Web-based Training for Polymer Plant Operators using Process Simulation

Internetbasierte Schulung von Polymeranlagenfahrern mittels Prozesssimulation

Rasmus Nyström, Andreas Kroll, Alexander Frick, Christian Stich, Klaus Hütten, Franz Langhauser, Wilhelm Seebass, and Meinolf Kersting

This paper describes a framework offering educational services for chemical plant operators using a process model simulated in real time over the Internet. An application service provider (ASP) approach avoids the barrier of capital investment of a site-based installation. Simulation-based education can be made available for virtually anyone at any time and anywhere through the use of thin clients. At the same time, the maintainability of the server software and model is greatly increased through the centralization of the simulation server. The utility described uses a rigorous dynamic process model for simulation but requires only a computer with a web browser and an Internet connection to establish real-time data communication with the simulation server.

In diesem Beitrag wird ein System vorgestellt, das die Schulung von Anlagenfahrern der Chemischen Industrie am Prozesssimulator in Echtzeit über das Internet gestattet. Der Dienst wird im Application Service Provider (ASP) Modell angeboten, das die Investitionshürde einer Vor-Ort-Installation vermeidet. Gleichzeitig verbessert sich die Wartungsfreundlichkeit des Systems durch den zentralen Simulationsserver wesentlich. Durch die Verwendung von Thin Clients wird die simulatorbasierte Ausbildung praktisch für jedermann, jederzeit und überall zugänglich. Die vorgestellte Lösung verwendet ein rigoroses dynamisches Prozessmodell, erfordert aber nur einen Computer mit Standard-Web-Browser und Internetzugang, um einen Echtzeitzugang zum Simulationsserver herzustellen.

Keywords: Operator training simulation, process learning, process education, Internet-based training, web-based training, e-learning

Schlagwörter: Operator-Trainingssimulation, Echtzeitsimulation, Prozessverständnisschulung, Web-based Training, E-Learning

1 Introduction

Computer-based educational services in the chemical process industries are expanding with growing computational resources and improved tools for process modelling and visualization. An example is operator training simulation (OTS), which provides a means of safe and eff cient training for plant operators and other production personnel by allowing the training of actual plant operation using computer simulation. With training simulators, operators may be trained to handle normal and abnormal plant operating conditions without endangering the actual plant (see [1] for a detailed discussion).

An OTS consists of a model of the process, controllers, at least one human system interface (HSI), training/learning utilities, optionally a database, and the software interfaces necessary to integrate these main components. The look and feel of OTS systems vary from a very high level of detail, where the OTS exactly mimics the operator station of the actual plant and is tuned to a particular site, to a more abstract level of detail, where the purpose of the simulator is to provide a general knowledge of the process behaviour. This paper focuses on the latter.

In order to make distribution and maintenance of simulators easy and/or to protect the knowledge with regard to the process model, it is beneficia to provide learn-

https://doi.org/10.1524/auto.51.11.525.19582 Die Verlagsveröffentlichung ist verfügbar unter www.degruyter.com

ing services on the worldwide web through an ASP. The idea is to enable access to the application any time and anywhere, as long as a computer with an Internet connection is available. The set-up uses a server that makes simulation services accessible through standard communication protocols. The simulation server should be able to serve one or several clients depending on demand, optionally with different modelling scenarios or process states; to conduct bookkeeping and supervision of the simulation sessions; and to provide adequate security, to protect proprietary information. Further, simulator crashes, broken connections, server-side memory leaks, and other software malfunctions should be handled to ensure continuous operation.

The client software should be as simple as possible in order to avoid installation and maintenance/licensing costs.

Web-based simulation services have gained recent interest and a few are already commercially available. AspenTech, Chemstations and SimSci, among others, provide such simulation services for engineering purposes. These offerings emphasize high fi elity of the process model, but not necessarily the ability to provide real-time or faster-than-real-time simulation with on-line data transfer. ABB University [2], and Honeywell Automation College, for example, provide web-based training for learning to interact with operator stations. Examples of web-based generic simulator training or learning tools are Honeywell's Shadowplant Web [3], the offering of BVCT [4], and Proper Educt [5]. Examples of customized OTS systems for polypropylene were published by Borealis [6] and ABB Simcon [7].

The purpose of this paper is to present a process-learning tool developed for Novolen Technology Holdings C.V. (NTH), a joint venture of ABB Lummus Global and Equistar. The training target is a polypropylene production plant using Novolen®technology. The utility is a next-generation development of the tool described in [8]. The Process Web Server (PWS, [9]) platform by ABB is used to create an HTTP-based service, which may be accessed by standard browser clients using HTML pages with Java applets. The use of several small Java applets enables a fast start-up of simulation sessions and requires only low bandwidth for data transfer, as opposed to, for example, remote desktop technologies. The user may have his/her preferred browsing functionality while navigating within the application.

2 Novolen Polypropylene Polymerization Process

The process targeted for the learning tool is the gas phase polypropylene process technology owned by NTH. In the Novolen process, polymerization is conducted in one or two gas-phase reactors as shown in Fig. 1. The reactors contain a bed of polypropylene powder that is agitated below the f uidization point by an agitator to keep the bed in motion and prevent powder agglomeration. A wide range of products can be produced with only two reactors connected in series, including super-high impact copolymers. The second reactor is used either to incorporate rubber into the homopolymer matrix produced in the f rst reactor, or to increase the capacity while producing homopolymers or random copolymers. Polymerization heat is removed from the reactors by external cooling circuits. Polymer powder is continually withdrawn from the reactors. The powder transfer from the f rst to the second reactor and from the second reactor to the gas/solids separation unit is pressure driven. In this gas/solids separation unit, polymer powder is separated from unre-

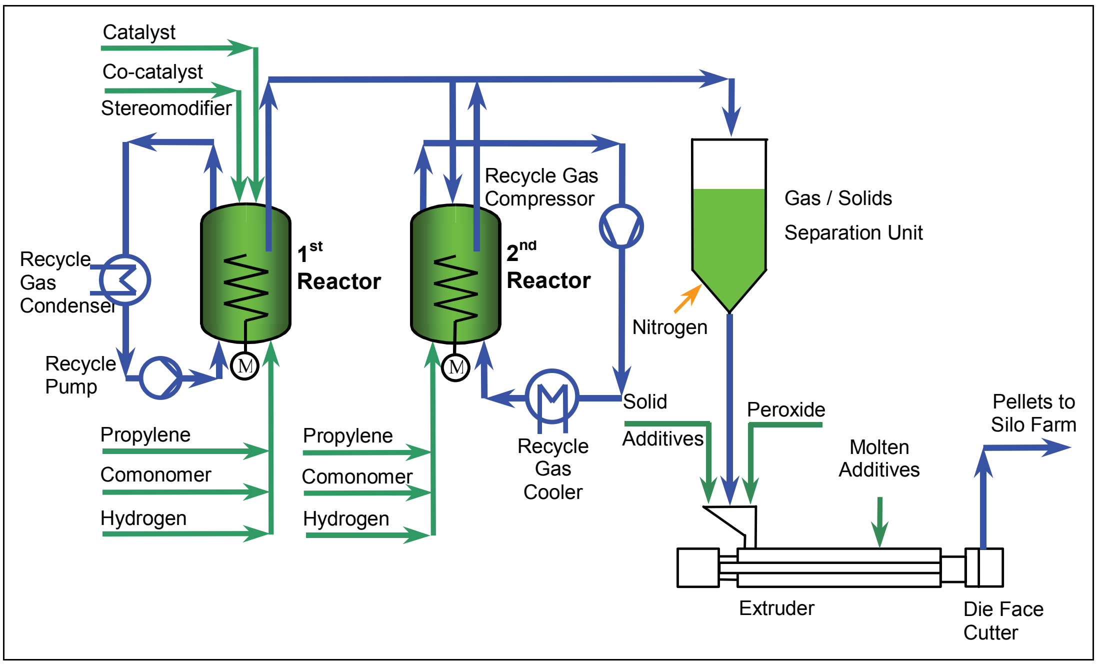

Figure 1: Flowsheet of the Novolen polypropylene process.

acted monomer and directly fed to the extruder for pelletizing.

The Novolen polypropylene plant operators need to know how to manipulate the product quality indices in order to make transitions from one product specif cation to another. They also need to be aware of the time constants involved (particularly the residence times and reactor dynamics), typical process responses, and what effects the process responses have in the plant.

3 Dynamic Process Modelling and Simulation

The dynamic process model for the learning tool does not need to be of "engineering quality", but should produce qualitatively adequate responses to any operator actions. That is, the model should be valid over a broad operating range so that scenarios such as cold/hot start-up and shutdown may be implemented. Moreover, it should be possible to simulate the model in scalable real time, particularly many times as fast as real time. Thus, special attention may be given to producing a model, whose differential equations can be solved quickly and is robust against numerical failure. It is thus of interest to: 1) reduce the number of states; 2) eliminate algebraic loops; 3) reduce the stiffness of the model; and 4) use smooth transitions instead of „if-then-else" conditions, all of which will either reduce the likelihood of simulator crashes, and/or speed up simulation.

In the learning tool described here, an Advanced Continuous Simulation Language (ACSL, [10]) model was used. The model was part of an earlier learning tool [8] and has since undergone some changes. It is a pure ordinary differential equations model with about 100 states, including two reactors, the cooling systems, and the control loops. The choice of the modelling language and simulation environment ACSL was based on the following considerations: 1) the model can be compiled into a binary library f le, which protects know-how and can be used without a license fee; and 2) the model, integration routines, and other features can be accessed via an application program interface (API). The model is relatively stiff, but can be satisfactorily simulated to support a speed-up factor of up to 3600 times real time.

The interfacing with the ACSL model is done through direct access of variables in the model's address space using the ACSL API. The API also allows the use of ACSL's simulator (integration software). The simulation can be started, stopped and restarted through the API, and other commands can also be passed to ACSL. During simulation, the simulator calls C++ callback routines, which eliminates the need of the simulation server stepping the model. Thus, the timing software can be kept within the model and the synchronization issues are addressed through caching.

4 Functionality of the Simulation System

The required functionality depends on the use cases and the business model of the simulation system. The process-learning tool is used by NTH, which is a global process technology licensor. The process designs of Novolen polypropylene plants delivered to licensees are similar, whereas the control systems vary. This makes a generic learning tool the preferred alternative to a fully tailored OTS system. The targeted trainees are plant operators, operator trainers, and process engineers. The plants are distributed around the world; remote locations are the standard rather than the exception. The pricing needed to be signif cantly below tailored, full-scale OTS systems, which typically range between US$ 0.5 and 4 million [1].

A generic operator station and generic control algorithms are used in order to avoid having to maintain several different control systems in the OTS. The model may be adjusted to ref ect different plant sizes. The models are not f ne-tuned to become a virtual replica of a particular plant. The system is web-enabled in order to offer attractive simulation services worldwide, at the same time being easy to introduce and use.

A major training objective is that the operators learn the dynamic characteristics of the plant during standard operation and in abnormal situations, including process disturbances, start-up/shutdown of the plant, and transitions between different product grades. Moreover, they obtain an understanding of the relationship between operating conditions and resulting product quality. This permits operators to improve their ability to operate plants and to diagnose problems. For this reason, key product properties are calculated and displayed. The simulation speed can be signif cantly accelerated (during operation) to avoid losing training time during uninteresting phases and to get quick feedback within what-if studies. Standard control systems visualization functions such as alarm animation, trend plots, and hierarchies of schematics, are available. Additionally, f aring is animated to give this unwanted situation more signif cance.

Table 1 lists some of the standard functions of the system. For remote operation, all required functions, such as (dis)connect to server or password management, are available. Besides the web-based training option, the system can also be fully installed and run on a PC.

The components in the user interface, such as valves, numeric text boxes, buttons, level indicators, and f ares, ref ect the current status of the simulation. Where necessary, the components can be used to change the course of the simulation. For instance, valves can be closed/opened by clicking on them, and specif c numerical entries can be fed by using the numeric text boxes.

The simulation can be started and stopped by the user, and restarted from its initial state. The user can also select between a set of preconf gured initial states when restarting the simulation. Further, the snapshot functionality enables

Function	Description
Run/restart	Start the simulation (*run*) or *restart* it from the beginning.
Freeze/resume	Pause (*freeze*) and then continue (*resume*) the simulation session.
Snapshot save/restore	Save system state in an instant *(snapshot save*). This state can later be *restored*.
Evoke malfunction	Evoke an abnormal behavior such as failure of cooling cycle or stirrer.
Field operator function	No distinction is made between f eld operator input and control station input.
Fast/slow time	Simulation time can be scaled by an acceleration factor (1 = real-time).
Self-paced training	Run a training session without an instructor.
Set initial condition	Choose pre-conf gured initial condition from a set of available ones.
Trend data	Trend graphs showing historical data, updated with the current status of the process.

Table 1: Training-related system functions.

the storage of the current simulation state, which can be used as an initial state in subsequent simulations.

The major simulation variables can be displayed in trend windows, which are initialized using historical data, and which evolve with the simulation. The user can conf gure the trends in several ways.

5 Software Architecture

The software architecture required for ASP-based online process learning will be presented in the following. The most important requirement to enable real-time simulation is to provide a real-time data stream from server to client and vice versa. In addition, limitations are caused by the bandwidth of the client-server connection. The estimated worst-case lower bound is given by an analogue modem connection with 57.6 K Baud. A non-monolithic approach overcomes long loading and starting times; web pages are loaded and live data f ow is established on demand. The general architecture of the application is illustrated in Fig. 2. The interaction between the client and the server takes place via Hypertext Transfer Protocol (HTTP) and downloading of Java applets from the server. The process-learning tool is implemented in such a way that a thin client, a standard Internet browser with Java, can access the simulation service. The client-side applets are upwards compatible and are running with Java version 1.0 and upward. This enables all major browsers with Java capabilities to be used for server access.

On the server side, the Process Web Server (PWS, [9]) by ABB provides the functionality needed to offer dynamic data f ow from and to live data on web pages, and consists of a multi-tier-architecture. The core is an Apache

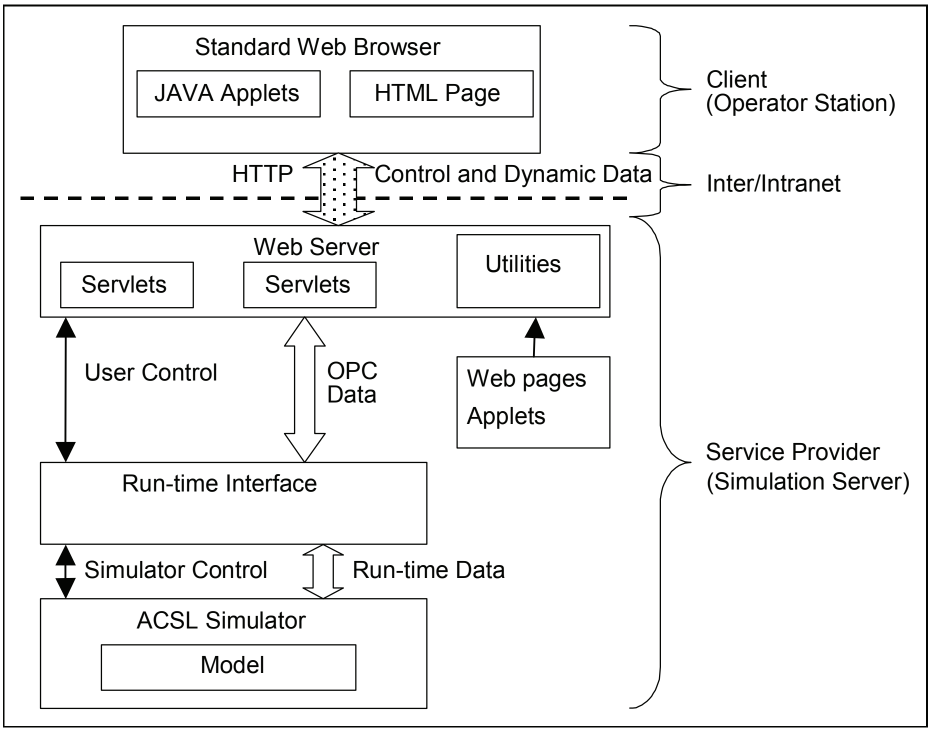

Figure 2: Overview of the software architecture.

web server with a Tomcat servlet engine. The PWS framework extends it by OPC connectivity. Data can stream from the internal OPC client to Java servlets, which are connected via HTTP to Java applets "living" on the client side. ABB developed the internal OPC client by using the Java Native Interface to bridge the gap between Windows 32 applications and Java, and to enable data f ow from Microsoft's DCOM-based OPC server via the OPC client to Java servlets.

The run-time interface (RTI) encapsulates the ACSL process model and simulator. Thus, they become abstracted from the server software, so that models written on other simulation platforms can be plugged in with minimum effort in the future. The simulation runs cyclically, and the order of magnitude of the cycle time is one second. Hence, the OPC client connects asynchronously to the RTI to avoid waiting for current simulation data.

A screenshot of the client GUI is shown in Fig. 3. The applets on the client side visualize the OPC tags of the connected simulation engine. The data exchange for a screen is handled via one connection applet, which serves as data sink and source for all dynamic applets of a loaded and displayed web page. All applets are conf gured to subscribe to their required dynamic data via the connector applet. This applet communicates with a matching servlet via HTTP. The HTTP uniform resource locator connection of Java's net class handles all communication between server and client by input and output streams. Thereby the usage of URL messages guarantees the cyclic data communication between the connector applet and the server. In addition, the usage of the standard HTTP port 80 avoids the complication of f rewalls or proxies between server and client. ABB's Process Web Server can also manage different sessions. A session cookie stores the session data for the current instance of client and server. Web pages are only accessible if the right session key is set after log in of the user. User interactions are recorded in several log f les. These log f les are produced by the underlying apache web server and can be interpreted and evaluated by standard tools to track the user behaviour in detail and to determine possible payments for the usage of the application.

To enhance the process-learning experience, different small-scale Java applets present the data in different manners:

– Trends showing the values of the process variables over time in coordinate systems
– Numeric displays, optionally editable, showing the present values of process variables
– Dynamic/animated images, which change their appearance depending on the current values

Internally, these applets are mapped to OPC tags and possess read or read/write functionality as required. The applets are conf gured through applet parameters. In some cases the applet parameters need to be reconf gured by the user. This is done using a corresponding servlet, which generates a new context for the applet.

The HTML-based approach provides for f exible linking to other information sources, and enables integration with

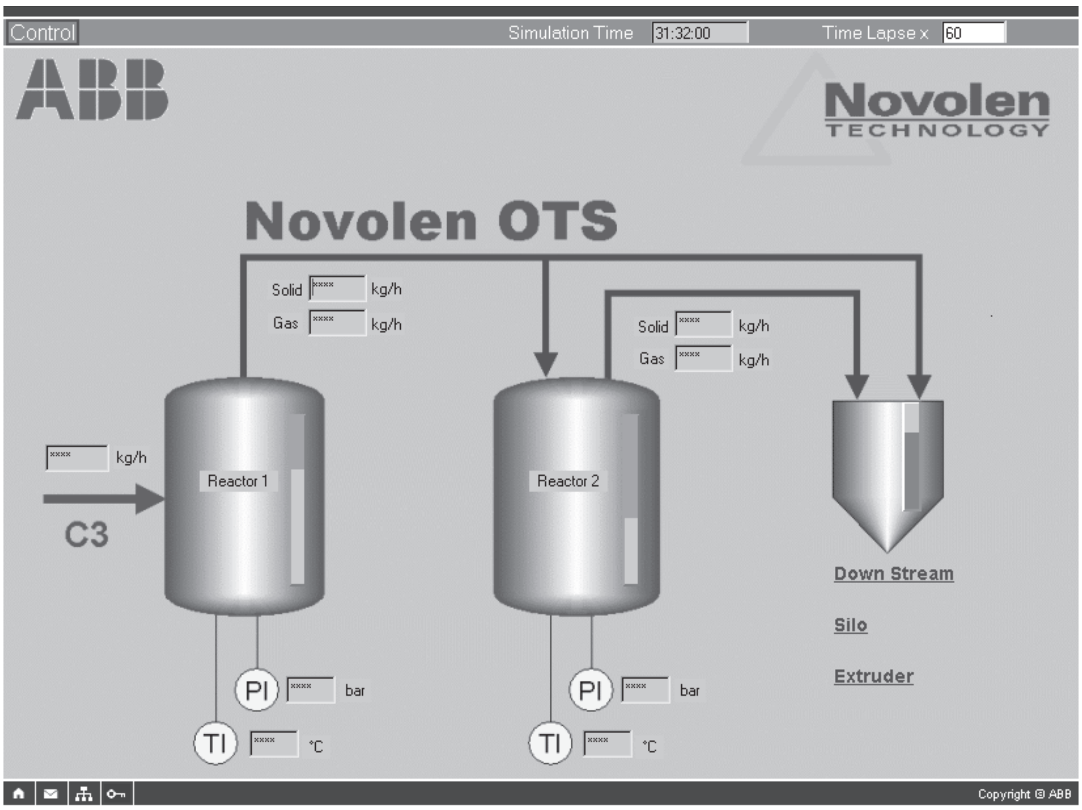

Figure 3: Screenshot of graphical user interface for process overview.

other e-learning systems. The Process Web Server delivers a plug-in for Microsoft's Frontpage to simplify the development of the web pages and the embedded applets. A designer can easily drag and drop HTML page elements and ABB's applets, and configure the applet parameters. This speeds up development of web-based GUIs for server-hosted applications. Provided style sheets enable the uniform design of an application.

A normal simulation session starts by navigating to the web site of the server. After a login procedure in which the access rights are checked, a simulation is started with one of several pre-configured scenarios. The client first receives an HTML page, then the according Java applets of the page are transmitted and the dynamic data flow is established via HTTP to the web server. The user may influence the system by transmitting writable values, which are received by the simulation engine via servlets and OPC on server side. Due to the client–server concept, data are stored only on the server side. Users can save different scenarios or snapshots on the server and can access their data later according to their privileges.

6 Summary and Outlook

The demand for operator training has increased due to the demand for reduced downtime, for improved operations, and due to staff fluctuation/generation change. Web-based process simulation can support the fulfilment of this demand in a cost-effective manner. A simulation system that permits training of Novolen polypropylene plants operators on a rigorous dynamic plant model via Internet, Intranet, or disconnected was presented. The application makes use of state-of-the-art web technologies.

Acknowledgements

The European Union is gratefully acknowledged for supporting the post-doctorate research of R. Nyström under the Marie Curie fellowship, contract no. HPMI-CT-2001-00138.

Abbreviations

ACSL	Advanced Continuous Simulation Language
API	Application Program Interface
ASP	Application Service Provider
GUI	Graphical User Interface
HSI	Human System Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
OPC	OLE for Process Control
OTS	Operator Training Simulation/Simulator
PWS	Process Web Server
RTI	Run-time Interface
URL	Uniform Resource Locator

References

[1] *A. Kroll*. Trainingssimulation für die Prozessindustrien: Status, Trends und Ausblick. Teil 1 + 2. Automatisierungstechnische Praxis atp (2003) 45, H. 2 (S. 50–57) + H. 3 (S. 55–60).

[2] *ABB*. Web based training: Reduced cost for effective qualification. Cottbus: ABB Automation Systems GmbH. See also URL: http://www.abb.de/wbt-en.

[3] *Honeywell*. URL: http://www.iac.honeywell.com.

[4] *F. Winter, F. Schmidt*. Online-Trainingssystem für Bedienerschulungen in Unternehmen der verfahrenstechnischen Industrie. In: Automatisierungstechnik im Spannungsfeld neuer Technologien. VDI Berichte (2001) Nr. 1608, S. 457–467. VDI-Verlag, Düsseldorf.

[5] *L. Urbas*. Entwicklung und Realisierung einer Trainings- und Ausbildungsumgebung zur Schulung der Prozeßdynamik und des Anlagenbetriebs im Internet. Fortschr.-Ber. VDI Reihe 10 (1999) Nr. 614, VDI-Verlag, Düsseldorf.

[6] *T. Kokko, S. Smeds, L. Hammarström, J. Lindholm, K. Yli-Opas*. Training simulators for polyolefin processes. Automaatio 2001, 4.–6.9.2001, Helsinki.

[7] *W. Bucey, D. C. Pathe*. 1990. Operator training: process simulator key to emergency training. 1990 Honeywell User Coference, Phoenix, Arizona.

[8] *C.-U. Schmidt, R. Böttcher*. 2000. ProveT – Training simulator for Novolen Process. 1st European Conference on the Reaction Engineering of Polyolefins, 3.–6.7.2000, Lyons, S. 114–116.

[9] *P. Müller, A. Zehnpfund*. 2003. Ein Framework für web-basierte Serviceanwedungen. Telematics Conference, 17.–18.6.2003, Siegen,
URL: http://www.telematics-conference.de.

[10] *The AEgis Technologies Group, Inc*.
URL: http://www.ACSLsim.com.

Manuscript received: 13th May 2003.

Dr. Rasmus Nyström, Dr. Andreas Kroll, Dr. Alexander Frick, and Christian Stich are employed at ABB Corporate Research Center Germany, Department "Industrial IT Software & Applications".
Dr. Nyström works as post-doctorate researcher under a Marie Curie fellowship sponsored by the European Union. Dr. Kroll leads the subdepartment "Control & Optimization". Dr. Frick works as scientist and project manager. Mr. Stich leads the subdepartment "Advanced Process Management."

Address: ABB Corporate Research Germany, Wallstadter Straße 59, D-68526 Ladenburg, Germany, Fax: +49 (0) 6203-716253, E-Mail: {Rasmus.Nystroem, Andreas.A.Kroll, Alexander.Frick, Christian.Stich}@de.abb.com.

Klaus Hütten, Dr. Franz Langhauser, Dr. Wilhelm Seebass, and Dr. Meinolf Kersting are employed at ABB Lummus Global GmbH, Novolen Technology Division.
Mr. Hütten, Dr. Langhauser, and Dr. Seebass work as Senior Technology Managers for Electrical Engineering & Instrumentation, Process Technology and Sales & Aftersales, Dr. Kersting is currently Director of Technology Support Services.

Address: ABB Lummus Global GmbH, Gottlieb Daimler Straße 8, D-68165 Mannheim, Germany, Fax: +49 (0) 621-4381465, E-Mail: {Klaus.Huetten, Franz.Langhauser, Wilhelm.Seebass, Meinolf.Kersting}@de.abb.com.