Simulation As a Tool for Debugging and Testing of Control Programs for Process Control Systems in Coal Mining*

Victor Okolnishnikov, Sergey Rudometov, and Sergey Zhuravlev
Design Technological Institute of Digital Techniques
Novosibirsk, Russia
okoln@mail.ru

***Abstract*—**The article describes the use of the model integrated with an actual process control system for debugging and testing of control programs for this process control system. Three ways how to use the integrated model are considered. Control programs of the process control pumping system for a coal mine were debugged and tested using hardware-software test bench which includes the integrated model. The integrated model has been developed using a new visual interactive discrete simulation system specialized in simulation of technological processes. Some models of coal mining processes were developed. These models can be used for development of new mining robotized techniques.

***Keywords*—**visual interactive simulation; process control system; coal mining

I. Introduction

Technological Institute of Digital Techniques of Siberian Branch of the Russian Academy of Sciences in Novosibirsk develops process control systems for coal mining. Arising problems are the following:

- The complete testing of control system using the programmer's tools is almost impossible because of inability to connect to real equipment.
- There is no way to make the complete testing of control system on-site because of inability to reproduce alarm situations or emergencies.
- Startup and live testing time of control system on-site is limited.

The most suitable way to solve these problems is simulation. A means for solving these problems is a model integrated with an actual process control system. The model can run as a part of the actual process control system. On the other hand, the model can use software components of the actual process control system, for example, control programs, the operator workstation, and others. The model can emulate processing equipment of the actual process control system.

There are many examples of the use of emulation models for testing automated production lines or automated material handling systems [1].

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

This paper describes the use of integrated models for developing of industrial process control systems for underground coal mines in Kuznetsk Coal Basin (Russia, Western Siberia).

These models were built with the help of the new visual interactive discrete simulation system intended for the development and execution of models of technological processes.

Section 2 gives the review of architecture and capabilities of this simulation system.

Section 3 presents the use the model, integrated with the actual control system in the framework of hardware-software test bench.

II. The Overview of Architecture and Capabilities of the Simulation System

At present, simulation tools are required for rapid development of simulation and emulation models for various industrial applications [2]. Such models can be used as parts of the actual process control system for developing, testing, optimization, and operator training [3, 4, 5]. They can be also used in marketing to present some industry solutions to the customers.

One of the requirements is to reduce or to exclude the participation of specialists in industrial simulation in the process of the simulation model contraction. It is required to ensure transparent access to simulation system for the specialists from application area having minimal knowledge in information technologies [6].

These requirements were taken into account while designing a new visual interactive simulation system of manufacturing processes MTSS (Manufacturing and Transportation Simulation System) [7, 8, 9]. MTSS is a set of program interfaces for building specialized libraries of *elementary models* and then for construction of complex models using elementary ones.

The elementary model is a simulation model of an object of equipment in a technological system. It consists of the following parts:

- Two-dimensional and three-dimensional graphic images.
- Input and output parameters.
- Functionality algorithm describing dependence between parameters.
- States that the elementary model can reach during the simulation process.
- Control commands defining switching process between elementary models states.

The process of development of the elementary model consists of creation a conceptual model of an equipment unit and its translation into Java in accordance with the structure of the elementary model in MTSS.

Images of different elementary models can connect to each other, with the help of graphical port mechanism. Such approach allows building simulation models of complex technological systems using elementary models.

The elementary model is a model of an equipment unit and its low-level control for it. The model in MTSS constructed by graphical connection of elementary models images. MTSS is also the system for running of complex models built from elementary models. The running model ensures the advances of the model time and visualization. Statistics are also collected.

Industrial process control systems often have two levels: the low level of equipment and simple control logic and the upper level of complex control of production. Therefore, one of the features of MTSS architecture is a forced split of the logic of the simulation into two parts: low-level logic and an upper level one.

Such division in architecture allows not only correspond to the usual structure of the process control systems but use such models for embedding them into actual process control systems in the following ways: to emulate equipment, to simulate upper level logic, and to send signals to actual systems for the visualization. This division into upper and lower logics allows also organizing a switch between various implementations of the decomposition. It allows coexisting simulation of upper level logic and a proxy that allows communicating with the upper level logic of actual process control systems.

The upper level logic executes coordinated control of all elementary models or of a group of elementary models. The possible difficulty of such approach is that each elementary model must be specified to allow easy creating as many models as possible. This approach requires elementary models simulate rather exactly the units of equipment. In the case of such approach, there is a good chance to have an exact simulation model.

MTSS is effective in solving the task for the rapid building of correct simulation model by engineers in their fields. Usually engineers have no enough experience and qualification to build a simulation model in details, but they are good at connecting correctly elementary models to build the required topology.

The architecture of the system consists of the next main components: a simulator, libraries of elementary models, a graphical engine, visual interactive interface, and so on.

Architecture of MTSS allows different simulators to be used. Currently, simulators can be:

- The simulator that presents each instance of elementary model as a thread process. This appears to be the simplest and universal approach to simulation of technological processes, when total amount of elementary model instances is not so numerous.
- The simulator that applies agent-based approach to the simulation.
- Distributed simulator.

While elementary models are simulation models of technological equipment, there is also a separate layer called “disposition”. In this layer, programs (or their models) that manages technological process as a whole can be implemented. Each instance of an elementary model in the simulation model can send a signal to the disposition layer when it is not possible for the logic of this instance to define its future behavior. In other words, the instances of elementary models can “see” only “round” itself (and this is how the original technological objects behave), while disposition layer has a global picture, and can make a decision based on all current variables of the simulation model.

One of essences of MTSS is in function based on “relaxed” simulation [10]. It is true for all simulators mentioned above. Recognized correctness paradigm of parallel and distributed simulation comes to the following: the sequence of events in the parallel and distributed simulation written on the time axis must exactly match the sequence of events in a sequential simulation. For relaxed distributed simulation it is not necessarily in case when small disturbances of correctness do not affect the final simulation result. There are several researches for such approach to the distributed simulation. The main point in these researches is the possibility of the usage of some third-party observations that will work in pair with the main simulator to increase the precision and correctness of the simulation.

MTSS employs visual based simulation as this third-party observation mentioned above. Bear in mind that, when user of the simulation model analyzes the process of simulation model run and s/he can arrive at the conclusion about correctness of the simulation just looking at the model performance or its statistical data gathering. Such conclusion is usually made far before model run is completed. Then the simulation run itself can be stopped by the user of the simulation model. MTSS allows this. User then can check all the statistical data gathered, correct future simulation experiments and (or) make modifications in the simulation model.

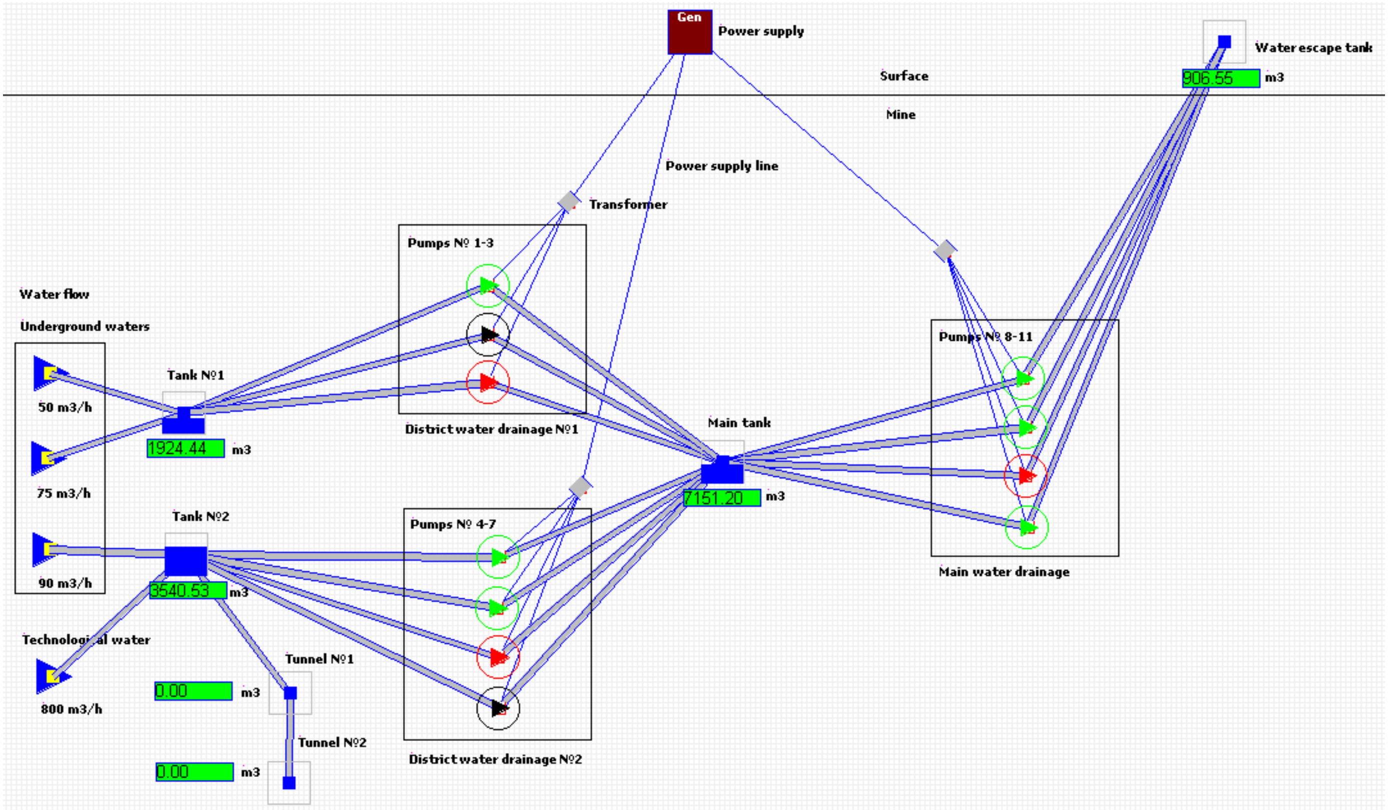

Fig. 1. Fragment of the model of pumping system.

III. Hardware-Software Test Bench

In order to debug and test control programs for actual process control pumping system we use the model, integrated with the actual control system in the framework of hardware-software test bench.

A fragment of the model of pumping system is shown in Fig. 1. This system refers to mine safety systems. It averts mine flood. Pumping system model consists of pumps, tubes, tanks, water flows and requires elements of electricity system. Pumping simulation based on two parts: fluid distribution and energy consumption simulation. The simulation model of this system provides technological and underground water pumping out processes and energy consumption simulation. The main settings are the following: pump efficiency, tube throughput, pump energy consumption, tanks capacity, and so on.

The following problems solved with the help of the model of pumping system:

- Pumps centralized turning on.
- Pump turning on in according with selected control mode.
- Selection of optimized control mode for pumps.

The integrated model is developed and performs in simulation system MTSS. The usual structure of the actual process control system is shown in Fig. 2. The hardware-software test bench is shown in Fig. 3.

While carrying out the task three ways of the hardware-software test bench were used.

A. Autonomous model

This model not restricted by the actual control system. In this approach, the processing equipment of the actual control system replaced by the emulation model of processing equipment. The control programs in this case are different (light, but more often – hard!) versions of control programs for the actual control system. An autonomous model approach usually uses to achieve the following goals:

- Design of a perspective, non-existent system.
- Solution for optimization problems.
- Experiments with the model of the system when the experiments with system simulated are expensive or dangerous.
- Definition of performance boundary conditions of simulated system.
- System improvement.
- Approbation of different ideas in computer-based simulation itself.

There are simulations of three systems of a coal mining in Kuznetsk Coal Basin (Russia, Western Siberia) made using MTSS:

- Model of an underground coal pipeline system.

- Pumping system.
- Model of coal mining process.

There is also an ongoing work on new models of coal mining processes. Some results of this work are presented in Fig. 4.

Coal mining simulation models solve the tasks of prediction and comparison of performance and safety of well-known ways for open or underground mining in flat coal layers, both using existing and (or) perspective winning machines. These models can be used also for development of new mining technics, robotized, without people.

The use of MTSS and simulation models of coal mining has possibilities:

- Development of complex models of robotized mining techniques that includes interoperating simulation models of winning machines and a model of coal layer;
- Visual 2-dimentional and 3-dimentioal presentation of simulation model performance. This includes a possibility for direct management both for separate machines and for scenarios of system functioning as a whole.
- Setup and change parameters of coal layer, to define various emergencies.

These possibilities are achieved due to a model structure. Model consists of exemplars of parameterized elementary models of flat coal layer, common disposition that contains management program for whole simulation system, and other options that MTSS allows.

The special case is a way to define a program for disposition. It is defined as a set of filters, created after semantic parsing of phrases in human language. These phrases uses ontology described for the model.

The main distinction from the existing systems that uses natural language is in the way it is used. Most often systems with natural language control use natural language to provide direct commands to the controlling subject. Possible applications: operative control to robots, with commands in natural language. Our new approach allows describing some relations with ontology objects and their attributes. These relations are then used as a filter, to determine possible future actions of the system under control. Coal mining is a very good application to test such approach, because it is strictly bounded in its ontology, and does not require appending it with any new objects.

Fig. 4 shows a coal mining facility with coal layer accessible from the front, and situated at some small angle to the horizon (15 degrees in this sketch model). On the Fig. 4, there are both 2-dimentional (from top) and 3-dimentional views (as from front) of the facility. There are four winning machines that mine coal in their areas and four transports that move the coal (product) mined to the common unloading point. On a 3-dimentional view, darker areas are the “corridors” winning machines produce in coal layer. When such a corridor touches the “roof” of a coal layer, its angle must change from horizontal to follow this new angle.

The 2-dimentional view on Fig.4 contains the same elements (coal layer, winning machines, transporters, unloading point), but strictly from top. All the elements are rectangles on this view.

B. Data integrated model

The data integrated model is a source of signals for the actual control system instead of the actual processing equipment. The structure of the data integrating model is given in Fig. 3 without the loop of a control programs exchange.

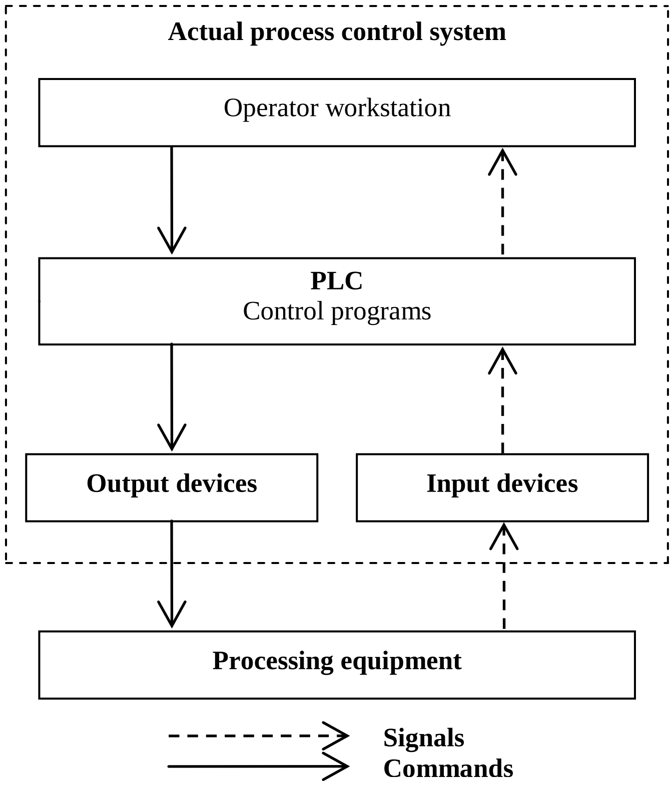

Fig. 2. Structure of usual process control system.

The data model is designed to debug and test all software of the actual control system (upper and lower levels) excluding the control programs. In order to debug the lower level of the actual control system and input and output devices a part of signals generated by the model is transformed to the form of signals from actual sensors of processing equipment. In order to debug the upper level of the actual control system a part of signals generated by the model is transformed to the form of signals transmitted through local area network of the actual control system.

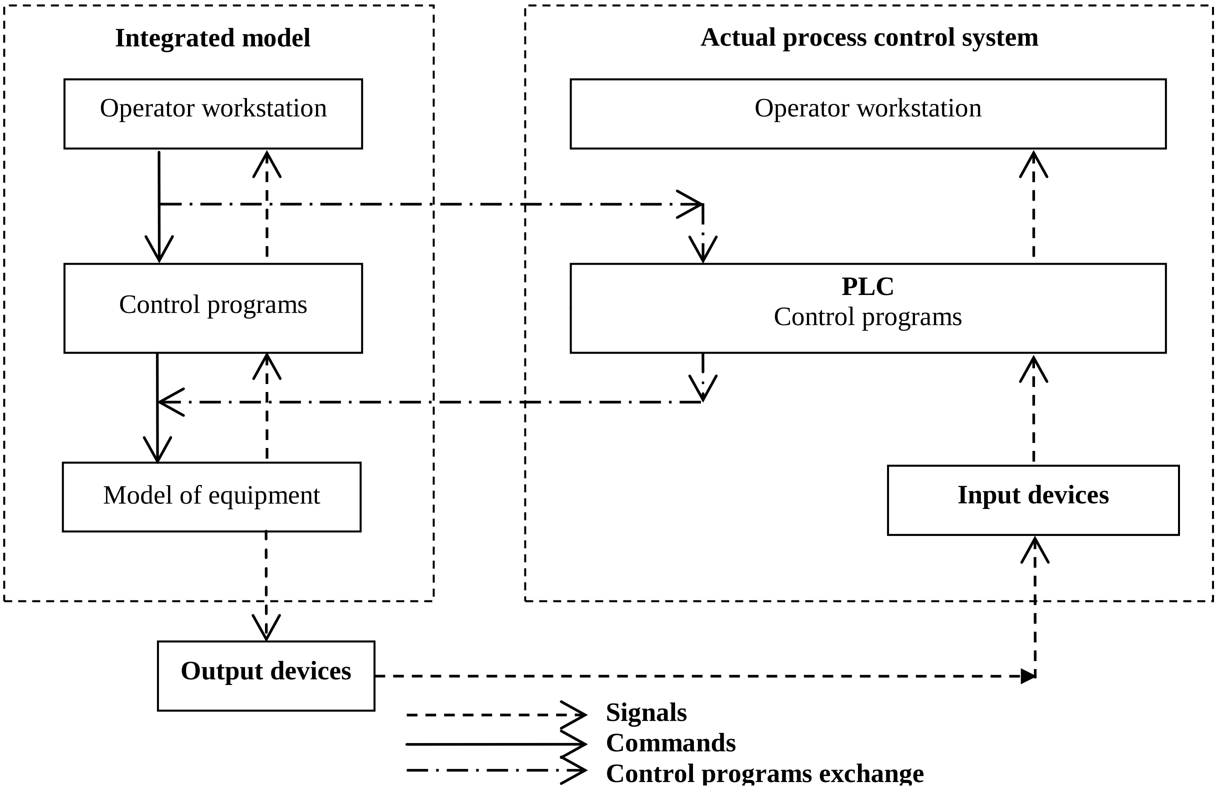

Fig. 3. Model integrated with the actual process control system.

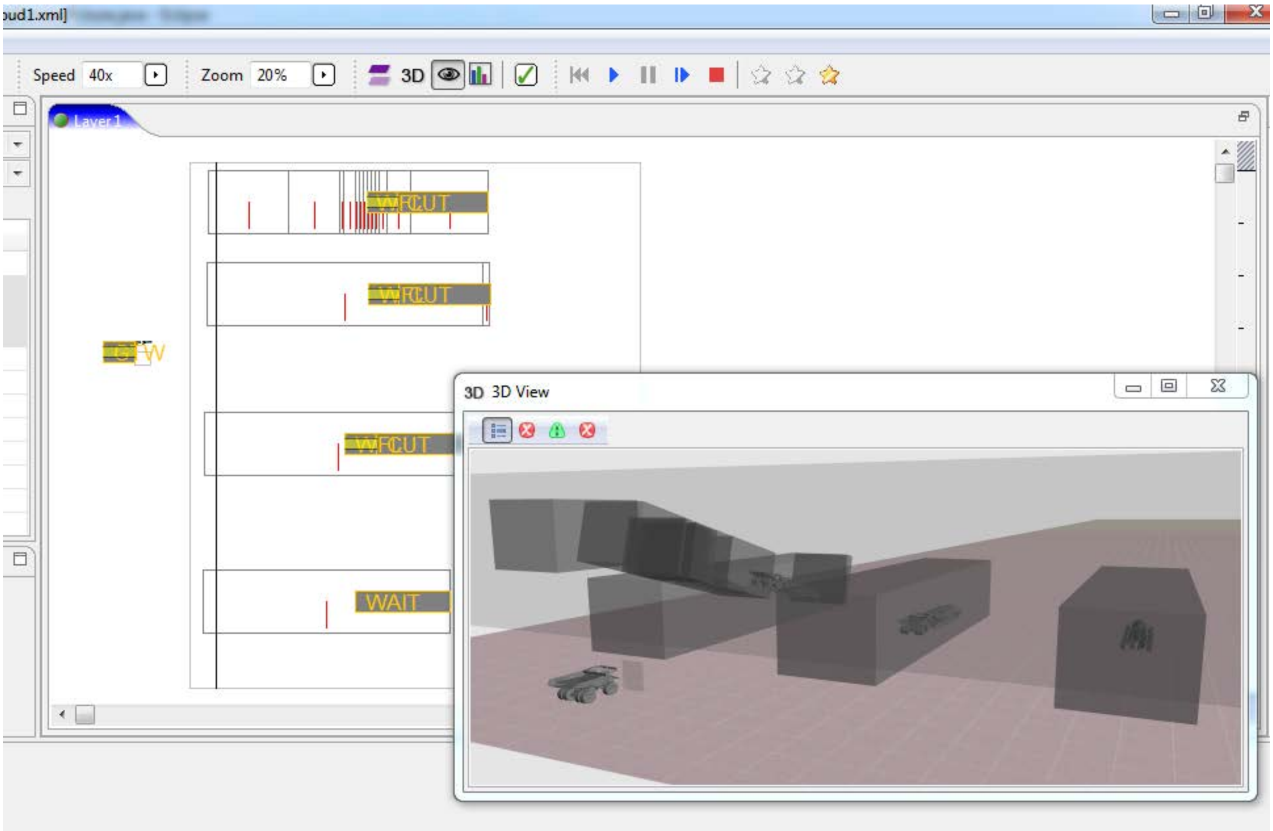

Fig. 4. Simulation of angle-situated coal mining (4 winning machines and 4 self-propelled trolleys).

The data integrated model can be used to achieve the following goals:

- Debugging of software of upper and lower levels of the actual control system.
- Simulation of emergency and accident-related situations.
- Demo for a customer;
- Simulator for training of operators.

C. Controlling integrated model

The controlling integrated model is aimed to debug and test all software of the actual control system (upper and lower levels) including the control programs. The structure of the controlling integrated model is given in Fig. 3. In the controlling integrated model a light version of control programs is substituted while execution for a release version performed in a programmable logic controller of the actual control system. Control programs "don't know" in what environment they perform in an actual system or an integrated model. This way the validation of the model is obtained.

The controlling integrated model can be used to achieve the above mentioned goals for the data integrated model but also for additional ones:

- Maintenance of the actual control system during its life cycle, design, development, debugging, testing, commissioning, operational testing, optimization, evolution.
- Predictable conduct of the actual control system depending on situation and activity (non-activity) of the operator.
- Use of the model of external environment.

IV. Conclusion

Using hardware-software test bench with the integrated model control programs of the process control pumping system for a coal mine in Kuznetsk Coal Basin was debugged and tested.

The usage of the integrated model let us reduce the time and cost of the development of process control pumping system.

Hardware-software test bench can be used not only for simulation of existing coal mining techniques but also for perspective techniques (like robotized semi-independent mining).

References

[1] N. Koflanovich and P. Hartman, "Live Modernizations of Automated Material Handling Systems: Bridging the Gap between Design and Startup Using Emulation," in Proc. 2010 Winter Simulation Conference, Baltimore, 2010, pp. 1716–1726.

[2] D. Carroll, "Rapid-Prototyping Emulation System Co-emulation Modelling Interface for SystemC Real-Time Emulation," in Proc. of the 12th WSEAS International Conference on Systems, Heraklion, Greece, July 22-24, 2008, pp. 691–697.

[3] O. Mere, A. Elias, and G. Marcos, "Data Mining and Simulation Processes as Useful Tools for Industrial Processes," in Proc. of the 5th WSEAS Int. Conf. on Simulation, Moddeling, and Optimization, Corfu, Greece, August 17-19, 2005, pp 243–249.

[4] A. Espinosa-Reza, A. Quintero-Reyes, and R. Garscia-Mendoza, "On-Line Simulator of Electrical Distribution Networks for Decision Support in Distribution Control Centers," in Proc. of the 12th WSEAS International Conference on Automatic Control, Modelling & Simulation, Catania, Italy, May 29-31, 2010, pp 138–143.

[5] F. Rivas-Echeverria, "Plenary Lecture 3: Simulation, Artificial Intelligence and Virtual Systems Applications in Industrial Processes Education," in Proc. of the 9th WSEAS International Conference on System Science, and Simulation in Engineering, Japan October 4-6, 2010, pp. 17–18.

[6] E. Ginters, "Plenary Lecture 4: Simulation Highway – Step by Step to Common Environment," in Proc. of the 11th WSEAS International Conference on Automatic Control, Modelling and Simulation, Istanbul, Turkey, May 30-June 1, 2009, p. 19.

[7] V. Okolnishnikov, "Use of simulation for Development of Process Control System," in Proc. of the 2008 IEEE Region 8 International Conference, Novosibirsk, Russia, 2008, pp. 248–251.

[8] V. Okolnishnikov, S. Rudometov, and S. Zhuravlev, "Simulation environment for industrial and transportation systems," in Proc. of the International Conference on Modelling and Simulation, Prague, Czech Republic, 2010, pp. 161–165.

[9] V. Okolnishnikov, S. Rudometov, and S. Zhuravlev, "Monitoring System Development Using Simulation," in Proc. of the 2010 IEEE Region 8 International Conference, Irkutsk, Russia, July 11-15, 2010, vol. II, pp. 736–739.

[10] Narayanan V. Thondugulam, Dhananjai Madhava Rao, Radharamanan Radhakrishnan, Philip A. Wilsey, "Relaxing causal constraints in PDES," in Proc. of the International Parallel and Distributed Processing Symposium/International Parallel Processing Symposium (IPDPS/IPPS), San Juan: IEEE, 1999, pp. 696–700.