

Development of process control systems with the use of emulation models

Victor Okolnishnikov

Abstract— This paper describes the use of emulation models to provide feedback for industrial process control systems in order to test their logical operation prior to commissioning. The new visual interactive discrete simulation environment intended for the development and execution of simulation and emulation models is presented. Main possibilities that are provided by the environment are: the rapid building of models from libraries elements, 3D representation, and the interaction of models with real control systems. The paper presents three cases of successful development and use of emulation models for process control systems in various industrial fields created with the help of this environment. The use of emulation models reduces the time and the cost of testing and optimization control systems. Emulation models were also used for training of operating personnel.

Keywords—Emulation model, monitoring system, process control system, simulation, simulation environment.

I. INTRODUCTION

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

- 1) The complete testing of control system using the programmer's tools is almost impossible because of inability to connect to real equipment.
- 2) There is no way to make the complete testing of control system on-site because of inability to reproduce alarm situations or emergency situations.
- 3) Startup and live testing time of control system on-site is limited.
- 4) It is difficult to train the control system operators.

The most suitable way to solve these problems is simulation. A means for solving these problems is a model integrated with a real control system. The model can be run as a part of a real control system. On the other hand, the model can include software components of a real control system, for example, control programs, the workstation of an operator, and others. These models are known as emulation models [1].

There are many examples of the use of emulation models for

Revised version received September 8, 2011. This work is supported by the Russian Foundation for Basic Research (Project 10-08-01211-a) and by the Presidium of the Russian Academy of Sciences for Basic Research (Project 14.9).

Victor Okolnishnikov is with Design Technological Institute of Digital Techniques of Siberian Branch of the Russian Academy of Sciences, 630090 Russia (phone: + 7(383)330-92-42; e-mail: okoln@mail.ru).

testing automated production lines or automated material handling systems [2]–[6].

This paper describes the use of emulation models for testing of process control systems of wider applications. These models were built with the help of the new visual interactive discrete simulation environment intended for the development and execution of the simulation models and the emulation models for process control systems. Section 2 gives the review of architecture and capabilities of this simulation environment.

Section 3 presents the use of emulation model for the process control system for North-Muya railway tunnel (Baykal-Amur railway, Eastern Siberia). Section 4 presents the use of emulation model for the monitoring system of the technological infrastructure of oil producing company (Tyumen region, Western Siberia). Section 5 presents the use of emulation model for the process control system of underground coal mining (Kemerovo region, Kuzbass).

II. THE OVERVIEW OF ARCHITECTURE AND CAPABILITIES OF THE SIMULATION ENVIRONMENT

At present, simulation tools are required for rapid development of simulation and emulation models for various industrial applications [7]. Such models can be used as parts of process control system for developing, testing, optimization, and operator training [8]–[10]. 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 environment for the specialists from application area having minimal knowledge in information technologies [11].

These requirements were taken into account while designing a new simulation environment. Process Control System Simulation Environment (PCSSE) is a set of program interfaces for building elementary models and then for construction of complex models using elementary ones. The elementary model is a model of an equipment unit and its low-level control for it. The model in PCSSE is constructed by graphical connection of elementary models images. PCSSE is also an environment for running of complex models built from elementary models. The running model ensures the advances of the model time and visualization. Statistics are also being gathered.

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 PCSSE 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 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 accept signals from actual systems to become a visualizing tool. 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 real upper level logic.

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 detailed 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.

PCSSE 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 environment consists of the next main components: a repository of elementary models, a simulator, and a graphical engine.

Repository of elementary models allows:

- 1) Adding new elementary models, adding their descriptions in menu for the elementary model selection in a model editor.
- 2) Creating elementary models as objects store and returning a reference on such objects.
- 3) Working with models of high-level control.

Simulator allows:

- 1) Adding processes of active elementary models.
- 2) Running model in the step-by-step mode, the real-time mode, the mode of the maximum allowed speed, and the mode with selected speed.
- 3) Rerunning model by the change of initial conditions.

Graphical engine can work in modes:

In editor mode. In this mode new elementary models can be added to the main window and can be moved. The elementary model can be selected as active. In this case, it is possible to change the properties of this elementary model.

In model run mode. In this mode the images of elementary models will be redrawn fixed times per second, thus it allows to animate the model run and do not overload computer with redraw.

PCSSE uses the 2D as the main editor. Each elementary model must have 2D image. Such approach seems more natural for industrial engineering, when all installations and machines appear first on 2D plans (on paper). For 3D representation elementary models must have additional 3D images. 3D representation is an important tool for investigating of emulation models.

The Java and Eclipse RCP is used as a technology and a platform for implementation of PCSSE. Such selection allows having portable implementation on many platforms. Also Java technology is good enough to meet all needs, including performance and graphics (2D and 3D).

The Eclipse platform is based on plugging new functionality to existing core, thus allowing building any kind of desktop applications. PCSSE uses this to organize its program structure.

PCSSE as Eclipse program is a set of plugins (this term is used in Eclipse itself). There is one PCSSE plugin and a number of plugins for the libraries of elementary models. Also there are a number of Eclipse product descriptors that defines an exact set of libraries used plus main PCSSE plugin.

III. THE PROCESS CONTROL SYSTEM FOR NORTH-MUYA RAILWAY TUNNEL

North-Muya railway tunnel is located in East Siberia in Russia. It is a part of the Baikal-Amur Railroad. The length of North-Muya railway tunnel is 15.3 km. It is the longest railway tunnel in Russia. The tunnel has two portals and four ventilating shafts.

The process control system should control the work of a set of units of processing equipment (heaters, centrifugal and axial fans, valves, gates, pumps, and switches). The total quantity of controlled and inspected processing equipment is more than 450 units. The process control system handles more than 2500 input and output parameters. The process control system should support a condition of a microclimate inside the tunnel within certain limits and ensure safe passage of trains through the tunnel.

The process control system has two levels: the top level, and the bottom level. Both levels are incorporated with the help of a local network.

There are Pentium workstations (Operator workstation, Signaling workstation, Engineer workstation, and Server) at the top level. The top level works under Windows NT operating system.

The operator workstation is designed for visualization of the state of equipment and values of inspected parameters with the help of the mimic diagram. The operator workstation provides the operator with an interface for sending control instructions. Control programs handle these operator control instructions. If these instructions can be executed, they are transmitted to the actuators.

There are programmable logic controllers (Octagon Systems, Fastwel), input/output modules (Grayhill) connected to sensors and actuators at the bottom level. The bottom level

works under QNX 4 operating system. Control programs are carried out on the programmable logic controllers.

A simulator was realized for process control system of North-Muya railway tunnel [12]. The simulator contains the following programs and models: the operator workstation, control programs, the emulation model of processing equipment, the model of railway traffic, the model of the environment, and the model of microclimate inside the tunnel.

Real operator workstation allows the interaction of the operator with the simulator and allows the visualization of the run of simulation. Control programs in the simulator are the same ones as in the real process control system, but they are executed in simulation environment.

The emulation model of processing equipment is a set of interconnected elementary models of processing units of equipment (heaters, centrifugal and axial fans, valves, gates, pumps, and switches). The model emulates the state of these units and performs control instructions from control programs to these units.

The model of railway traffic implements real-world railway timetable through the tunnel. The model of the environment simulates the values of air quality parameters in tunnel entrances in winter. These values were calculated on the basis of many years observations information.

The model of microclimate inside the tunnel calculates the values of parameters inside the tunnel (temperature, pressure, speed of air flow) in relation to states of processing equipment, environmental conditions, and railway traffic.

The certain values of these parameters initiate automatic start of control programs. Control programs change states of processing equipment. New states lead to the change of parameters of tunnel's microclimate. The structure of the simulator is shown in Fig. 1.

The simulator was used for following purposes:

Testing of control programs offline. The model of the environment and the model of railway traffic were executed with different input data for comprehensive debugging and testing of control programs.

Optimization of control programs offline. Control programs were modified to solve the optimization problem using a feedback (control programs, the model of processing equipment, and the model of tunnel microclimate) in order to test their logical operation prior to commissioning.

Training of operating personnel. The operator workstation was used for training of operating personnel of process control system.

Testing of control programs on-site for commissioning. Testing of process control system on-site for commissioning was realized under the conditions of the partial availability of processing equipment. A part of processing equipment was real. Another part was replaced the model of processing equipment.

The simulator can be operated in real time and can be operated faster than real time for the simulation of a long period of time.

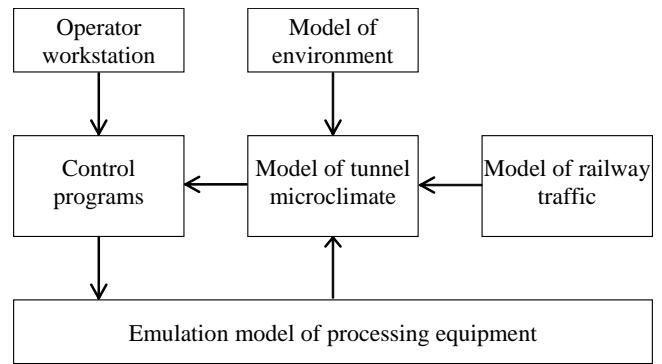


Fig. 1 mapping the structure of the simulator

The distinctive feature of this work is the automatic generation of some part of control programs. There are three levels of control: an individual control level (the manual remote control), a group control level, and an automatic control. In the case of the manual remote control the operator chooses a single unit of equipment and sends a control instruction to it. For example, there are two instructions for a centrifugal fan: turn on and turn off. In order to turn on a group of fans it is required to make a sequence of similar operations.

In the case of the group control the operator chooses certain control program and sends a single start-up instruction to perform a group operation (for example, turning on a group of fans).

Such control program is realized as sequence control program with steps (step-by-step program). After execution of the start-up instruction such program is carried out sequentially step-by-step, beginning from step 1. Each step of such program is a control instruction to a unit of equipment. If the step is finished successfully, the following step (on number) is carried out. If the step was finished unsuccessfully (e.g., the critical time has elapsed), alternative step is carried out.

Control programs are carried out on the programmable logic controllers (PLC). Own PLC programmable system is used for development of step-by-step control programs [13]. PLC programmable system is intended for development of step-by-step control programs not in the programming language but in the simple specification language with the help of the special interface. The main window of own PLC programmable system is shown in Fig. 2.

The interface is used to provide the user with the possibility to see known designation of equipment units. Instead of filling in the table the user selects from the represented lists the required unit and the required instruction.

These lists reduce from a database and are sorted by types and location of units. In the list not only KKS (Kraftwerk-Kennzeichnungs-System) codes of units are given, but also their informal designations and text comments. Special steps, for example, specified time delay can be included in the step-by-step program.

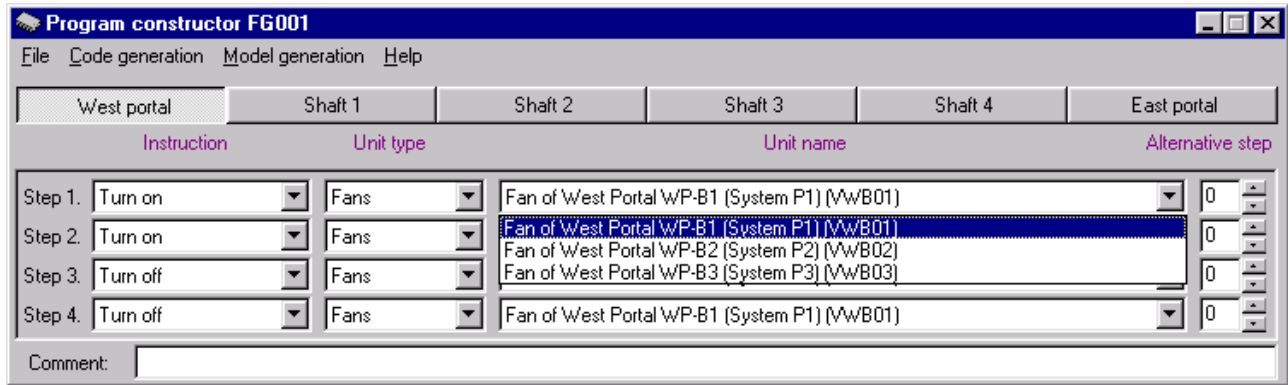


Fig. 2 mapping main window of step-by-step program constructor

When the step-by-step program will be defined by the user the PLC programmable system generates automatically a control program in language C++. Besides, the PLC programmable system generates automatically a model in source language of the simulation system.

It is a possibility of automatic generation of a text document in Microsoft Word together with automatic generation of a control program in language C++.

In the case of the automatic control the truth of some conditional expression initiates automatic start of the control program that performs a group operation (for example, turning on a group of fans). Operands of the conditional expression are current or average values of inspected parameters of the process control system (temperature of outside air, contents of radon in air inside the tunnel and so on).

Together with the step-by-step control program the user can set a condition as a conditional expression. At process control system operation the truth of this conditional expression calls automatic start of the step-by-step control program.

PLC programmable system contains the following components: a simple friendly window interface, an archive for storing ready control programs and subroutines, a code generator for automatic development control programs and conditions of their automatic start in C++, and a report generator for automatic generation of the text document in Microsoft Word. The structure of the PLC programmable system is shown in Fig. 3.

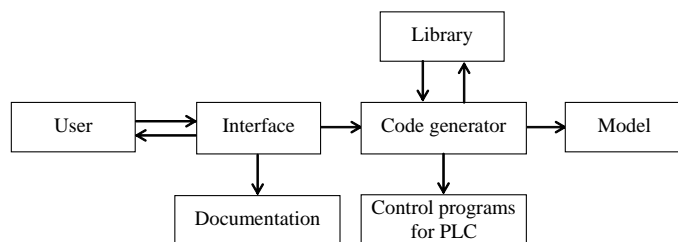


Fig. 3 mapping the structure of PLC programmable system

The process control system for North-Muya railway tunnel has successfully operates for 5 years.

IV. THE MONITORING SYSTEM OF THE TECHNOLOGICAL INFRASTRUCTURE FOR OIL PRODUCING COMPANY

The monitoring system must carry out the following actions:

- 1) Collection and storage of measured values, events in real system and control actions from control system.
- 2) Detection and analysis of trends of key real system parameters that have an influence on producing capacity and safety.
- 3) Making decision for multiple criteria optimization of elements of a technological process.
- 4) Making decision for optimization of equipment repair and maintenance processes.

The structure of the monitoring system is shown in Fig. 4. Monitoring system consists of three subsystems: for data collection, for data storage, and for making decision support.

Data collection subsystem receives input data from wide range of sources, including stationery technological objects and bulk-oil vehicle and pushes them to the data storage subsystem.

Data storage subsystem contains databases of current values of parameters and their changes.

Making decision support subsystem uses methods of artificial intelligence to create a proposed schedule for repairing of the technological objects. It solves also some optimization tasks, recognizes some critical situations, and notifies the operator about recommendations for optimal control of technological objects.

The emulation model generates state parameters values of technological objects (including bulk-oil vehicle) in the same format as the actual system does.

For developing and testing of the monitoring system it is supposed to implement in the model the set of some scenarios that may occur in the actual system including emergency situations.

The implementation of making decision support subsystem is based on the recognition of such situations and solving of optimization tasks. These tasks lead to the decrease of the use of electrical energy and other resources and the increase of safety.

In the model mode the user can execute these

recommendations to control not the actual objects but a model objects with the help of the visual interface. He can check the possible influence of these recommendations by speedup the model, to check visually the effectiveness of the proposed recommendations.

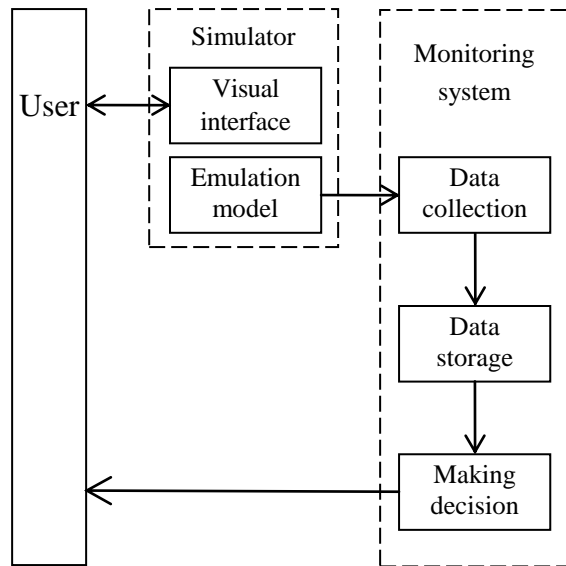


Fig. 4 mapping the structure of the test bed for monitoring system

The connection of the model with the monitoring system provides a possibility to solve some problems. The main problem is debugging and testing of the monitoring system in the test bed. The model generates different input data for the comprehensive debugging of monitoring system programs.

The additional problems are the optimization of the monitoring system and training of operating personnel. The availability of a feedback allows estimating the results of the making decision support subsystem and to improving this subsystem.

The interface of the model is a prototype of an interface of the actual monitoring system that allows attracting of operating personnel for actual monitoring system at the early stages of the monitoring system development.

The model includes several oil well clusters, oil well fluid gathering station, and fleet of bulk-oil vehicle. Some of the oil well clusters are connected with the gathering station by tubes. Others of them are connected with the gathering station by the motor roads.

Each oil well cluster is a submodel which consists of injection pumps and producing pumps. An injection pump injects water into oil well. A producing pump lifts up oil well fluid which goes then into a storage collector. Each pump is connected with a power transformer.

There is a simplified submodel of the oil pool. This submodel estimates total production rate of oil well cluster as a function of producing pumps amount, their productivity, a water injection volume, and the operating schedule of injection pumps operation.

There is a specialized library for oil production application in PCSSE. This library has the following elementary models: an electricity line, a power transformer, an oil well, injection and producing pumps, a tube, an oil pool, an oil well fluid storage collector, a road segment, a bulk-oil vehicle unit, an oil well fluid gathering station.

The important feature of PCSSE is a rapid model building from elementary models with the help of graphical model editor. With the help of the editor the user selects elementary models from library, places and connects their images in the model window, and sets initial values for object and external parameters. A fragment of the model is shown in Fig 5.

The following elementary models are used in this fragment: an electricity line, a power transformer, a pump, a tube, a road segment, a car.

A pump has the following main emulated parameters: a pump delivery, an electric power consumption, and so on. A road segment has the following parameters: length, geographical coordinates of the beginning and of the end of the road segment.

A bulk-oil vehicle unit (a tanker) has the following main emulated parameters: a tanker identifier, a road identifier, geographical coordinates of the tanker on the road, a movement direction, speed, and so on. The traffic of tankers can be visually observed in the model window.

The model generates various situations which may happen in the actual system including emergency situations.

The monitoring system develops a number of different scenarios of control actions depending on the situation and on the chosen optimization criteria.

As an example one can consider selection of oil well pump mode that depends on such criteria as:

- 1) The reliability – to run equipment using allowed modes and to reduce repair costs during the life cycle of the equipment.
- 2) The profit – to increase load on the equipment to receive greater profit and replace equipment before its breakdown.

Several scenarios have been implemented in the model for solving the next optimization problems:

- 1) The repair schedule for technological equipment that takes into account possible prophylactic and accompanied repairs.
- 2) The optimization of summary production rate from the oil well cluster depending on the use of pumps with various performances, a number of working injection and producing pumps, and the operating schedule for these pumps.
- 3) The optimization of the traffic of tankers that takes into account the petrol price, drivers work payment, and so on.

Several scenarios have been implemented in the model for generation emergency situations like overload of power transformer, oil pipeline break, and so on [14].

At present the development of a prototype of the monitoring system is completed.

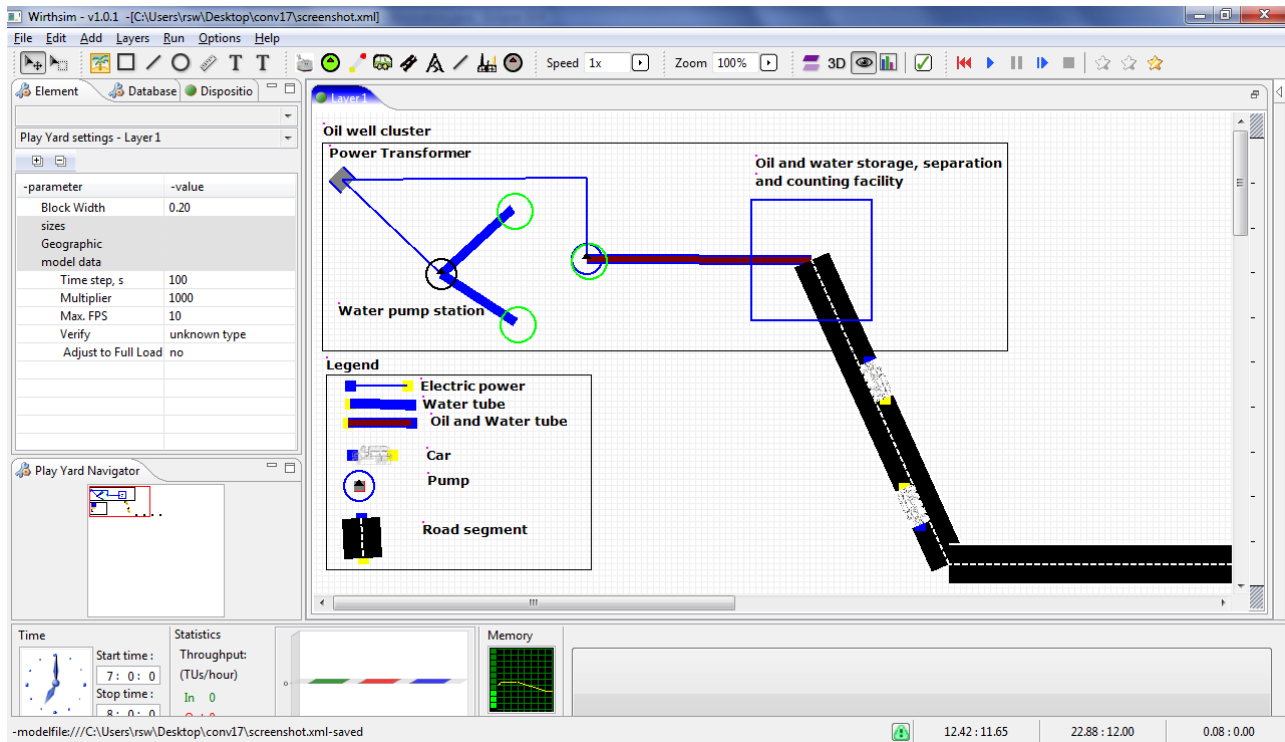


Fig. 5 mapping a fragment of emulation model for monitoring system

At present the emulation model can simulate a normal operation of oil production enterprise and certain emergency situations. New scenarios will be added in the model as the monitoring system will be developed.

Model can be used as a stand-alone application for:

- 1) The comparing of alternative scenarios.
- 2) The answering the questions of the type "what if...".
- 3) The formalizations and examination of expert knowledge that can be used in the monitoring system later on.

V. THE PROCESS CONTROL SYSTEM FOR UNDERGROUND COAL MINING

A coal transportation system contains such units of equipment as: a mine face producing the random flow of a coal, a conveyor having an accepting load and length, a middle-way bunker with defined load and unload speeds. PCSSE library for the simulation of coal mining processes contains appropriate elementary models for emulation of mine face, conveyors, bunkers, and so on.

A user of PCSSE places the graphical images of elementary models in a model window and defines their characteristics. For example, possible properties for conveyors can be: an angle with horizontal direction, a conveyor length, a conveyor direction, and so on.

When model is built, the special parameters must be specified. For example, the simulation start time, simulation stop time (in real clock format), simulation step (how many milliseconds are covered by one time step in the model).

A model of a conveyor network which is developed with the

help PCSSE library for the simulation of coal mining processes is shown in Fig. 6.

The elementary model of conveyor has a button for manual sending of commands to conveyor. In simulation run a user can send simple or complex commands to conveyor and other elements of mining transportation model.

Models created with the help of the library for simulation of conveyor networks have two-level structure and two-level logic. Low-level logic is intended for control over models of the individual units of equipments. It is embedded in elementary models. There is a set of commands and technological restrictions for control over each elementary model. For example, if a conveyor is out of repair, then it is impossible to perform the control command "turn on a conveyor".

Upper-level logic is intended for control over models of all units of equipments or a subset of them. It also has own restrictions. For example, the performance of the control command "turn off a conveyor" will be detained while the previous conveyor (in coal flow direction) will not be stopped. In contrast to the low-level logic the upper-level logic can have some alternative versions. The opportunity to select the version of the upper-level logic allows solving problems of optimization.

The upper level logic can either be the part of the model, or be replaced with the operating logic of the communicated real control system. In the first case, the model is executed separately. It is used for developing, debugging, and comparing of the control algorithms.

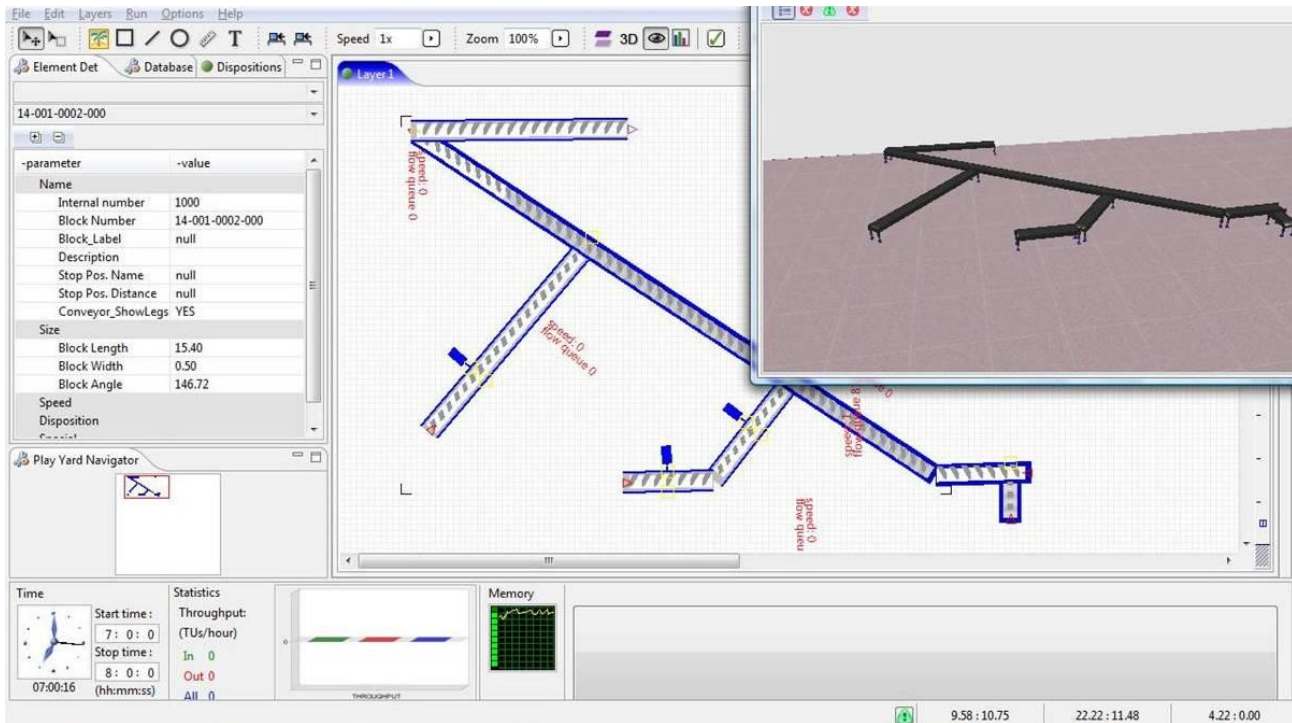


Fig. 6 mapping an example of a conveyor network model

Such a model is performed as a simulation model. In the second case, the model is connected with a real control system. It is used for testing control programs of the real control system. Such a model is performed as an emulation model.

With the help of the emulation model the following problems for the real process control system were solved:

- 1) Centralized turning on conveyors at the direction opposite to a coal flow.
- 2) Turning on a conveyor after the achievement of required speed of a previous conveyor.
- 3) Turning on conveyors in defined operating mode.
- 4) Regulation of the speed of a conveyor belt depending on size of a flow coal.
- 5) Emergency turning off whole conveyor network by turning off all conveyors at the same time (halt of the conveyor network, for example, if a methane level is high), and so on.
- 6) Optimizing of the energy consumption (lowering of energy consumption needed by conveyor network to work).
- 7) Creation of training tool to teach the coal mining personnel.

In the future we intend to use PCSSE for simulation possible strategies for underground mining without miners. These strategies are: replacement of miners by remote-controlled manipulators, robotics-based mining, control of mobile machines from surface, and control of mobile machines by on-board control systems.

VI. CONCLUSION

The ability of interaction of actual control systems with models eliminates the need to recreate control logic in simulation models. Incorporating emulation into a project is an extremely valuable approach for development process control systems.

We demonstrate the use of this approach by the example of several developed process control systems. This result into cost savings, reduction in schedule, rapid startup at a reduced cost and an increased safety.

Emulation models for process control systems in various industrial fields were created with the help of new visual interactive discrete simulation environment PCSSE.

PCSSE is divided into simulation workflow and a set of libraries of elementary models, therefore its application can be enhanced by other industry areas by developing mode libraries.

REFERENCES

- [1] I. McGregor, and R. A. J. Walters. (2001). Emulation Overview. Available: <http://www.automod.de/media/doc/Mcgregor.pdf>
- [2] G. Mueller, "Using Emulation to Reduce Commissioning Costs on a High Speed Bottling Line," in *Proc. 2001 Winter Simulation Conference*, Arlington, 2001, pp. 1461–1462.
- [3] I. McGregor, "The Relationship between Simulation and Emulation," in *Proc. 2002 Winter Simulation Conference*, San Diego, 2002, pp. 1683–1688.
- [4] C. Versteegt, and A. Verbraeck, "The Extended Use of Simulation in valuating Real-Time Control Systems of Avgs and Automated Material Handling Systems," in *Proc. 2002 Winter Simulation Conference*, San Diego, 2002, pp. 1659–1666.

- [5] C. Starner, and M. Chessin, "Using Emulation to Enhance Simulation," in *Proc. 2010 Winter Simulation Conference*, Baltimore, 2010, pp. 1711–1715.
- [6] 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.
- [7] 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.
- [8] 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, Modeling, and Optimization*, Corfu, Greece, August 17-19, 2005, pp 243–249.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] V. Okol'nishnikov, "Use of Simulation for Development of Process Control System", in *Proc. 2008 IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering*, Novosibirsk, Russia, July 21-25, 2008, pp. 248–251.
- [13] D. Chernakov, V. Okolnishnikov, "Control Program Development System", in *Proc. of the Second IASTED International Multi-Conference Automation, Control, and Application (ACIT-ACA)*, Novosibirsk, Russia, 2005, pp. 142–145.
- [14] V. Okolnishnikov, S. Rudometov, and S. Zhuravlev, "Monitoring System Development Using Simulation," in *Proc. 2010 IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering*, Irkutsk, Russia, July 11-15, 2010, vol. II, pp. 736–739.