Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

DATABASE DRIVEN FACTORY SIMULATION: A PROOF-OF-CONCEPT DEMONSTRATOR

Lars G. Randell
Gunnar S. Bolmsjö

Division of Robotics
Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, SWEDEN

ABSTRACT

The paper presents a database-based method to reduce the development time and project lead-time for large discrete-event simulation models of entire factories. The database used to automatically generate and drive the simulation model is a copy of the production planning database. A set of proof-of-concept tools and a database have been generated to verify the method and it has been shown that it is feasible to run a simulation using the production planning data as the only information source. The software developed is modular and designed to work in heterogeneous environments. The method is expected to reduce the modeling and maintenance effort considerably when modeling entire factories. The method will result in a holistic and fairly accurate assessment of performance measures for an entire factory.

1 INTRODUCTION

In general the most time taking phases of a discrete-event simulation study is the input data collection and model building (Banks, Carson, Nelson, and Nicol 2000; Bernard 2000; Umeda and Jones 1997; Liyanage and Perera 1998; Trybula 1994). Building large DES (Discrete-Event Simulation) models of entire factories with traditional methods is thus time consuming and the required maintenance effort is bound to lag behind. A better solution would be to have model information stored in a continuously updated database. However, creating and maintaining such databases is also time consuming. To motivate such databases the information should be shared by several applications, thus reducing the total information management time for the applications sharing the information.

Several authors have presented integrated methods. In enterprise modeling the enterprise is modeled and that representation can then be reused to generate simulation models (Delen, Benjamin, and Erraguntla 1998; Delen, Benjamin, and Erraguntla 1999; Heim 1997; Srinivasan and Jayaraman 1997; Whitman, Huff, and Presley 1997; Zhang and Browne 1999). Bernard (2000) has reused the information generated from a process design tool to generate simulation models using STEP (Standard for the Exchange of Product Model Data). Others have developed an advanced method for building simulation models from a set of simulation model components stored in a database (Gmilkowsky, Eckardt, and Palleduhn 1997; Gmilkowsky, Eckardt, and Palleduhn 1998). Johansson (2001) have presented methods to store, share and communicate manufacturing system information using STEP. Giaglis (1999) and Eatock, Serrano, Giaglis, and Paul (1999) use DES to simulate the business process, the information system and how they interact and affect each other. Love, Clarke, and Gooden (1987) integrated a simulation model with an MRP (Manufacturing Resource Planning) system. The two systems interacted by exchanging input and output data on a period by period basis. Iuliano (1995), Iuliano and Jones (1996), Iuliano, Jones, and Feng (1997), Iuliano (1997), Ellis, Jones, and Lee (1998) presents the integrated Manufacturing Engineering Toolkit (METK) prototype developed at the National Institute of Standards and Technology (NIST). Drake and Smith (1996), Judd and Abell (1996), Peters, Smith, Curry, LaJimodiere, and Drake (1996), Smith, Wysk, Sturrock, Ramaswamy, Smith, and Joshi (1994) and Smith and Peters (1998) presents an integrated set of tools for a shop-floor control system development. The integrated tools retrieves data from a database in a manner similar to the work presented here.

Bernard (2000) makes a distinction in between *integration of information* and *system integration*. Information integration enables sharing of common information between applications while system integration allows control of one application from another. The research presented here will be limited to the integration of information.

The basic problem with large simulation models is, besides the size, the complexity of the model logic. Conventional methods embed data and logic into the simulation model, which works well until the size of the simulation model reaches a certain size. This size is reached when the simulation model cost reaches the potential cost savings,

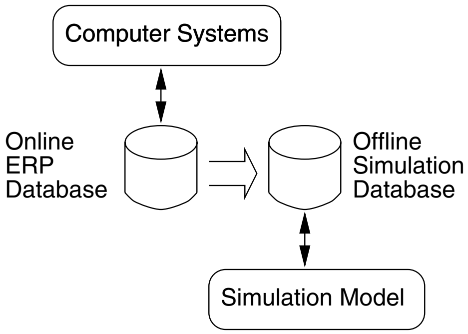

Figure 1: The Used Data Input Method

the system under study changes before the model is ready, the results arrive after the point of usage (Ball and Love 1994) or when the maintenance effort exceeds practical limits. When this size is reached other methods have to be considered.

Previous research performed at BT Products (Randell 2000; Randell, Holst, and Bolmsjö 1999a; Randell, Holst, and Bolmsjö 1999b; Randell, Holst, and Bolmsjö 2000; Bolmsjö, Lorentzon, and Randell 1999) has presented methods to build and maintain large models using traditional techniques. The focus has been on design process integration and parallel development to reduce development time and project lead-time.

This paper presents an approach to reuse readily available manufacturing system information without creating new databases; instead information stored in an ERP (Enterprise Resource Planning) database is reused.

The model is currently built manually, but will be built automatically from the resource register in the final implementation. Adding additional fields for geometry, shift and location in the resource register will allow automatic generation of the visual model as well. The resource register from the previous simulation will be saved and compared to the resource register in the next simulation run. The user will be prompted to add geometry, shift and location in the database for the new objects.

The model objects share common logic to retrieve the information required to run the model. The input data collection is reduced to identifying the correct ERP database tables and copy them to the off-line simulation model database, as shown in Figure 1. Furthermore, the model logic can be simplified considerably since the operation lists in the database has information about routings, which generally has to be embedded in the simulation model. The result is thus a completely data-driven simulation model.

2 PLANNED USAGE

The set of tools developed is planned to be used for testing master plans for the BT Products plant in Mjölby a few months ahead. BT Products is a world-leading manufacturer of electrical warehouse trucks. The plant at Mjölby has 1'000 employees and produces approximately 35'000 warehouse trucks a year. To test master plans the entire factory has to be modeled and all products produced in the factory have to be simulated. The simulation will be fed with a master plan that then is broken down to production orders. The simulation is then run to find bottlenecks and delayed deliveries. By identifying the bottlenecks possible countermeasures can be implemented in the simulation database before the simulation is run again. Since the simulation is entirely driven by a database all modifications to the simulation model is performed in the database that then is used to generate the actual simulation model. Possible countermeasures are e.g. added shifts, outsourcing to suppliers or added resources. The impact of future investments in new resources can also be tested. A rough estimate is that approximately four experiments will be performed during a day before a final master plan is established.

3 REQUIREMENTS

The following requirements have been regarded in the initial phases of the project;

Time The method should reduce development time and project lead-time.

Modeling Creating and running a simulation model should require little or no manual intervention.

Software independent The database should be built and maintained independently of the DES software used. The DBMS (DataBase Management System) should also be exchangeable.

Reusable The method should be transferable from one project to another (Holst, Randell, and Bolmsjö 2000).

Modular The method should be modular to support the reusability requirement.

Scalable The method should be applicable for large or even very large enterprises.

Heterogeneous The method should work in heterogeneous environments. Furthermore, the software should be portable in between different platforms.

Geographically transparent The method should be transparent to geographical separation.

Maintenance The simulation model maintenance effort should be limited.

Usability Data used to drive the simulation should be generated and maintained in software and through interfaces the users are used to. User interfaces for production plan inputs and database modifications should be simple for fast generation of new experiments.

Simulation speed Each simulation should be possible to run within two hours, say. This will make several iterations with model modifications possible within one day. This is a user requirement, BT Products believes that not more than one day can be spent on this activity each month.

The following sections will present the method and how these requirements have been achieved.

4 USED SOFTWARE

The modeling and simulation tool used is QUEST from DELMIA (DELMIA). The project is managed using CVS (Concurrent Versions System) (Cederqvist 1993) for configuration management. Documentation of the project and the model utilizes previously developed methods for integrated documentation (Randell, Holst, and Bolmsjö 2000; Randell 2000). The database used, MySQL, is a fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL is free software and is licensed under the GNU General Public License (MySQL). Using SQL enables portability between different DBMSs. A SQL DBMS being available on a number of platforms was chosen for portability reasons since the developed tools will be used on both SGI IRIX and MS Windows NT.

5 DEVELOPED SOFTWARE

The database integration is performed through a set of modules. The first module, dubbed QDBC (Quest DataBase Connection), connects the QUEST model to the database. The QDBC module contains server and client routines. The SQL module is used to format queries. The third module contains the routines to run a model driven by a database using the former two modules. A fourth module that is to break down the master plan is to be developed.

The routines are written using abstract data types; i.e. the implementation details are hidden from the simulation model builder, thus enabling exchange of modules without affecting the other modules. This modular approach enables the exchange of, e.g. the QDBC interface to the commercially available Delfoi Integrator (Delfoi) or changing the SQL syntax if another DBMS is used.

5.1 The QDBC Server

The QDBC server is implemented in C and relies on the MySQL C interface and the socket routines supplied with QUEST. The database, host, user, password and port to communicate through are specified when starting the server. The server reads messages on the TCP/IP socket and passes the query to the database. The reply is then sent back to the client together with message identification, message size and the number of rows and columns. Using TCP/IP allows the database and simulation to reside on different computers anywhere on a network.

Table 1: The Database Tables and Fields in the Proof-of-Concept Model

Table	*Fields*
orders	order number, article number, number of items, start date, stop date
operations	article number, resource number, operation number, setup time, load time, cycle time, unload time
articles	article number, article name, standard cost
resources	resource number, resource name, setup time, load time, cycle time, unload time, resource cost

5.2 The QDBC Client Routines

The QDBC client routines are implemented in QUEST SCL (Simulation Control Language). When a query has been sent the routines reads the raw reply and splits it into rows and fields that are stored in a dynamic list data structure. A dynamic list structure is used for scalability, i.e. any number of rows, fields and any size of the fields are allowed. A number of routines are supplied for easy retrieval of individual rows or fields. The QDBC client routines are planned to be rewritten in C/C++ to enable exchange of the simulator.

5.3 The SQL Routines

The SQL routines are supplied to simplify the programming and hide the SQL implementation from the user. When another DBMS is used it is thus simple to change the SQL syntax in the routines without affecting the simulation model. This set of routines is also planned to be rewritten in C/C++ for portability reasons.

6 THE DATABASE

The ERP database is exported as field delimited ASCII files into a “standard” simulation database to facilitate exchange of ERP systems. The simulation database is created and data is loaded using a few SQL commands. In the proof-of-concept model the tables and fields in Table 1 have been used. In the final implementation at BT Products more tables and fields will be added. The simple database can be said to be a miniature ERP system database.

A work calendar and shifts for all resources in the factory will be defined. This is currently not implemented in the ERP system and will thus be defined manually.

7 THE SIMULATION MODEL OBJECTS

7.1 The Orders

The *temporary objects* (Pidd 1992) passing through the simulation model are in this case not parts, but orders. An order contains order information and thus the search keys necessary to retrieve information from the database at any point in the simulation.

As an order passes through the system, the article database is updated, i.e. articles are required, decreasing an article register, and generated, increasing an article register. When all the operations for an order have been completed the order is considered to be completed and the database is updated.

Orders will be generated from a master plan defined by the user. Each product can be made in a number of variants and historic data will be used to set the product variant mix. From the master plan and the structure register, products will be broken down to the article level and production orders will be generated over time. The master plan break down in the ERP system has long lead-time and will therefore be performed in a separate module in the simulation. The order sequence is of importance in order to achieve a balanced flow downstream.

7.2 The Scheduler

The scheduler first determines the first start date from the production plan, i.e. the simulation time origin. Starting from the first start date, each date is stepped through day by day. The scheduler retrieves the orders of the current simulation date from the database and then each order is passed to the first element in the operations list. Thus the material flow has been started for the entire simulation model using only one element. The dates are generated using the built in date routines in the database server, removing the need to explicitly build a calendar routine.

7.3 The Resources

Each resource is of the same class and has the same logic. The difference between instances of a class is the geometric representation, location, shifts and the name.

The general resource process logic retrieves the setup, load, cycle and unload times from the database and executes them before passing the order to the next element. The process logic of the resource class is straightforward as shown in the pseudo code below.

```
require order
curr_op = get_current_op(art_nbr)
work res_setup_time()
for ii = 1 to nbr_of_items do
   work res_load_time()
   work res_cycle_time()
   work res_unload_time()
endfor
```

The route logic is general and is used by all elements driven by the database, including the scheduler. The route logic determines the next resource on the operation list and then routes the order, shown in the pseudo code below.

```
while (num_routed < num_to_route) do
   next_res = get_next_res(art_nbr, \
                           curr_op)
   route order to next_res
   num_routed = num_routed + 1
endwhile
```

7.4 The Terminating Element

The terminating element is common for the entire simulation model. It stores statistics and updates registers before removing the temporary elements from the simulation.

8 BUILDING THE MODEL

The simulation model can either be built from information in a database or manually. From the resource register it is relatively easy to generate a QUEST BCL (Batch Control Language) script that creates the required resources. However e.g. object geometries and their placement on the layout are not stored in the ERP database. Building a simulation model from another representation requires a very high resemblance in between the digital representation of the system and the real system. Moreover, the digital representation is required to be *complete* and *unambiguous*, i.e. it must contain all the information required to build the simulation model and there should be one and only one representation. The resource register will therefore be complemented with the required additional information for a fully automatic generation of the simulation model.

9 RUNNING THE SIMULATION

Generating orders from a manually defined master plan initializes a simulation run. Opening the database connection, creating all elements in the model and creating all element connections necessary initializes the simulation. The connections are created by stepping through the operations table, retrieving the resource of an operation, retrieving the next operations resource and then connecting the resources. This is performed for all operations in the database and is done automatically.

The article stock database can be initialized with the current actual stocks of material to reflect a probable outcome of the next period, referred to as *intelligent initialization* (Banks, Carson, Nelson, and Nicol 2000). The current manufacturing system state is however not available and a warm up period is required. The article stock database can also be set randomly to reflect a general situation.

10 DISCUSSION AND CONCLUSIONS

The case has shown that it is easy to integrate a simulation system with other computer systems. The proof-of-concept model has so far shown that it is feasible to model and simulate entire factories using existing ERP databases and that it can be performed with limited resources. Furthermore, it has been shown that it is relatively simple to generate the software and routines necessary to implement a database driven simulation model. Actually, this is, at the same time, the smallest and largest model built by the authors. The smallest in the sense that the required effort to generate the routines to generate and run a simulation has been limited so far. The largest in the sense that we have never before been able to model and simulate such a large manufacturing system.

It is generally recommended to build simulation models with the stochastic behavior correctly modeled (Banks, Carson, Nelson, and Nicol 2000; Law and Kelton 1991). The presented simulation model is dynamic, but has limited stochastic behavior due to the lack of data for failures and cycle time variations in the ERP database. This is the trade off when using readily available, but not complete, information. However, the simulation accuracy is expected to be better that a static Capacity Requirements Planning (CRP) analysis, but less than the accuracy achieved with traditional DES methods. This is the price paid for the highly reduced modeling time while being able to model large factory models. Still, the simulation is dynamic and interactions are captured.

The demonstrator is currently being extended for real world applications and implemented at BT Products. Several problems are to be solved due to the complexity of a real ERP database. However, when the project is finished most of the routines and software will be generalizable to any enterprise. The method presented is general in that ERP databases contain the same sets of data although being stored differently in another enterprise. By developing a database export interface the simulation database can be made independent of the original ERP database. Furthermore, the method is also platform and DBMS independent. It is expected to be possible to simulate another factory from scratch with small efforts.

Costs for maintenance of the simulation model is anticipated to be reduced considerably compared to traditional methods of building simulation models. Since the simulation data is updated automatically as the ERP database is updated. The simulation model needs to be updated only when resources are modified, removed or added which is easily traced and happen rarely. To add an element to the model, a resource with location, geometry and shift is added to the database. Such an update could be performed within minutes, which guarantees continuous updates.

As pointed out by Iuliano (1995) different versions of the used software tools will create compatibility problems over time, not only in between versions of the same software tool but also in between different integrated software tools. Due to the highly modular design these problems will be minimized.

The simulation model can be designed to handle several different performance measures. Furthermore, the performance measure sampling is easily implemented in all the objects in the simulation model by changing a few general routines. Delivery date, wait times, lead-times and WIP (Work In Progress) are to be used as performance measures. Filters will be added to facilitate fast and easy identification of possible problems in the large amount of output data.

The holistic factory model will be used for operative assessment of production plans; i.e. to verify that the next few months production plan will result in on time deliveries or to calculate delayed deliveries beforehand. These results can be used actively or passively. In the latter case the results can be used to warn the market department about upcoming delayed deliveries. In the active case, the simulation is used to analyze where the bottlenecks are and then, e.g. add extra shifts or operators in those sections. The active countermeasures are then tested in the model to verify the desired results. These actions are performed in a few iterations within a day.

The developed methods for integration of existing ERP databases can also be used for manufacturing system redesign or modifying control strategies. Delayed deliveries will work as a measure of how well a production plan can be fulfilled. Wait times indicates bottlenecks. The differences in between delayed and early deliveries in combination with wait times can be used to trace, not only, what orders are delayed, but also why. The larger the difference between delayed and early deliveries the more reason to modify control strategies. A large number of both early deliveries and delayed deliveries imply that there are synchronization problems, i.e. the early deliveries have been processed at the expense of the delayed deliveries. The simulation model can thus be used to increase the order promise accuracy, maximize utilization of production capacity and to modify and test control strategies before implementation.

The simulation database is a copy of the real database not to affect the daily operations. By running the simulation off-line the database can be updated from the simulation and modified to simulate how, e.g. an investment, would affect performance measures without affecting the daily operations.

ACKNOWLEDGMENTS

We would like to thank BT Products for making this project possible. One thing is to develop a proof-of-concept model, it is quite another thing to make a real implementation, and BT Products has made that possible through their efforts.

REFERENCES

Al-Dabass, D., and R. Cheng. (Eds.) 1999. *UKSIM'99. Fourth National Conference of the UK Simulation Society*. Nottingham Trent Univ.

Andradóttir, S., K. J. Healy, D. H. Withers, and B. L. Nelson. (Eds.) 1997. *Proceedings of the 1997 Winter Simulation Conference*. Winter Simulation Conf. Board of Directors.

Ball, P. D., and D. M. Love. 1994. Expanding the capabilities of manufacturing simulators through the application of object-oriented principles. *Journal of Manufacturing Systems* 13 (6): 412–23.

Banks, J., J. S. Carson, B. L. Nelson, and D. Nicol. 2000. *Discrete-Event System Simulation*. 3rd ed. Upper Saddle River, New Jersey 07458: Prentice-Hall, Inc.

Bernard, B. 2000. *Integration of manufacturing simulation tools with information sources*. Licentiate Thesis, Computer Systems for Design and Manufacturing Division, Department of Manufacturing Systems, Royal Institute of Technology, Stockholm, Sweden.

Bolmsjö, G., U. Lorentzon, and L. Randell. 1999, May. Integration of development of product and production system using advanced simulation tools. See Brussel, Kruth, and Lauwers (1999), 389–98.

Brussel, H. V., J.-P. Kruth, and B. Lauwers. (Eds.) 1999, May. *The 32nd CIRP International Seminar on Manufacturing Systems — New Supporting Tools for Designing Products and Production Systems*. Katholieke Universiteit Leuven - Departement Werktuigkunde.

Cederqvist, P. 1993. *Version management with CVS: for CVS 1.10*. Box 2044, 580 02 Linköping, Sweden: Signum Support AB.

Charnes, J. M., D. J. Morrice, D. T. Brunner, and J. J. Swain. (Eds.) 1996. *Proceedings of the 1996 Winter Simulation Conference*. SCS Int.

Delen, D., P. C. Benjamin, and M. Erraguntla. 1998. Integrated modeling and analysis generator environment (IMAGE): a decision support tool. In *Proceedings of the 1998 Winter Simulation Conference*, ed. P. Benjamin, M. Erraguntla, and D. D. R. Mayer, 1401–8: IEEE.

Delen, D., P. C. Benjamin, and M. Erraguntla. 1999, December. An integrated toolkit for enterprise modeling and analysis. See Farrington, Black Nembhard, Sturrock, and Evans (1999), 289–97.

Delfoi. Available online from `http://www.delfoi.com`, [accessed February 3 2001].

DELMIA. Available online from `http://www.delmia.com`, [accessed December 21 2000].

Drake, G. R., and J. S. Smith. 1996. Simulation system for real-time planning, scheduling, and control. See Charnes, Morrice, Brunner, and Swain (1996), 1083–90.

Eatock, J., A. Serrano, G. M. Giaglis, and R. J. Paul. 1999. A case study on integrating business and network simulation for business process redesign. See Al-Dabass and Cheng (1999), 114–18.

Ellis, K., A. Jones, and T. Lee. 1998. Requirements analysis: Process plan specification - workstation level. Technical Report NISTIR 6172, National Institute of Standards and Technology.

Farrington, P. A., H. Black Nembhard, D. T. Sturrock, and G. W. Evans. (Eds.) 1999, December. *Proceedings of the 1999 Winter Simulation Conference*, Phoenix, Arizona, USA. IEEE.

Giaglis, G. M. 1999. Integrated simulation of business processes and information systems. See Al-Dabass and Cheng (1999), 119–25.

Gmilkowsky, P., F. Eckardt, and D. Palleduhn. 1997. Concept of an integrated system for modeling, simulation and analysis of discrete production processes. In *Proceedings of the 1997 Summer Computer Simulation Conference Simulation and Modeling Technology for the Twenty-First Century*, ed. M. S. Obaidat, 329–34: SCSI.

Gmilkowsky, P., F. Eckardt, and D. Palleduhn. 1998. Automated simulation model generation: a knowledge-based approach within an integrated system for modeling, simulation and analysis of discrete production processes. In *Proceedings of the Simulators International XV*, ed. R. Ades, M.; Griebenow, 157–62: Society for Computer Simulation.

Heim, J. A. 1997. Integrating distributed simulation objects. See Andradóttir, Healy, Withers, and Nelson (1997), 532–8.

Holst, L., L. Randell, and G. Bolmsjö. 2000, June. Integrated development of manufacturing systems using simulation — Proposing the fundamentals for a joint research project. In *The 33rd CIRP International Seminar on Manufacturing Systems*. Stockholm.

Iuliano, M. 1995. Overview of the manufacturing engineering toolkit prototype. Technical Report NISTIR 5730, National Institute of Standards and Technology.

Iuliano, M. 1997. The role of product data management in the manufacturing engineering toolkit. Technical Report NISTIR 6042, National Institute of Standards and Technology.

Iuliano, M., and A. Jones. 1996. Controlling activities in a virtual manufacturing cell. See Charnes, Morrice, Brunner, and Swain (1996), 1062–7.

Iuliano, M., A. Jones, and S. Feng. 1997. An analysis of AP213 for usage as a process plan exchange format. Technical Report NISTIR 5992, National Institute of Standards and Technology.

Johansson, M. 2001. *Information management for manufacturing system development*. Ph. D. thesis, Division of Computer Systems for Design and Manufacturing, Department of Production Engineering, Royal Institute of Technology, SE-100 44 Stockholm, Sweden.

Judd, R. P., and J. Abell. 1996, July. Off-line development of factory control code using simulation. In *Proc. of the 1996 Summer Computer Simulation Conference*, 92–96.

Law, A. M., and D. W. Kelton. 1991. *Simulation modeling and analysis*. 2nd ed. Industrial Engineering and Management Science. McGraw-Hill Inc.

Liyanage, K., and T. Perera. 1998. Design and development of a rapid data collection methodology. In *International Conference on Simulation '98*, 297–304: IEE.

Love, D. M., S. R. Clarke, and D. I. Gooden. 1987. The use of simulation to determine operational policies in an MRP/FMS environment. In *Second International Conference on Computer-Aided Production Engineering*, ed. J. A. McGeough, 261–6. Bury St. Edmunds, UK: Mech. Eng. Publications.

MySQL. Available online from `http://www.mysql.com`, [accessed March 22 2001].

Peters, B. A., J. S. Smith, J. Curry, C. LaJimodiere, and G. R. Drake. 1996. Advanced tutorial-simulation-based scheduling and control. See Charnes, Morrice, Brunner, and Swain (1996), 194–8.

Pidd, M. 1992, January. *Computer simulation in management science*. 3rd ed. Chichester, England: John Wiley & Sons Ltd.

Randell, L. 2000. *A methodology to reduce time consumption in discrete-event simulation projects*. Licentiate Thesis, Lund University, Department of Mechanical Engineering.

Randell, L., L. Holst, and G. Bolmsjö. 1999a, September. Concurrent development of large discrete-event simulation models. In *Proceedings of the Second World Manufacturing Congress, WMC'99*, ed. S. Hahavandi and M. Saadat, 77–83. Durham, United Kingdom: International Computer Science Conventions: ICSC Academic Press.

Randell, L., L. Holst, and G. Bolmsjö. 1999b, December. Incremental system development of large discrete-event simulation models. See Farrington, Black Nembhard, Sturrock, and Evans (1999), 561–8.

Randell, L., L. Holst, and G. Bolmsjö. 2000, June. Documenting discrete-event simulation models. In *Proceedings of INFORMS-KORMS Seoul 2000 Conference*, 1483–90. INFORMS/KORMS.

Smith, J. S., and B. A. Peters. 1998. Simulation as a decision-making tool for real-time control of flexible manufacturing systems. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, 586–90 vol.1: IEEE.

Smith, J. S., R. A. Wysk, D. T. Sturrock, S. E. Ramaswamy, G. D. Smith, and S. B. Joshi. 1994. Discrete event simulation for shop floor control. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 962–9: IEEE.

Srinivasan, K., and S. Jayaraman. 1997. Integration of simulation with enterprise models. See Andradóttir, Healy, Withers, and Nelson (1997), 1352–6.

Trybula, W. 1994. Building simulation models without data. In *1994 IEEE International Conference on Systems, Man, and Cybernetics. Humans, Information and Technology*, Volume 1, 209–14: IEEE.

Umeda, S., and A. Jones. 1997, July. Simulation in Japan: State-of-the-art update. Technical Report NISTIR 6040, National Institute of Standards and Technology, U.S. Department of Commerce, Technology Administration.

Whitman, L., B. Huff, and A. Presley. 1997. Structured models and dynamic systems analysis: the integration of the IDEF0/IDEF3 modeling methods and discrete event simulation. See Andradóttir, Healy, Withers, and Nelson (1997), 518–24.

Zhang, J., and J. Browne. 1999, May. New supporting tools for designing production systems. See Brussel, Kruth, and Lauwers (1999), 23–32.

AUTHOR BIOGRAPHIES

LARS RANDELL is a PhD student at the Department of Mechanical Engineering at Lund University. He received a MSc in mechanical engineering from Lund University 1996. His email and web addresses are <lrandell@robotics.lu.se> and <www.robotics.lu.se>.

GUNNAR BOLMSJÖ received the MSc and PhD degree in Mechanical Engineering from Lund University in 1981 and 1986 respectively. From 1986 he was a research associate at the Dept. of Mechanical Engineering, Lund University, and is since 1987 Professor in Robotics at the same department. He is a member of IEEE, SIRES, AAATE and JOM. His email and web addresses are <gbolmsjo@robotics.lu.se> and <www.robotics.lu.se>.