Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

REUSABLE MODELS: MAKING YOUR MODELS MORE USER-FRIENDLY

Onur M. Ulgen
Timothy Thomasma

Production Modeling Corporation
One Parklane Blvd., Suite 1604W
Dearborn, MI 48126

and

University of Michigan - Dearborn
4901 Evergreen Rd.
Dearborn, MI 48128

Norm Otto

Ford Motor Company
Research Staff
Dearborn, MI 48124

ABSTRACT

In this tutorial we identify four different types of Reusable Simulation Models (RSMs), namely, (1) fully documented simulation models, (2) parameterized simulation models, (3) special purpose simulation program generators, and (4) general purpose simulation program generators. Each type of RSM may be accessible through a different kind of Simulation Support Environment (SSE). We identify three major classes of SSEs, namely, (a) procedure oriented programming language based SSEs, (b) spreadsheet program based SSEs, and (c) object-oriented, visual interactive SSEs. Out of the possible twelve combinations of RSMs and SSEs, we discuss eight examples (case studies) of the six most popular combinations.

1 THE IMPORTANCE OF REUSABILITY

Simulation is a useful tool in analysis and design of complex manufacturing systems because it can give insight into manpower and equipment requirements, cycle time, inventory, and throughput for alternative designs before any physical changes are made in an existing system or any commitment is made in hardware or facilities for a new system.

A large part of the work in a simulation study consists of writing, testing, and debugging the simulation program. Even though simulations are written in languages that are designed for simulation programming, they are like programs written in any language in that they are time consuming and labor intensive to write, test, and debug. This limits the applicability of computer simulation as a modeling and analysis tool. Complex simulation models must be built by simulation experts that are expensive to hire and hard to find. Currently, it is not easy for the average plant engineer to build a simulation model of a complex system.

We think that simulation should be employed as much as possible by the plant engineers who have the most intimate knowledge of the systems they want to simulate. Plant engineers who can conduct their own simulation studies are able to obtain reliable data inexpensively. Their day-to-day experience in their plant allows them to validate models as they build them. The study can incorporate input from people in the plant at all stages while the model is built, validated, and exercised. Therefore, support for implementing the solution developed in the study is high.

Complex simulation models are often written by outside simulation experts who may not be available to help the plant engineers after the simulation study has been completed. The full potential of the simulation models may not be realized if the models can't be reused by the plant people for follow-up analyses of the system. Figure 1 depicts the relationship between the ultimate model user (the plant engineer) and the simulation model. If the simulation model is not designed to be user-friendly from the beginning of the project, the user may never be able to exercise the model directly. The need for a model developer in order to use the model may hinder the extensive use of the model for system analysis. The proper simulation support environment may alleviate this problem and increase the use of simulation.

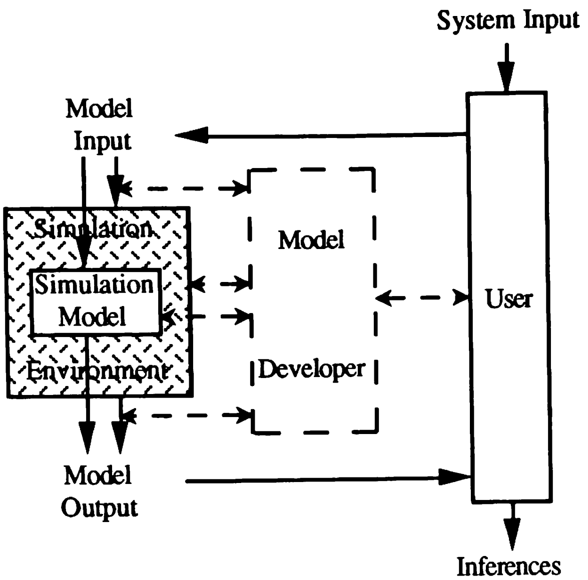

Figure 1: Roles in a Simulation Study

In collaboration with some of its clients, Production Modeling Corportaion is building user-friendly simulation support environments (SSEs) for popular simulation programming languages and simulators in order to increase the use and reuse of simulation models by plant engineers.

2 A CLASSIFICATION

In order to increase the payoff from an investment in simulation modeling, one would like to be able to reuse the models to solve more than one problem or design more than one system. Reusability is facilitated by SSEs. Balci *et al.* (1990) define an environment as "an *integrated* set of hardware and software tools that provide cost-effective, automated support throughout the *entire* development life cycle." For our purposes we are interested in the *problem solving* or *system design* life cycle, which might or might not include model development (Figure 1).

Wyatt (1990) distinguishes between formal and informal software reuse in software engineering. Fully documented simulation models (FDSMs) permit formal reuse of the software in which the model is embodied, by providing all specifications and logical design documents to the user along with the code. It is not practical for a user to informally reuse selected data structures and code from a model, unless the model or support environment is constructed to facilitate such reuse. We distinguish three levels of support for this informal software reuse.

Parameterized simulation models (PSMs) can be reused by changing numeric inputs to the model (parameters), as in an experimental frame. This type of reuse can be implemented without an environment, just by the way the model is constructed.

Special purpose simulation program generators (SPSPGs) allow the user to reuse pieces of simulation code and data structures by automatically aggregating those pieces together into a model, according to the user's specification. The SPSPGs include modelware packages such as WITNESS, ProModelPC, SIMFACTORY, which Mathewson (1989) classifies as application-specific. The reason that these are "application-specific" or "special purpose" is that behavior of a system can only be modeled if the SPSPG contains the code and data structure for the system element that exhibits that behavior.

General purpose simulation program generators (GPSPGs) support the same kind of model reuse that SPSPGs do, but also provide some mechanism (finite state machine, activity cycle diagram, Petri net, operation network) that allows the user to graphically specify new behavior. There are several efforts being made to develop these (Ulgen, *et al.* 1989, Thomasma, *et al.* 1990, Balci, *et al.* 1990, Gordon *et al.* 1990, Zeigler 1990).

Both SPSPGs and GPSPGs are by nature supported by model development environments, and thus are presented and studied in the context of SSEs. Balmer and Paul (1990) have given an excellent survey of these. We classify SSEs on the basis of the software technology used to implement them. Procedure oriented language based simulation environments (PLSEs) are SSEs that are constructed using a combination of FORTRAN or C subroutines, generally in conjunction with a transaction-flow based simulation language like GPSS, SLAM or SIMAN. Spreadsheet program based simulation enviornments (SSSEs) use spreadsheets (Excel, Lotus 1-2-3) in conjunction with models that are constructed either by using simulation languages (GPSS, SLAM, SIMAN) or modelware (WITNESS, ProModelPC, SIMFACTORY). Object oriented, visual interactive simulation environments (OOSEs) make visually explicit to the user the encapsulation of simulation code and data structures that model the real-world system components.

Even though FDSMs and PSMs can be reused without simulation environments, SSEs are usually provided for them. Therefore, in our classification or reusable models, we use two axes: the type of model (1. FDSM, 2. PSM, 3. SPSPG, 4. GPSPG) and the type of simulation support environment that is packaged with it (a. PLSE, b. SSSE, c. OOSE). Figure 2 shows the twelve possible combinations of reusable simulation models and SSEs.

Reusable Models

	(1) FDSMs	(2) PSMs	(3) SPSPGs	(4) GPSPGs
(a) PLSEs	✓	✓	✓	
(b) SSSEs		✓✓✓		
(c) OOSEs			✓	✓

SSEs

Figure 2: Classification of Case Studies

3 EXAMPLES AND OBSERVATIONS

Of the twelve possibilities, we discuss eight examples (case studies) of the six most popular combinations (Figure 2). The examples are taken from actual applications of simulation done for electronics and automobile related industries by Production Modeling Corporation. A variety of simulation languages (SIMAN/Cinema, GPSS, WITNESS), spreadsheet programs (Excel, Lotus 1-2-3), and procedure-oriented and object-oriented programming languages (FORTRAN, C++, Smalltalk) are used in these case studies.

We have found that different types of reusability are appropriate for different situations. Table 1 summarizes the various characteristics, strengths, and weaknesses of the reusable model types.

When building models and environments for users to use and reuse, one needs a "user-centered", not a "user-friendly" approach. The model building should be customer-driven. Standard best practices in user interface design should be used, but everything should be reviewed, tested and approved by the customer, and the customer's suggestions for improvements in the user interface should be implemented. If the user is going to make significant use and reuse of models, and if models will become corporate resources (Johnson 1990), then the model developer must be prepared to support the model and continuously improve the model building process. If the model and the support environment are well designed, it is far less costly to provide the required support than it is to completely rebuild a new model whenever a new problem is to be solved or a new system is to be designed.

Table 1: Characteristics of Reusable Model Types

	FDSMs	PSMs	SPSPG	GPSPG
Problem domain	Fixed	Fixed	Fixed	Some flexibility
Process flow	Fixed	Some flexibility	Flexible	Flexible
Equipment types	Fixed	Some flexibility	Some flexibility	Flexible
Simulation experience of user	High	Medium-Low	Low	Medium
User /	***devel-***	***oper***	***inter-***	***action:***
- Specs	High	High	None	None
- Development	High	Low	None	None
- Transfer	Medium	Medium	Low	High
- Follow up	Medium	Medium	Low	Low
Model development time	2 - 6 months	1 - 4 months	6 - 24 months	12 - 36 months
Model	Process oriented	Process oriented	Event oriented	Object-event oriented
User modeling time	1 - 3 weeks	Within hours	Within hours	Within hours/ days

REFERENCES

Balci, O., R.E. Nance, E.J. Derrick, E.H. Page, and J.L. Bishop. 1990. Model generation issues in a simulation support environment. In: *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski, and R.E. Nance, 257-263. Institute of Electrical and Electronics Engineers, New Orleans, Louisiana.

Balmer, D.W. and R.J. Paul. 1990. Integrated support environments for simulation modeling. In: *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski, and R.E. Nance, 243-249. Institute of Electrical and Electronics Engineers, New Orleans, Louisiana.

Gordon, R.F., K.J. Gordon, E.A. MacNair, and J.K. Kurose. 1990. Hierarchical modeling in a graphical simulation system. In: *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski, and R.E. Nance, 499-503. Institute of Electrical and Electronics Engineers, New Orleans, Louisiana.

Johnson, M.E. 1990. An examination of corporate model management. In: *Proceedings of the 1990 Summer Computer Simulation Conference*, eds. B. Svrcek and J. McRae, 243-248. Society for Computer Simulation, Calgary, Alberta, Canada.

Mathewson, S.C. 1989. Simulation support environments. In: *Computer Modeling for Discrete Event Simulation*, ed. M. Pidd, 57-100. London: J. Wiley and Sons.

Thomasma, T., Y. Mao, and O.M. Ulgen. 1990. Defining behavior in a hierarchical object oriented simulation program generator. In: *Proceedings of the 1990 Summer Computer Simulation Conference*, eds. B. Svrcek and J. McRae, 266-271. Society for Computer Simulation, Calgary, Alberta, Canada.

Ulgen, O.M., T. Thomasma, and Y. Mao. 1989. Object oriented toolkits for simulation program generators. In: *Proceedings of the 1989 Winter Simulation Conference*, eds. E.A. MacNair, K.J. Musselman, and P. Heidelberger, 593-600. Institute of Electrical and Electronics Engineers, Washington, District of Columbia.

Wyatt, D.L. 1990. A framework for reusability using graph-based models. In: *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski, and R.E. Nance, 472-476. Institute of Electrical and Electronics Engineers, New Orleans, Louisiana.

Zeigler, B.P. 1990. *Object-Oriented Simulation with Hierarchical, Modular Models*. New York: Academic Press.