UDC 519.876.5=111

V.V. Okolnishnikov, S.V. Rudometov

A SYSTEM FOR COMPUTER SIMULATION OF TECHNOLOGICAL PROCESSES

A new simulation system for computer simulation of technological processes is described. The paper contains a brief description of the simulation system and its characteristics, in comparison to analogous commonly used systems. Also there is a description of existing and perspective applications for this simulation system that proves its usability and reliability.

SIMULATION; TECHNOLOGICAL SYSTEMS; AUTOMATED PROCESS CONTROL SYSTEMS.

В.В. Окольнишников, С.В. Рудометов

СИСТЕМА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

Описана новая система имитационного моделирования технологических процессов. Приведено описание системы, ее характеристики, сравнение с существующими аналогичными системами моделирования. Также описаны существующие приложения, подтверждающие простоту использования и надежность этой системы моделирования.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ; ТЕХНОЛОГИЧЕСКИЕ СИСТЕМЫ; АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ.

Design Technological Institute of Digital Techniques, Siberian Branch of the Russian Academy of Sciences (DTIDT) produces a number of hardware and software for different industries, like coal and oil mining. These products are Automated Process Control Systems (APCS) including both Programmable Logic Controllers (PLC) and Process Control Software (PCS). PLC are mostly used for in-place control of Technological Objects (TO) like conveyors in coal mining, whereas PCS is used to control the whole enterprise, including, but not limited by, an energy consumption control, personnel control, etc.

The nature of TO under control implies they are extremely dangerous in running. It means, that APCS must be very reliable, and do not allow potentially dangerous situations to appear. On the other hand, these systems must be «trained» to react adequately in the case of any emergency. To achieve this, computer-based simulation was introduced. The primary goal of such simulation was to check that the PLC and PCS cannot be a source of dangerous situations, and these systems are reliable and safe. The other goal is to show, how the control programs will work in non-standard situations. And finally, the simulation can be used to train personnel that will use new APCS.

To achieve these goals the simulation system MTSS (Manufacturing and Transportation Simulation System) was created. There are also other applications of this simulation system which are not limited by applications inside the DTIDT. These applications are the simulation for an automated warehouse [1] and the simulation model for a complex computing network.

Requirements for MTSS

Non-formal requirements. MTSS supposes that creation of simulation model will be done by subject matter experts (SME), but not by specialist in simulation. Simulation models in MTSS are created and analyzed visually, without application of the simulation theory. Later these models can be used by their authors in their work, without any interference of the specialist in simulation.

Non-formal requirements are:

• to create the simulation models by subject matter experts who are not specialists in simulation;

• to create the simulation models of complex technological systems combining simulation models of technological objects in these systems;

• to hide any aspects of simulation from a final user (only the possibility to tune parameters must be presented; these parameters must be clearly understandable for the final user (SME));

• to create the simulation models fast, allowing users to focus on the problem solving but not on development of simulation itself (and then on a simulation model);

• to control simulation run visually, with ability to pause simulation at any time and to examine or change simulation parameters;

• to examine statistical parameters at any time during simulation, in a form that is clearly understandable for the final user.

Formal requirements for MTSS. These requirements are given below in a form that can be applied to technological systems:

• visual interactive interface for simulation creation and execution;

• usage of graphical tools for model creation;

• support for fast model creation;

• simulation model is created from existing, ready-to-use simulation models;

• simulation model creation by SME;

• simulation model and simulation system must be able to be connected to any external systems;

• simulation model works in two-dimension (2D) and possibly in three-dimension (3D) views;

• simulation results are presented as complete analysis, without necessity of any additional analysis;

• any statistical data can be exported and analyzed externally.

The system MTSS was realized in accordance with all of these requirements.

Architecture of MTSS

Elemental models and disposition. Technological systems (TS) consist of technological objects (TO) which can be defined by their type. TS have input products; these products are processed, and then leave TS. Each TO in TS can be controlled by a supervising (control) program which is usually a part of modern TS.

Elemental model (EM) is a simulation model of a TO type. Each item of EM in the model presents an item of TO in a real system.

Each TO in TS interacts with some other TO in this system. It means that in simulation there must be possibility for EM items to interact. This interaction will simulate the correspondent interaction in real TS. Such connection can be presented by «port ontology» [2].

Port ontology in its original form is a way to organize communications between EM items

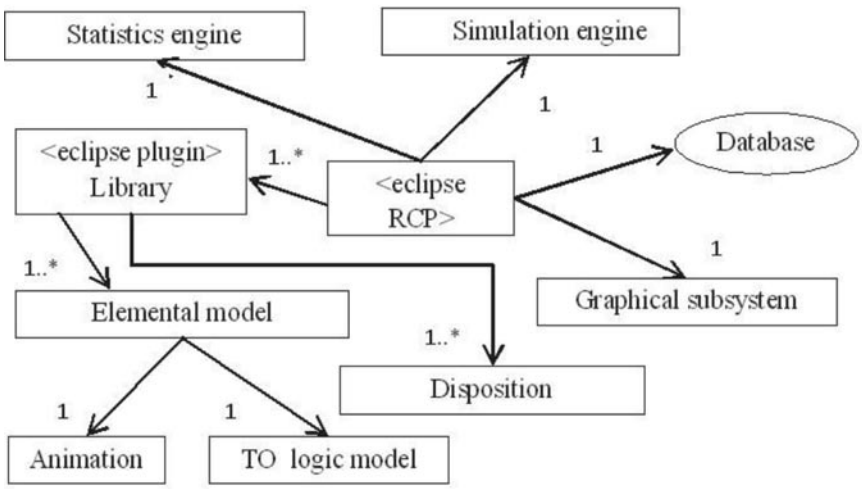

Fig. 1. Architecture of MTSS

encapsulating different approaches to build the simulation. Later the items of the simulation model that are created with port ontology can be connected to each other. MTSS uses this feature for possible EM items to be connected visually. Such connection allows different EM items to communicate (to interact) to each other in the simulation model.

Important part of EM is its visual image. It is used to identify EM item in simulation model, both when model is developed and run.

Control programs in original TS must also have its presentation in the simulation model. MTSS allows to create such models as a disposition level. It means, the MTSS has two-tier architecture: the lower tier is EM items and the upper tier is disposition.

Structure. Fig. 1 is a graphical presentation of architecture of MTSS. The figure uses in UML notation. Next major components of MTSS are presented on this figure.

Simulation engine is used for encapsulation of the simulation aspects. Different ways can be used to create this engine – classical or distributed simulation, etc. All aspects of the computer-based simulation must be encapsulated in this component.

Graphical subsystem is designed for visual presentation of the simulation model, during its creation process and running. MTSS implies that the simulation model will be created and run in 2D. The 3D can be primarily used (or not used) for the goals of presentation.

Statistics engine is used for gathering, storing and processing of any statistics generated during the simulation.

Database is a way to connect to external systems, by data transfer between MTSS and external systems.

Library is a way to unite a number of EMs and dispositions. Later MTSS can be exported as a software product that contains one or many similar libraries. This allows to produce any problem-oriented software products.

Animation and TO logic model are the major parts of any EM. Animation is used to show EM states during simulation run, and also in visual model creation. TO logic model must be implemented to simulate behaviour of original TO.

MTSS was created by using Eclipse platform and Java programming language.

Perfomance of MTSS

One of the main possibilities of MTSS is to

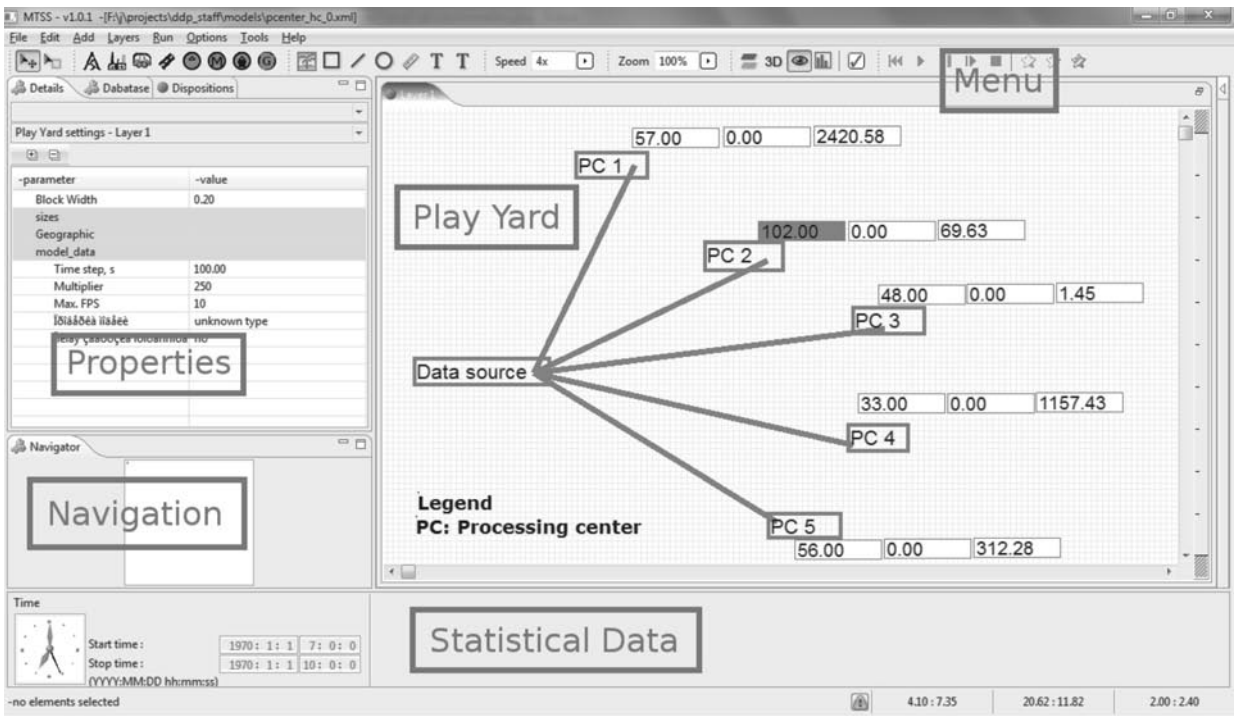

Fig. 2. User interface of MTSS

create the simulation models with SME fast. MTSS defines a structure of EMs that developers of EMs must follow. MTSS can be connected to external systems, to allow simulation to operate with data from real systems or create data for such systems.

The approach of creation of simulation models in MTSS allows users to create detailed and complex models. This fact, and possibility to connect simulation models to external systems, possibly, will allow to use MTSS even as a subsystem for APCS.

Simulation engine in MTSS is built as a modification of Mera system [3]. According to architecture of a simulation system, Simulation Engine is a module, and that is why one of the possibilities of MTSS is a usage of different implementations of simulation engines.

User interface of MTSS. Fig. 2 shows the user interface of MTSS. Main window of MTSS consists of following areas:

Play Yard is an area for model creation and running. Users of MTSS should place EM items here and connect them visually to other EM items which are already on Play Yard.

Properties are used for changing of the model parameters. Also, if EM item is selected in Play Yard, the properties for this item can be changed.

Navigation is an area for fast navigation in a model. This small view contains all its elements in it (while Play Yard can contain just a part of EM items in a model).

Statistical data area is used for output of common statistical parameters that depend on libraries used.

Menu is an application menu and a toolbox. This area allows selecting EM to add it to the model.

Creation of simulation model. Users build simulation models from ready-to-use EMs connecting the EM items visually. This allows the user to speed up models creation, to make them obvious for the users who are not familiar with aspects of computer-based simulation. For this purpose the port ontology is used.

In MTSS, ports have graphical representation and physical coordinates. Each EM can have more than one port. This allows the users to connect visually different ports to different EM items in the model, and to check visually correctness of the model. Ports in MTSS are connected to each other by dragging one port to another. Ports are counter connected if their coordinates overlay and EM implementation will allow to do it (by its internal logic). After connecting the ports, the information can be used during simulation run, in order to organize communication between different EM items.

Play Yard has its fixed physical size that can be seen in scaled Play Yard view. Navigation area can be used to move between parts of Play Yard. This allows user to build simulation models of TS, for example, buildings or big areas. MTSS also supports setting of geographical coordinates for Play Yard.

Model can be either stopped before end time of simulation, paused or stepped. The speed of model execution can be changed at any time of model running. Such handling gives the user, who carries out the simulation, a full control over the simulation process, the ability to examine the situations occur in the simulation model in details.

Animation in MTSS is asynchronous (independent on the simulation speed). Animation can be completely turned off to speedup model execution. It means that graphical engine is independent on simulation speed. If animation requires many resources, it can be delayed without need of whole simulation to be turned off. Animation can be turned off at any time during the simulation run.

The speed of the simulation can be varied during simulation run.

Statistics gathering and presentation occur during simulation run, and do not require additional calculations. During the simulation run the user will receive important information (speed, energy consumption, product flow) and some calculated values (e. g. energy consumption during the time period, system performance etc.). Using this data the user can decide to stop simulation before the actual simulation end time. Commonly-used tools are also presented in the system (charts, data export tool, and etc.).

Control of simulation run is mainly visual using the navigation tools and tools for simulation time management.

SME will not operate with any aspects of computer-based simulation theory (except «model time»). MTSS does not provide any possibilities for the final user that leads to aspects like queries, model decomposition etc.

In MTSS, SME will build simulation model fast, from ready-to-use EMs. Simulation experiments can also be carried fast, primarily visually and simulation results will be also achieved fast. There are no other specialists who can be involved in such investigation, there is no need to search for any specific information. Everything what SME needs, MTSS will provide. It is the major distinction between MTSS and other known simulation systems.

Creation of libraries with EM. MTSS allows programmers (specialists in simulation) to unite EMs and disposition programs in libraries. Programmers are free to use a common scheme for creation of simulation model in MTSS. The main distinction is that simulation model will be created not for the whole TS, but just for one TO type.

The programmer must keep in mind that the exact simulation model, the exact TS under simulation, and the exact set of simulation experiments are unknown when library of EMs is created. That is why the programmer must imply algorithms in models of logic for Ems more detailed, predict more use-cases for EM than that was originally specified. This can make creation of EMs complex. It means, that detailed investigation of TO and TS simulated must be done. But such specification turns the original task to wider one, and allows the final user to use EMs in more tasks than it was originally requested.

There are no special graphical tools in MTSS that may «help» to imply the logic of TO models as such implementation is complex. It means that advantages of graphical presentation of algorithm will be lost. According to different estimates up to 80 % of code will be written «by hands», even if special graphical tools exist in a simulation system.

Commands and states. MTSS allows (but does not insist on) the following approach to decompose the logic of EM. This approach will allow the programmer to simplify and formalize the model of the logic in EM. This logical part must be presented as a set of *states*; the transition between them must be done by *commands*.

State is an abstraction that allows the programmer to denote the current activity of EM. It is considered in MTSS that each EM item in each model time is in one state.

Command of EM is a rule to change the state of EM. The command includes the condition to start, initial state, end state, and algorithm for translation between states of EM. The command can be sent by EM itself, by the user, by another EM item, or a disposition. Each command will be added to the queue of commands existing in any EM item.

MTSS has an interface with Databases (Relational Database Management Systems – RDBMS). This allows creating simple and uniform way to connect to the external system. Java allows to connect to any RDBMS using its unified interface.

The usage of such interface can be not enough. That is why the programmers can use their own styles for accessing data from external systems. The only limitation is that this connection must be created in the disposition.

Comparison with Existing Simulation Systems

Computer-based simulation was invented as a way to solve problems that was unable to solve analytically (mostly mathematically). The original problem must be decomposed to the form that allows it to be solved by some existing approach (queuing theory, Petri nets, etc.).

Systems for computer simulation can be divided into classes:

simulation languages and libraries;

systems for visual composition of simulation models.

This classification is not exact: simulation languages can have a visual graphical environment, or systems for visual composition will translate simulation models to some existing simulation language.

But most important is that all of these systems can be used just by specialists in simulation. Only these specialists were able to made a decomposition of the original problem, create simulation experiments, validate and verify models, and most important thing is to made a backward decomposition of the simulation results to the final user (SME).

Over the past ten years a number of simulation systems proved that SME can be used [4, 5]. But these systems based on existing are built, previous versions of the same products can be appropriate only for specialists in simulation. As a rule, the interface of such system is untouched, only some subject-oriented components are added. And these systems are still based on decomposition of original TS, like in a classic simulation. But today computers are not limited in memory and processor resources, they are very good with user interfaces, and they allow to create simulation experiments with very detailed simulation models, with detailed and informative graphical representation.

Another disadvantage of such systems is that they try to be universal. This leads to overloaded interfaces and knowledge area of simulation system itself.

SME does not need anything of this. They just need a tool with elements they are familiar with, that will allow to build simulation models immediately, without any preparation like decomposition or data preparation. They even do not need to know if this is a simulation model. Such systems (and our experience shows it) SME are easier to understand and they are more preferable. MTSS was created to satisfy these requirements exactly.

Applications with MTSS

There is a number of existing applications built using MTSS [6–8].

One of the latest and perspective developments is the simulation of a network of calculation centres processing data that comes from different sources. This data can arrive from meteorological or seismic stations distributed over a large territory or even in space (satellites). Each node in this network has a complex architecture including data storage and queries, a control part and a computational part. The goal of simulation is:

- to predict loading of each node in such network: network topology, node performance and other parameters (some part of them might not be specified yet but will be added later) can be varied;
- to propose load balancing during planned upgrades of a set of nodes;
- to propose load balancing in a case of a number of nodes malfunction;
- to propose network behaviour when input data flow changes its behaviour.

The input from each data source is supposed to be Gamma-distributed. Original network of calculation centres is decomposed to a number of EMs:

1. EM for calculation centre. This complex EM contains parts simulating data storage, data queries, and data processing.

2. EM for wire between two calculation centres or data sources; such wire can be «broken» and will also simulate the delay when transmitting huge amount of data from the source to the destination.

3. EM for data source. This EM can be adjusted for various types of data, which will (depending on the data type) have a length, speed of transmission and frequency of generation. Frequency can be specified also by selecting the distribution and time interval.

This library of EMs is tested in order to be adequate in a certain ideal case that has analytical solution. Test results will be upper-approach analytical results for average and maximum value of node loading, and covariant matrix.

MTSS is the simulation environment that makes it possible to create very detailed and reliable simulation models by subject matter experts who are not familiar with computer-based simulation.

MTSS started as a simulation environment for technological systems. But its applications are not limited only by TS itself – using the same approach, the specialist in simulation can build libraries of EMs for wider range of systems. The example is the library for computer simulation of complex computational network.

This work is supported by the Russian Foundation for Basic Research (Project 13-07-98023 p_siberia_a).

REFERENCES / СПИСОК ЛИТЕРАТУРЫ

1. WirthLogistic GMBH [Online]. Available: www.wirthsim.com (Accessed 19.10. 2011).

2. **Liang V.C., Paredis C.J.J.** Foundations of multiparadigm modelling and simulation: a port ontology for automated model composition. *Proc. of the 2003 Winter Simulation Conference*, pp. 613–622.

3. **Rudometov S.V., Okolnishnikov V.V.** Development of Distributed Simulation System. *Proc. of the 7th Internat. Conference Parallel Computing Technologies.* Nizhni Novgorod, Russia, 2003, pp. 524–527.

4. **Harrell C., Ghosh B.K., Bowden R.** Simulation using ProModel. McGraw Hill, 2004, 733 p.

5. WITNESS simulation [Online]. Available: http://www.lanner.com (Accessed 12.02.2011).

6. **Okolnishnikov V.V., Rudometov S.V., Zhuravlev S.S.** Monitoring System Development Using Simulation. *Proc. of 2010 IEEE Region 8 Internat. Conference on Computational Technologies in Electrical and Electronics Engineering.* Irkutsk, Russia, 2010, pp. 736–739.

7. **Okolnishnikov V.V., Rudometov S.V., Zhuravlev S.S.** Simulation environment for industrial and transportation systems. *Proc. of Internat. Conference on Modelling and Simulation.* Prague, Czech Republic, 2010, pp. 337–340.

8. **Rudometov S.V.** Workflow for Rapid Simulation of Complex Distribution Centers. *Proc. of Internat. Conference on Modelling and Simulation.* Prague, Czech Republic, 2010, pp. 374–377.

OKOLNISHNIKOV, Victor V. *Design Technological Institute of Digital Techniques, Siberian Branch of the Russian Academy of Sciences.*
630090, Rzhanova Str. 6, St. Novosibirsk, Russia.
E-mail: okoln@mail.ru

ОКОЛЬНИШНИКОВ Виктор Васильевич – *ведущий научный сотрудник Конструкторско-технологического института вычислительной техники СО РАН, доктор технических наук.*
630090, Россия, г. Новосибирск, ул. Академика Ржанова, д. 6.
E-mail: okoln@mail.ru

RUDOMETOV, Sergey V. *Design Technological Institute of Digital Techniques, Siberian Branch of the Russian Academy of Sciences.*
630090, Rzhanova Str. 6, St. Novosibirsk, Russia.
E-mail: rsw@academ.org

РУДОМЕТОВ Сергей Валерьевич – *научный сотрудник Конструкторско-технологического института вычислительной техники СО РАН, кандидат технических наук.*
630090, Россия, г. Новосибирск, ул. Академика Ржанова, д. 6.
E-mail: rsw@academ.org