manufacturing System Simulation". In *Proceedings of the International Working Conference of the IFIP WG 5.7* (Karlsruhe, Germany, Oct.5-9). 170-178.

Lektauers, A. 2003. "A Virtual 3D Environment for Logistics Modelling". *In Proceedings of the International Workshop on Harbour, Maritime and Multimodal Logistics Modelling & Simulation* (Riga, Latvia, Sept 18-20), 242-248.

Milgram P. and F. Kishino. 1994. "A taxonomy of mixed reality visual displays". *IEICE Transactions on Information Systems,* No.12 (Dec), 95, 96.

Odhabi, H.I.; R.J. Paul; R.D. Macredie. 1998. „Developing a Graphical User Interface for Discrete Event Simulation". In *Proceedings of the 30th conference on Winter Simulation* (Washington, DC, Dec.13-16). 429-436

Ohta Y. and H. Tamura 1999. *Mixed Reality – Merging Real and virtual Worlds*. Springer Verlag, Berlin, 95, 96.

Rehn, G.D.; M. Lemessi; J.M. Vance; D.V. Dorozhkin. 2004. „Integrating Operations Simulation Results with an Immersive Virtual Reality Environment". In *Proceedings of the 2004 Winter Simulation Conference* (Orlando, FL, Dec.4-7).

Zeigler, B.P. 2003. „DEVS Today: Recent Advances in Discrete Event-Based Information Technology". In *Proceedings of the 11th IEEE/ACM International Symposium On Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (Orlando, FL, Oct.12-15). IEEE, Orlando, 148-161.

AUTHOR BIOGRAPHY

ARNIS LEKTAUERS received the M.Sc. in Computational Visualistics from Otto-von-Guericke University Magdeburg, Germany (1998). In 1999 he received a M.Sc. in Information Technology from the Department of Modeling and Simulation, Riga Technical University. He is a PhD student and part-time scientific assistant at the Department of Modeling and Simulation of the Riga Technical University. His e-mail address is : arnis@itl.rtu.lv