Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

STRUCTURING A SIMULATION MODELING ENVIRONMENT USING A COMMERCIAL MANUFACTURING SIMULATOR

R. Armacost, M. Mullens and W. Swart

Department of Industrial Engineering and Management Systems,
University of Central Florida, P.O. Box 162450, Orlando, Florida 32816-2450, USA

ABSTRACT

The objective of this paper is to describe a new structure which is being used for the development of a simulation modeling environment (SME) for industrialized housing. The structure is centered around a commercial manufacturing simulator which is closely coupled with a relational data base management system (RDBMS). The paper presents an overview of previous SME research, describes the new SME structure and, finally, presents findings from on-going SME development efforts.

1. INTRODUCTION

A major barrier to the wide use of simulation for analyzing real-world manufacturing systems is the lack of access by non-modeling specialists (Mize, et al 1992). Modeling is usually limited to a few experts who have spent a great deal of time learning about the model and how it works (Youngblood 1991). Even for experts, the modeling task is non-trivial and often takes many man-months of effort. The development of an improved simulation modeling environment (SME) has been proposed as a potential solution to these problems. An SME is a software tool which provides support to the user throughout the simulation project life cycle which includes: conceptualization/abstraction, data collection, model development, model verification and validation, output analysis and decision making (Ozdemirel and Mackulak 1993, Treu 1988). An SME should keep a user thinking about the problem to be solved rather than the mechanics of solving it (Centeno and Standridge 1993).

The objective of this paper is to describe a new structure which is being used for SME development. The structure is driven by two factors: 1) the need to develop an SME which parallels and supports the business/engineering decision-making processes for the intended domain and 2) recent developments in commercial manufacturing simulators and multi-media, relational database management systems (RDBMS). The paper presents an overview of previous SME research, describes the new SME structure and, finally, presents findings from on-going SME development efforts.

2. PREVIOUS RESEARCH

SMEs are widely used in the form of commercial manufacturing simulators such as ARENA, FACTOR/AIM, PROMODEL, SIMFACTORY and WITNESS. These packages provide features which greatly enhance a less-experienced user's modeling power and efficiency. Features include high-level manufacturing and simulation constructs accessible through a visual user interface. The interface utilizes object-oriented CAD-like factory design windows, icon-based, point-and-click mouse commands, menu-driven data entry and animation of simulated factory operation. Using these tools, simple models can be developed without programming. While research suggests wide use of these simulators, it finds them lacking in both flexibility and accuracy (Najmi and Stein 1989, Drolet 1992). To address this concern, most commercial manufacturing simulators now provide general purpose programming capabilities which allow the user to add detail to the model. Software vendors who provide this capability report that coding now represents less than 5% of the model development effort. While extending the flexibility of the simulator, this approach once again requires a more experienced user who is skilled in the nuances of the language and capable of software coding.

Related SME research has been varied. Ozdemirel and Mackulak (1993) classify the research in five categories: 1) program generators, 2) hierarchical modular model development, 3) object-oriented simulation, 4) production rule-based modeling, and 5) intelligent user interfaces. The most common approach in the literature is that of intelligent user interfaces. An SME designed around an intelligent user interface has four primary components: the user, a simulation

engine, a user interface and a database (Treu 1988). The simulation engine contains the actual simulation computer code. It executes the simulation and generates simulation results. The interface provides the link between the user and the simulation engine. The interface maintains and uses a database of permanent domain specific information, user-supplied information and raw output data from the simulation engine. This structure is frequently implemented by developing a library of high level modules, each module representing a domain-specific construct. The interface is then developed to assist the user in: 1) selecting constructs, 2) refining each selected construct by specifying key construct parameters and 3) linking constructs to form the specific, executable simulation model. Intelligent user interfaces have been characterized by the use of a conventional general purpose simulation language for the simulation engine, with an interface developed using a general purpose programming language or an artificial intelligence (AI) language. Examples include (for the simulation engine - interface respectively): a general manufacturing simulator using SIMAN - TURBO PROLOG (Ozdemirel and Mackulak 1993), a flexible manufacturing system (FMS) simulator using SIMAN - BASIC (Haddock 1988), a general manufacturing simulator using GPSS/PC - COMMON LISP (Ford and Schroer 1987), a general manufacturing simulator using GPSS/PC - TURBO PASCAL (Schroer 1989), and a robotic manufacturing cell simulator using Q-GERT - FORTRAN (Medeiros and Sadowski 1983).

3. A NEW STRUCTURE

The new structure introduced in this paper is an extension of the intelligent user interface SME research discussed previously. The need for a new structure is driven by two factors: 1) the need to develop an SME which parallels and supports the business/engineering decision-making processes for the intended domain and 2) recent developments in commercial manufacturing simulators and multi-media, relational database management systems (RDBMS). Treu (1988) found that a good interface design does not just happen. It must be carefully and systematically tailored to the class of user for a particular application and within a given system environment. The application domain of this research is industrialized housing. The user is defined to be an expert in the business/engineering side of manufacturing and proficient in PC skills, but not necessarily an expert in simulation modeling. The system environment is PC compatible systems.

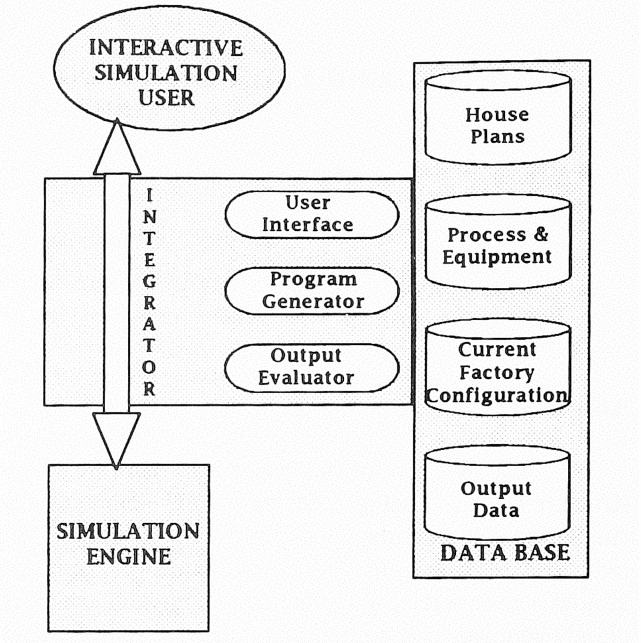

Figure 1. GIHMS Structure

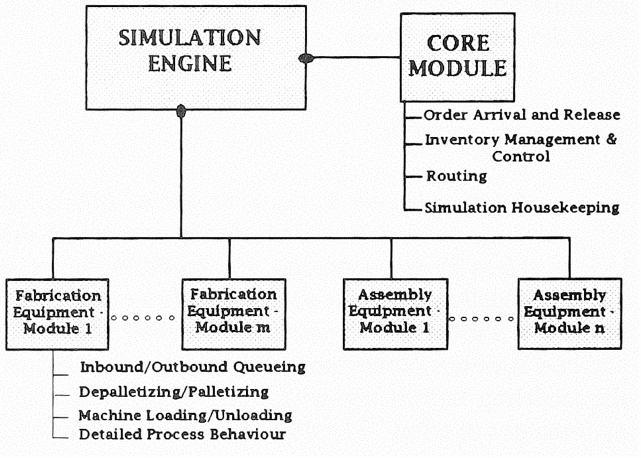

Figure 2. GIHMS Simulation Engine Structure

The structure of the Generic Industrialized Housing Manufacturing Simulator (GIHMS) is shown in Figure 1. The simulation engine is written using the PROMODEL (ProModel Corp. 1993) manufacturing simulator which runs in the PC WINDOWS™ environment. This approach not only shortened the simulation engine coding effort, it allowed a number of PROMODEL's visual interface windows to be incorporated directly into the GIHMS user interface. The structure of the simulation engine (Figure 2) is driven largely by PROMODEL capabilities. The simulation engine is a single pre-coded PROMODEL program consisting of multiple high level modules. A central core module houses support activities common to most factories, including order arrival and release,

inventory management and control, routing, and other simulation housekeeping tasks. The simulation engine also contains a process module for each potential piece of process equipment. Process behavior is defined at several levels. First, fundamental process behavior is embedded in the coding of each process module. Functionality includes value-added process behavior as well as support system behavior such as inbound/outbound queuing, depalletizing/palletizing and machine loading/unloading. Second, process behaviors are refined by initialization of simulation engine parameters at the beginning of a simulation run. Queue size is an example. Finally, unique characteristics (attributes) of each order are used to guide process routing decisions and to further refine process behavior for the order. PROMODEL does not support object oriented programming and simulation concepts such as inheritance or instancing. This necessitates considerable code duplication. For example, the simulation engine contains multiple modules (essentially identical) representing the maximum number of potential instances of the same piece of process equipment.

The integrator links the simulation user with the simulation engine, providing three primary services: user interface, program generator and output evaluator. Although the integrator appears to the user as "seamless", it is actually a highly-integrated virtual system, consisting of a customized CAD front-end, an RDBMS and the PROMODEL visual interface. The structure of the GIHMS integrator is best described relative to its functionality.

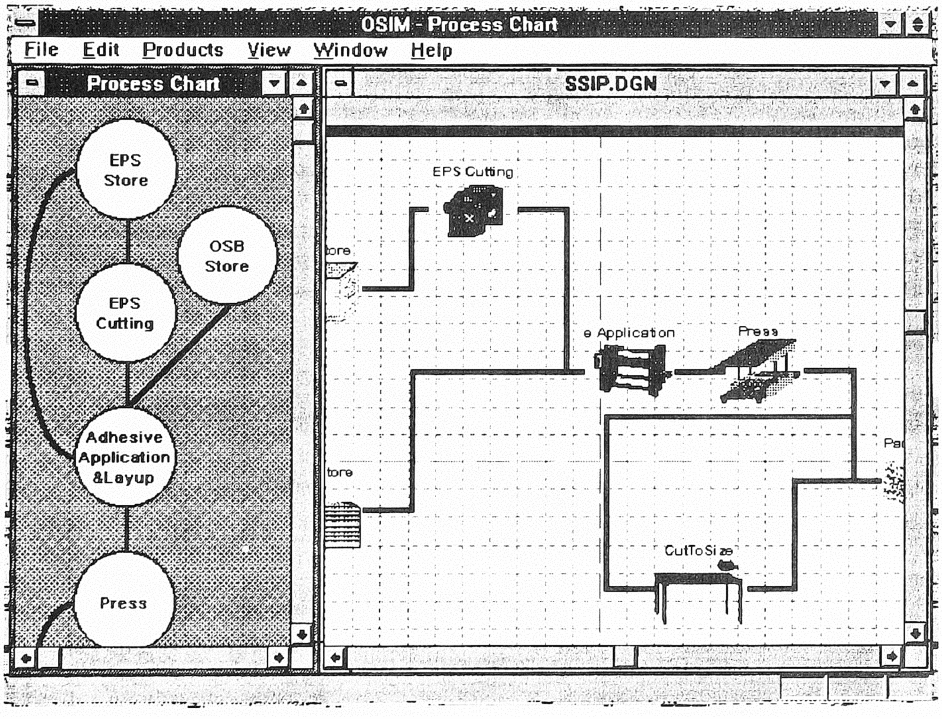

Figure 3. Preliminary Layout CAD Window

Factory configuration is performed in two serial stages. In the preliminary stage, the user first uses the Process Chart Window (Figure 3) to define an operations process chart for the product/processes to be used. The user is then prompted to select specific manufacturing equipment to perform each production operation. The user is allowed to select from comprehensive equipment libraries which are permanently maintained on the database. As equipment is selected, it is instantiated on the Preliminary Layout Window (Figure 3) as well as in a database associated with this factory configuration. The Preliminary Layout Window represents a preliminary physical layout of the manufacturing facility. Material handling equipment selection is performed in a like manner, resulting in directed material flow arcs between equipment icons. Correspondence is maintained between objects in the Process Chart Window (operation icons and precedence arcs) and objects in the Preliminary Layout Window (process equipment icons and material flow arcs).

Characteristics and behaviors associated with each equipment instance are maintained in a database corresponding to this factory configuration. Default behaviors are inherited from the permanent equipment library as equipment is instantiated. Specific behaviors may be reviewed and modified by selecting the desired equipment icon in the Preliminary Layout Window and expanding it to the desired level of detail. Note that each equipment instance in the factory configuration database corresponds to a unique equipment object in the Preliminary Layout Window and to a unique equipment module in the simulation engine. The GIHMS integrator uses the factory configuration database to produce an initializing data set which configures the simulation engine to represent the factory being modeled. Much of the integrator functionality described above is written in PARADOX (Borland International, Inc. 1993), a multi-media RDBMS which runs in the PC WINDOWS^{TM} environment. PARADOX was selected because of its high level WINDOWS^{TM} programming constructs and in recognition that the integrator is a data intensive application. Elements of the custom CAD windows are developed in C++ wherever PARADOX programming constructs proved to be inadequate. Communication between C++ and PARADOX is via Microsoft's Open Database Connectivity Standard (ODBC).

In the final factory configuration stage, the user arranges equipment on the factory floor, identifies physical paths (aisles) for the material handling equipment, refines material handling equipment specifications, and schedules labor. While working in the Final Layout Window (Figure 4), the user is presented with a scaled factory template complete with the same equipment icons which were active in the Preliminary Layout Window. Note that each icon corresponds to a unique equipment object in the Preliminary Layout Window, a unique equipment instance in the factory configuration database and to a unique equipment module in the simulation engine. This correspondence is maintained throughout the final factory configuration, as equipment objects are relocated and physical paths are developed. The final factory configuration functionality of the GIHMS integrator is provided by the object oriented visual interface of PROMODEL. The interesting mix of functionality in the final factory configuration stage is driven more by the capabilities of the PROMODEL visual interface than a logical structuring of design activities.

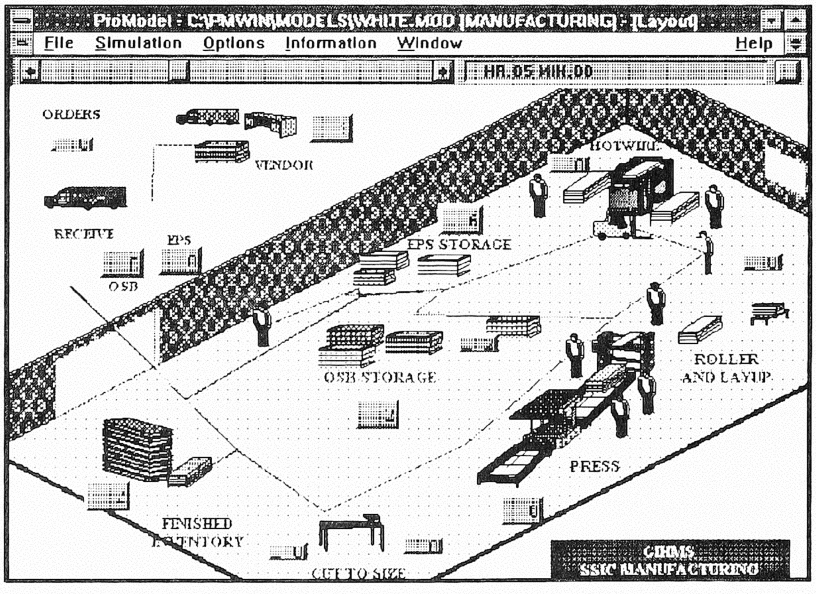

Figure 4. Final Layout CAD and Animation Window

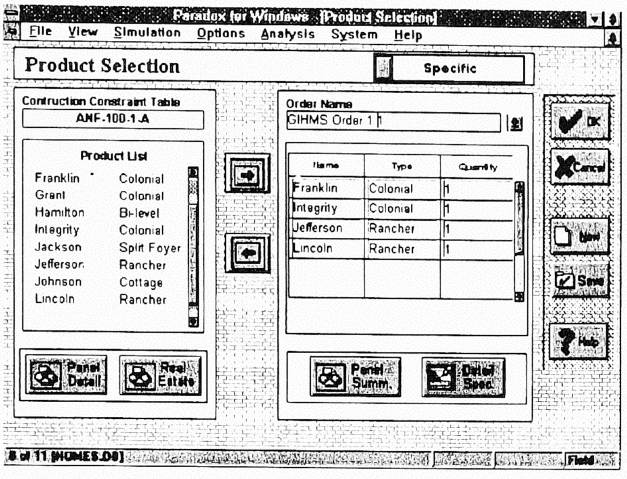

Figure 5. Product Selection window

The simulation engine is driven by customer orders specified by the user. The GIHMS integrator allows the user to select customer orders (homes to be built) from a permanent library of house plans, customize the orders to builder/customer specifications and, finally, schedule the orders for production (Figure 5). The GIHMS integrator then prepares a customer order data set from the specified orders. Unique characteristics (attributes) of each order are used by the simulation engine to make process routing decisions and to detail process behavior for the order. This GIHMS integrator functionality for customer orders is developed using PARADOX.

After the factory is configured and customer orders have been specified, the simulation engine is run. PROMODEL displays an animation of simulated factory operation (Figure 4) and generates and maintains output data which can be used for further analysis. PROMODEL's visual interface is used by the GIHMS integrator to allow the user to generate a variety of reports.

4. CONCLUSIONS AND FUTURE RESEARCH

This paper has described a new structure for SME development. The structure is centered around a commercial manufacturing simulator, which is closely coupled with an RDBMS. While the research found the structure to be feasible, it identified several shortcomings. The most significant is that process equipment and product selection is limited to those contained in the GIHMS libraries. It should be noted, however, that new equipment and products may be emulated by similar items included in the library and that equipment performance can be tailored somewhat by user specified parameters. GIHMS capabilities can also be extended by "learning" new equipment and products, expanding the GIHMS library and adding new equipment modules to the simulation engine. Another shortcoming, the use of two visually similar (but functionally unique) CAD windows for factory layout, is caused by the use of two programming environments (PARADOX/C++ and PROMODEL). As a result, the user cannot access all design functions from a single window.

The research also identified several shortcomings of the software development tools. Constructs provided by PARADOX were not ideal for CAD modeling, requiring the addition of a customized C++ CAD front-end. PROMODEL's general purpose modeling constructs did not support object oriented inheritance or instancing, resulting in considerable code duplication. This could eventually limit the number of machines and processes available for selection and may cause software maintenance difficulties. PROMODEL's inability to indirectly address key model entities and attributes introduced significant coding penalties. Finally, PROMODEL did not support the use of external routines such as expert systems and optimization. Newer software versions are expected to alleviate many of these problems.

Future research will address several areas. First, commercial design/engineering software will be integrated into GIHMS, allowing the import of home designs and the development of new designs within GIHMS. Second, intelligence capability will be added to GIHMS, in the form of rule-based expert systems, supporting factory configuration, houseplan selection, and operations management. Third, research will consider extending the scope of GIHMS out of the factory to the construction site.

5. ACKNOWLEDGEMENTS

This research was sponsored by the United States Department of Energy, through the Energy Efficient Industrialized Housing (EEIH) research program. The EEIH program is jointly conducted by the Center for Housing Innovation, University of Oregon, the Florida Solar Energy Center and the Department of Industrial Engineering and Management Systems, University of Central Florida. (DOE Contract No. DE-FC03-89SF17960)

6. REFERENCES

Centeno, M. and C. Standridge (1993). Databases: designing and developing integrated simulation

modeling environments. *Proceedings of the 1993 Winter Simulation Conference*. San Diego, CA, pp. 526-533.

Ford, D.and B. Schroer (1987). An expert manufacturing simulation system. *Simulation*, 48(5), 193-200.

Haddock, J. and R. O'Keefe (1990). Using artificial intelligence to facilitate manufacturing systems simulation. *Computers in Industrial Engineering*, 18(3), 275 - 283.

Haddock, J. (1988). A simulation generator for flexible manufacturing systems design and control. *IIE Transactions*, 20(1), 22-31.

Medeiros, D. and R. Sadowski (1983). Simulation of robotic manufacturing cells: a modular approach. *Simulation*, 40(1), 3-12.

Mize, J., et al (1992). Modeling of integrated manufacturing systems using an object-oriented approach. *IIE Transactions*, 24(3), 14-26.

Najmi, A. and S. Stein (1989). Comparison of conventional and object-oriented approaches for simulation of manufacturing systems. *1989 IIE Integrated Systems Conf. & Society for Integrated Manufacturing Conf. Proc.* Atlanta, GA.

Ozdemirel, N. and G. Mackulak (1993). A generic simulation module architecture based on clustering group technology model codings. *Simulation*, 60(6), 412-433.

- (1993). *PARADOX for Windows™ , Version 4.5, Getting Started*. Borland International, Inc. Scotts Valley, CA.

- (1993). *ProModel for Windows™ User's Guide, Version 1.05* . ProModel Corporation. Orem, UT.

Schroer, B. (1989). Improving the manufacturing simulation modeling environment. *Manufacturing Review*, 2(4), 283-289.

Treu, S. (1988). Designing a "cognizant interface" between the user and the simulation software. *Simulation*, 51(6), 227-234.

Youngblood, S. (1991). Increasing simulation and model utility via improved interface techniques. *Proceedings of the 1991 Summer Computer Simulation Conference*. SCS, 57-62.