Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

SIMULATION ENVIRONMENT FOR ELECTRONICS MANUFACTURING

Phillip A. Farrington, John S. Rogers,
Bernard J. Schroer, James J. Swain

University of Alabama in Huntsville
Huntsville, Alabama 35899, U.S.A.

John L. Evans

Chrysler Electronics
Huntsville, Alabama 35824, U.S.A.

Yun-cheng Feng

Beijing University of Aeronautics and Astronautics
Beijing, P.R.C.

ABSTRACT

This paper presents the development of an environment to simplify the creation of simulation models for high-volume electronics manufacturing systems. The simulation environment consists of a problem definer, a static analyzer and a code generator. Included in this paper is an overview of the environment, an application of the system at Chrysler Electronics, and a discussion of system advantages.

1 INTRODUCTION

The popularity of simulation, and the number of commercially available simulation packages has continued to grow in recent years. A recent survey of discrete event simulation packages and related products lists over 50 such products (Swain 1995) for personal computers. The range and capability of these products has grown as well, so that graphical interfaces, animated output, and models of almost unlimited size can be built with programs that run on desktop computers. Simulation is increasingly accessible to industry, and it is seen both as a way to validate designs and as a way to insure against costly mistakes (Church 1982 and Kay 1984).

For all of their power and capability, a review of large simulation projects indicates that problems still exist: model development tends to be labor intensive and error prone. As a result, model development time is often underestimated. A contributing cause is poor communication between the modeler and the user, with the modeler often left in a void with little user feedback or support. In addition, models cannot be readily reused or effectively utilized when modifications are desired. Final cost of development is generally higher than budgeted (Roth, Gass, and Lenoine 1978), and that, together with time overruns, often means that the model documentation is incomplete, inconsistent, inadequate, or in many cases, nonexistent.

A number of approaches are being used to address these problems. Simulators and domain specific simulation front-ends are being built to speed model development and make model building more accessible to non-simulationists, by directly representing the typical items of a given domain. In addition, a number of simulation languages are being organized hierarchically, with the simulation elements at a low level, and objects organized in a hierarchy of increasingly complex element groups that serve particular functions.

2 PREVIOUS RESEARCH

There is considerable interest in improving simulation, both to speed model development and to widen the attractiveness of simulation modeling in general. Henriksen (1983) discussed a number of issues related to how simulation languages could be improved; a number of which are addressed in the design of SLX, a hierarchical successor to GPSS/H (Henriksen 1993). Standridge, Vaughan, and Sale (1985) and Centeno and Standridge (1993) have developed the integration of database management techniques for each stage of the simulation model development process. Balci (1986) describes the requirements for a general model development environment, while Hlupic and Paul (1994) detail ongoing work with computer aided simulation modeling.

In recent years several simulators have emerged on the commercial market, such as SIMFACTORY, ProModel, Witness, and XCell, a number of which are examined in Mathewson (1989). In addition, considerable research has been conducted on the application of front-ends and simulators. One of the earliest approaches to expediting model development was a natural language interface by Heidron (1974). Ford and Schroer (1987) developed the prototype Electronics Manufacturing Simulation System (EMSS) for modeling electronics assembly. Brazier and Shannon (1987) have developed an environment for modeling automated guided vehicle systems (AGVS). The system uses an interactive user dialogue to define the AGVS and generate SIMAN code. Ozdemirel and Mackulak (1993) have developed a flexible manufacturing system front-end which generates SIMAN code. Endesfelder and Tempelmeier (1987) have developed a model processor that generates SIMAN models based on a library of predefined modules. These modules are for modeling a flexible manufacturing system and include load/unload stations, machining centers, tool storage, and pallet storage. The Automatic Manufacturing Programming System (Schroer, et al. 1988) uses an interactive dialogue to assist the user define the problem. The system writes GPSS/PC models. Haddock and Davis (1985) have developed a flexible manufacturing system simulation generator. Murray and Sheppard (1988) have developed a knowledge based model construction system to automate model definition and code generation. Other research into code generators include Mathewson (1984), Khoshnevis and Chen (1986), and Oren and Zeigler (1979). Mize, et al. (1992) have outlined an object oriented approach to modeling manufacturing systems. Armacost, Mullens, and Swart (1994) developed a housing manufacturing simulation environment using a ProModel and a relational data base management system. A review of new commercial products such as ARENA and SLX finds that they contain many of the features outlined in the cited research.

3 APPLICATION DOMAIN

The Chrysler Huntsville Electronics Division (HED), located in Huntsville, Alabama, is a large electronics manufacturing facility which supplies boards and controllers for use in Chrysler products. The 550,000 square foot facility contains fifteen assembly lines, with an approximate employment of 2,300 and annual revenues of $900 million. The HED facility is typical of Huntsville's electronics manufacturing capabilities which includes such companies as SCI, AVEX, Intergraph, and Motorola.

The HED facility is typical of high volume electronics manufacturing. Their products and facilities are undergoing continual changes and improvements. Planning is particularly difficult for new products, since planning proceeds concurrently with the design itself, and during this cycle there may be changes to materials, methods of placement, density of components, precision of the placements, and the equipment that will be used to realize these changes. Finally, flexibility in planning is required between disparate corporate functions, no one of which is in complete control of the process.

4 SYSTEM OVERVIEIW

Figure 1 provides an overview of the simulation environment being developed at UAH to rapidly model and evaluate high volume electronics manufacturing systems. This environment consists of three basic elements: problem definer, static analyzer, and model generator. Each element will be interconnected through data transfer links and feedback loops. The three elements provide increasing levels of modeling capability, while aiding simulation model development.

4.1 Problem Definer

The problem definer is used to develop the initial definition of an electronics assembly line, through the use of a graphical interface. Individual line elements are defined by selecting the appropriate icon from a master template of elements. A new element of the type selected then appears in the center of the displayed area of the line layout window. The element is then placed by using the mouse to drag the displayed icon to the correct location. Components are defined for line entry and exit points, assembly stations, buffers, inspection stations, ovens, conveyors, as well as line convergence and divergence points. Table 1 lists a subset of the manufacturing elements located at the Chrysler facility and contained in the model database. The icon template and a partial line definition are shown in Figure 2.

The attributes of individual manufacturing elements and the production line process flow are defined through component definition tables. The user interface is designed to facilitate data collection and to be as intuitive and user-friendly as possible. The environment is linked to an external database which contains lists of the specific elements available for the simulation for each type of manufacturing component.

Table 2 lists the input information required for each modeling element. Drop down menus are used to enter information for fields where multiple choice selections exist. Data type checking and internal unit conversions are integrated into the system to provide an easy mechanism for numerical entries while maintaining data consistency. Trend information for failures, repairs, downtimes, and production throughputs can be entered as specific constants or as statistical distributions. A sample input screen for an inspection station is shown in Figure 3.

Table 1: Subset of Chrysler Manufacturing Line Components

Component	Component Type
Board Inverter	Assembly
Curing Oven	Oven
Final Test	Inspection
Fine Pitch Placement Machine	Assembly
Functional Test Oven	Oven
General Conveyor	Conveyor
General Purpose/Vertical Stacker	Buffer
Glue Placement Machine	Assembly
In-Circuit Test	Inspection
Phillips FCM16 Placement Machine	Assembly
Reflow Oven	Oven
Solder Paste Application/Screen Printing	Assembly
Solder Paste Inspection	Inspection
TDK Bar Code Applicator	Assembly
TDK Bar Code Verification	Inspection
TDK CX-5 Placement Machine	Assembly
TDK RX-5 Placement Machine	Assembly
X-Ray Inspection	Inspection

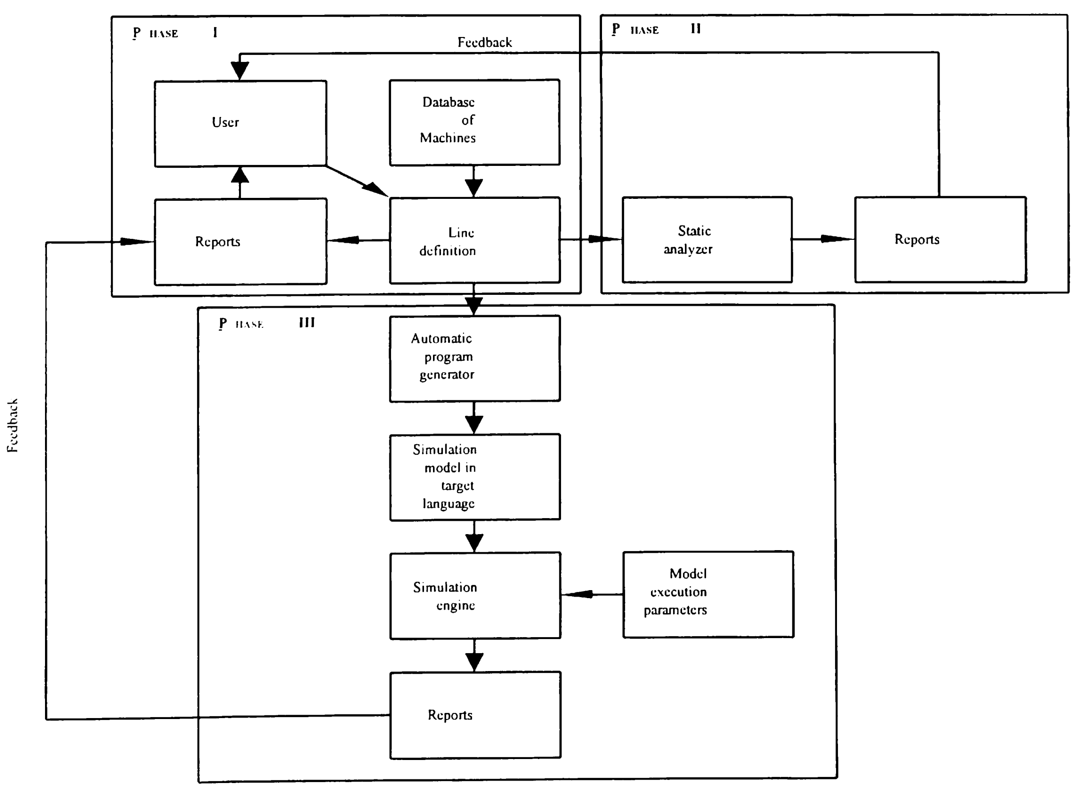

Figure 1: Simulation Environment System Overview

Table 2: Input Requirements for Modeling Elements

Component Type	Enter Node(s)	Exit Node(s)	Throughput Rate	Downtime/ Repair Frequency	Speed and Length	Other
Start		x	x			
Assembly	x	x	x	x		
Buffer	x	x	x	x		Capacity, Index Scheme
Conveyor	x	x		x	x	
Inspection	x	x	x	x		Failure Following
Oven	x	x		x	x	Capacity
Convergence	x	x			x	Number of Convergence's
Divergence	x	x			x	No. of Divergence's
Stop	x					

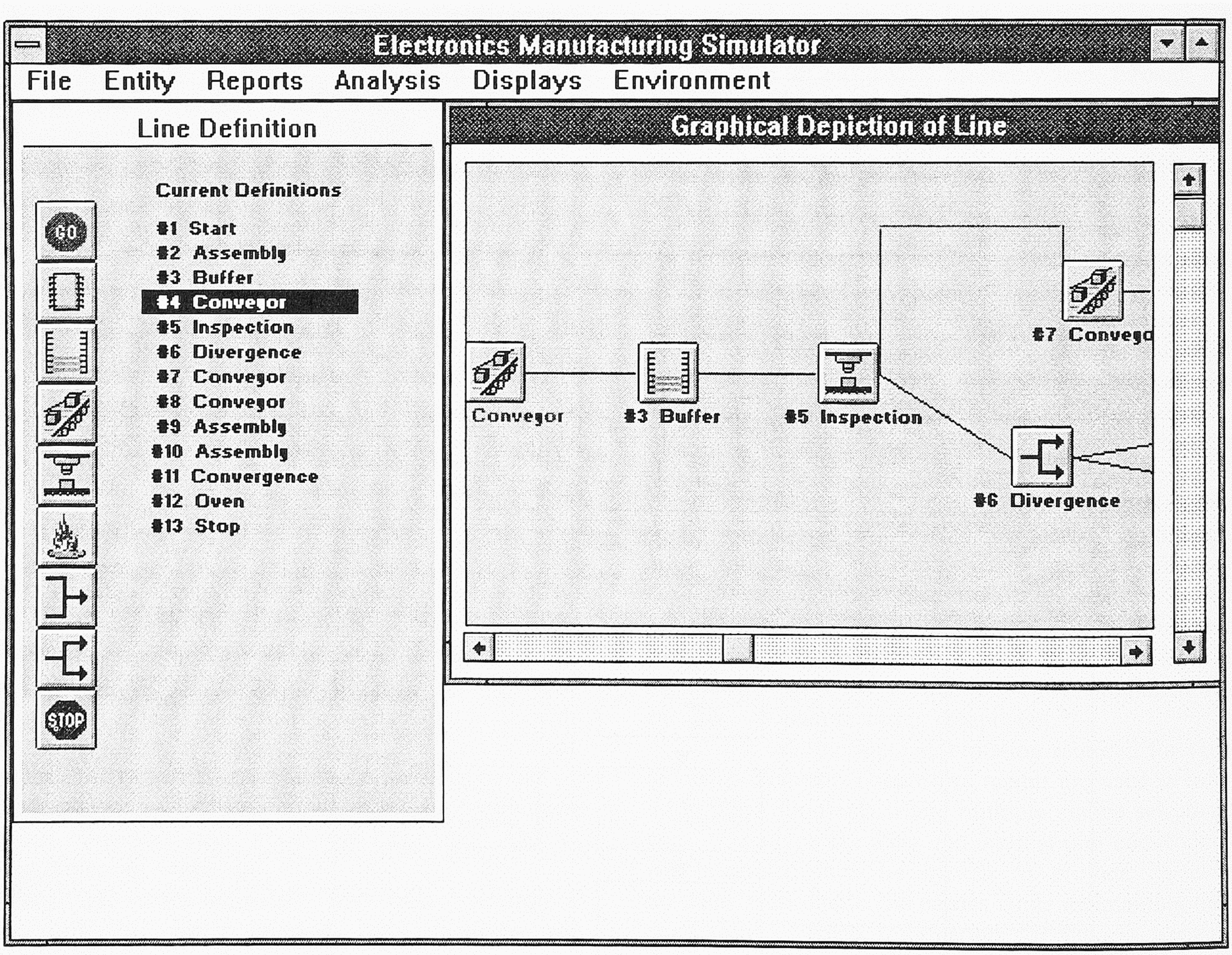

Figure 2: Graphical Layout Tool and Icon Template

4.2 Static Analyzer

The static analyzer performs a static analysis on each component of the manufacturing line model. The analyzer utilizes the component processing times to generate a maximum throughput for each component of the line for a specified time period. Individual processing times are defined as constants, distributions, or as the sum of the processing and indexing times for subcomponent placement. The system internally converts the input parameters to minutes to assure consistency in calculations and comparisons; however, output displays retain the units that were originally entered by the user.

Bottleneck locations for non-branching model segments can be identified by locating the elements within the segment with the smallest throughput values. At convergence points, however, two or more line throughputs are combined and become the input to the downstream elements meaning that the bottleneck may not be at the element with the smallest throughput value. In a similar manner the upstream elements of a divergence point must handle the summed throughputs of the downstream sublines. Each subline component of the convergent system (shown in Figure 4) must be considered separately.

Likewise, inspection stations represent divergence points within the line because of the possibility of the manufactured items not successfully passing inspection and being routed back for rework, further complicating the throughput analysis. Figure 5 depicts such a rework loop or circuit. In this case, the input to the assembly station following the intersection point on the line consists of the newly manufactured parts plus reworked previous failures. Therefore, the station can represent a bottleneck although the original throughput number appears higher than other stations on the line. At steady-state the rework volume, R, can be determined as the product of the probability of a manufactured item failing at the inspection station, P(Fail), and the smallest throughput value of a station in the main path of the inspection circuit, C':

$$R = P(Fail)C'$$

The maximum number of new manufactured items that can enter the circuit, N, is calculated as the difference between the minimum circuit throughput value and the rework number:

$$N = C' - R$$

The effects of the limitation of input values and output for each line element is propagated throughout the system and presented in the output reports.

4.3 Reporting Features

The manufacturing model environment has two basic types of output reports: input information and static analysis. Input information can be displayed in a summary or by individual line element.

A static analysis summary can be displayed in chronological order based on element creation or as a sorted listing of the elements based on the analysis results. Static analysis information consists of the component name, maximum throughput possible for each line element, and the expected steady state throughput.

4.4 Code Generator

A feature currently under development is simulation code generation. The code generation capability will consist of modules which produce input files in required formats for the simulation packages currently in use at Chrysler and UAH, including GPSS, ARENA, and WITNESS. The model files generated will be used as input definition files for the simulations. This will require the definition and development of a generic set of line elements which can be used to create models for several simulation packages.

5 CONCLUSIONS

The development effort is currently progressing down two parallel paths, the development of the line definition program in Visual Basic, and the development and validation of distinct modeling elements in WITNESS and ARENA. The current goal is to establish capabilities for rapid model development in the two primary simulation systems (i.e., WITNESS and ARENA) and in the future provide linking mechanism to automatically transfer data captured in the Visual Basic line definition program. These parallel development paths also allow us to determine the most appropriate set of modeling elements to include in a planned custom simulator extension for the Visual Basic line definition program.

Based on needs observed at Chrysler and other such facilities it is felt that this system such as this provides a number of distinct advantages, including:

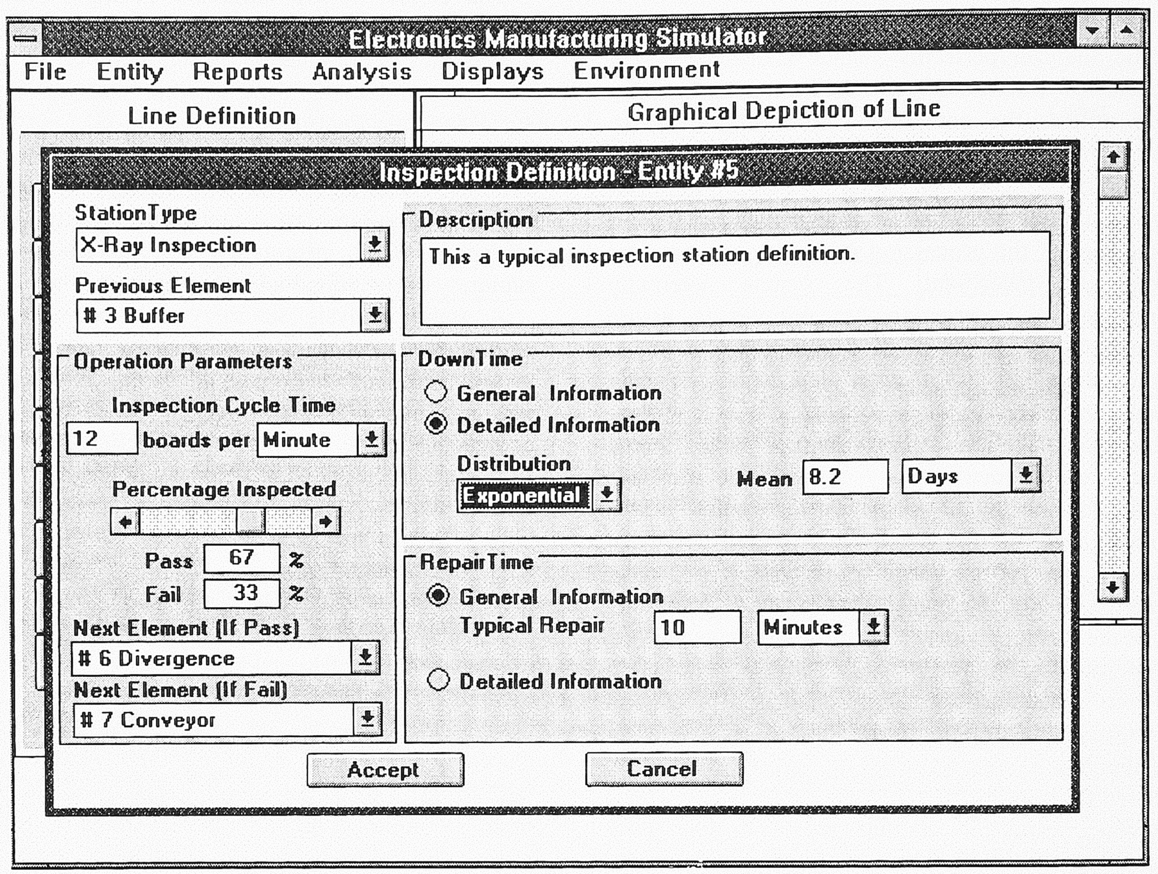

Figure 3: Assembly Station Input Template

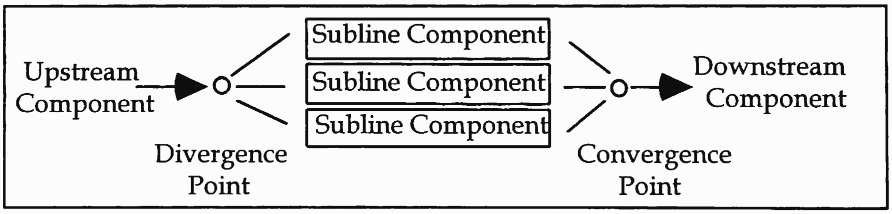

Figure 4: Divergence and Convergence within the Manufacturing Line

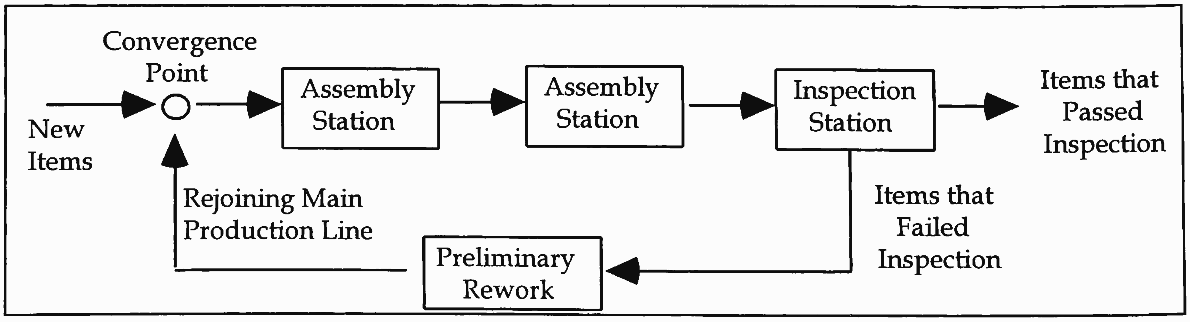

Figure 5: Inspection Rework Circuit Figure

• Rapid prototyping. The system will produce executable simulation code that is error free.

• Increased productivity. A significant increase in modeler productivity should occur which can be measured in lines of simulation code written per hour.

• Easier maintenance. The simulation code that is generated should be easier to modify.

• Improved documentation. If model changes are made using the user interface, the system configuration is constantly updated.

• Domain specific. Model developers are presented with modeling elements that are representative of real equipment from that domain. This lowers the level of user abstraction required to model the system.

ACKNOWLEDGMENTS

This project is being funded in part by the National Science Foundation, Chrysler-Huntsville Electronics Division, and Alabama Industrial Development Training (AIDT).

REFERENCES

Armacost, R., M. Mullens, and W. Swart. 1994. Structuring a Simulation Modeling Environment Using a Commercial Manufacturing Simulator. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 935-940. Institute of Electrical and Electronics Engineers, Orlando, California.

Balci, O. 1986. Requirements for Model Development Environment. *Computers and Operations Research* 13(1): 53-57.

Brazier, M. and R. Shannon. 1987. Automatic Programming of AGVS Simulation Models. In *Proceedings of the 1987 Winter Simulation Conference*, ed. A. Thesen, H. Grant, and W.D. Kelton, 703-708. Institute of Electrical and Electronics Engineers, Atlanta, Georgia.

Centeno, M.A., and C.R. Standridge. 1993. Databases: Designing and Developing Integrated Simulation Modeling Environments. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G.W. Evans, M. Mollagasemi, E.C. Russell, and W.E. Biles, 526-534. Institute of Electrical and Electronics Engineers, Los Angeles, California.

Church, J. 1982. Simulation Aspects of Flexibly Manufacturing Systems Design and Analysis. In *Proceedings 1982 Fall Industrial Engineering Conference*, 426-431. Atlanta, Georgia.

Endesfelder, T., and H. Tempelmeier. 1987. A Flexible Software Tool for the Generation of SIMAN Simulation Models. In *Proceedings of the European Simulation Multiconference.* Vienna, Austria.

Ford, D., and B. Schroer. 1987. An Expert Manufacturing Simulation System. *Simulation* 48(5): 193-200.

Haddock, J., and R. Davis. 1985. Building a Simulation Generator for Manufacturing Cell Design and Control. In *Annual International Industrial Engineering Spring Conference Proceedings*, 237-244. Los Angeles, California.

Heidron, G. 1974. English is a Very High Level Language for Simulation Programming. *SIGPLAN Notices* 9(4): 91-100.

Henriksen, J. 1983. The Integrated Simulation Environment (Simulation Software of the 1990's). *Operations Research* 31: 1053-1073.

Henriksen, J. 1983. SLX, The Successor to GPSS/H. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G.W. Evans, M. Mollagasemi, E.C. Russell, and W.E. Biles, 263-268. Institute of Electrical and Electronics Engineers, Los Angeles, California.

Hlupic, V., and R.J. Paul. 1994. The CASM Environment Revisited. In Proceedings of the 1994 Winter Simulation Conference. ed. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 641-648. Institute of Electrical and Electronics Engineers, Orlando, Florida.

Kay, J. 1984. The Use of Modeling and Simulation Techniques in the Design of Manufacturing Systems. In *International Conference on Development of Flexible Automation Systems*, 55-61. New York, New York.

Khoshnevis, B., and A. Chen. 1986. An Expert Simulation Model Builder. *Intelligent Simulation Environment* 17(1): 129-132. San Diego: Society for Computer Simulation.

Mathewson, S.C. 1984. The Application of Program Generator Software and its Extension to Discrete Event Simulation Modeling. *IIE Transaction* 16(1): 3-18.

Mathewson, S.C. 1989. Simulation Support Environment. In *Computer Modelling for Discrete Simulation*. ed., M. Pidd, 57-100. New York: John Wiley and Sons.

Mize, J., H.C. Bhuskute, D.B. Pratt, and M. Kamath. 1992. Modeling of Integrated Manufacturing Systems Using an Object Oriented Approach. *IIE Transactions* 24(3): 14-26.

Murray, K., and S. Sheppard. 1988. Knowledge Based Simulation Model Specifications. *Simulation* 50(3): 112-119.

Oren, T., and B. Zeigler. 1979. Concepts for Advanced Simulation Methodologies. *Simulation* 32(3): 62-82.

Ozdemirel, N., and G. Mackulak. 1993. A generic Simulation Module Architecture Based on Cluttering Group Technology Model Coding. *Simulation* (6): 412-433.

Roth, R., S. Gass, and A. Lenoine. 1978. Some Considerations for Improving Federal Modeling. In *Proceedings of the Winter Simulation Conference*, ed. H.J. Highland, L. G. Hull, and N.R. Nielsen, 213-217. Institute of Electrical and Electronics Engineers, Miami Beach, Florida.

Schroer, B., F. Tseng, S. Zhang, and J. Wolfsberger. 1988. Automatic Programming of Manufacturing Simulation Models. In *Proceedings of the 1988 Summer Computer Simulation Conference*, 569-574. Seattle, Washington.

Standridge, C., D. Vaughan, and M. Sale. 1985. A Tutorial on TESS. The Extended Simulation System. In *Proceedings of 1985 Winter Simulation Conference*, ed. D.T. Gantz, U.W. Pooch, and S.L. Solomon. Institute of Electrical and Electronics Engineers, 73-79. San Francisco, California.

Swain, J.J. 1995. Simulation Survey: Tools for Process Understanding and Improvement. *OR/MS Today* 22(4): 64-79

AUTHOR BIOGRAPHIES

PHILLIP A. FARRINGTON is an Assistant Professor of Industrial and Systems Engineering at the University of Alabama in Huntsville. He holds a Ph.D. from Oklahoma State University in Industrial Engineering and Management from Oklahoma State University. He is a member of IIE, SME, ASQC, Tau Beta Pi, and Alpha Pi Mu. His research interests include system simulation, manufacturing systems design, and integrated product and process design.

JOHN S. ROGERS is a Senior Research Associate with Center for Automation and Robotics at the University of Alabama in Huntsville. He is currently completing work toward a Ph.D. in Mechanical and Aerospace Engineering at UAH. He is a member of ASME and AAAI.

BERNARD J. SCHROER is Associate Vice President for Research and Professor of Industrial and Systems Engineering at the University of Alabama in Huntsville. He has a Ph.D. in Industrial Engineering and Management from Oklahoma State University. He is a member of SCS, NSPE, the Technology Transfer Society, and a Fellow of IIE.

JAMES J. SWAIN is Associate Professor of Industrial and Systems Engineering at the University of Alabama in Huntsville. He has a Ph.D. in Industrial and Systems Engineering from Purdue University. His research interests include Monte Carlo methods and variance reduction, statistical modeling, and output analysis. He has been active in the WSC and INFORMS College on Simulation and is program chair for WSC 1996.

JOHN L. EVANS is an Advanced Technology Coordinator at the Chrysler's Huntsville Electronics Division. He holds a Ph.D. in Industrial and Systems Engineering from the University of Alabama in Huntsville. He is a member of ISHM and Alpha Pi Mu.

YUN-CHENG FENG is a Professor in the School of Management at the Beijing University of Aeronautics and Astronautics. His research interests are neural networks, expert systems, and simulation.