Proceedings of the 1989 Winter Simulation Conference
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

HOW TECHNOLOGY LIMITS SIMULATION METHODOLOGY

Chair:
C. Dennis Pegden
Systems Modeling Corp.
Sewickley, PA 15143

Panelist:
Cindy Morey
Brian Unger
Peter Welch
Deborah Medeiros
Timothy Thomasma

1. INTRODUCTION
C. Dennis Pegden, Systems Modeling Corp.

In the past decade, a number of technological changes and creative developments have taken place to enhance the effectiveness of tools used in simulation modeling. These include developments in: computers, simulation modeling languages, user-interface design, graphical animation, and statistical analysis techniques. These changes and developments have had a direct impact on the practitioners of simulation. Compared to a decade ago (when simulation was largely batch oriented, required a high level of programming skills, and lacked graphics for model construction and/or animation) the simulation practitioner today has considerably more powerful and easier to use tools for simulation. As a result, simulation projects today are both less difficult to perform, and are more likely to be successful.

As we look to simulation in the 1990's, there are a number of technologies that are currently developing which will likely further impact our simulation tools. These include a continuation of advances in some of the current high activity areas (i.e. computer technology, graphics, user-interface design, animation, and statistical analysis) and also new developments based on emerging ideas in software development and expert systems. In specific, many of the new ideas developed in object-oriented software programming will undoubtedly impact and improve simulation languages.

The purpose of this panel is to examine some of the constraints that current technology places on simulation and to discuss the merits of some emerging technological changes related to simulation. Our objective is to provide insight into the impact of new technologies on the simulation practitioner of the 1990's.

The panel will begin by assessing the current limitations imposed on simulation from the viewpoint of an experienced practitioner of simulation. Next, the panel will discuss the potential impact that recent developments in parallel processing may have on computational intensive simulations. This will be followed by a discussion of potential advances in statistical analysis methods and the emerging role of graphics in simulation. The panel will conclude by discussing the benefits that the object-oriented framework may bring to simulation modeling.

2. A PRACTITIONER'S VIEW - WHAT IS NEEDED.
Cindy Morey, General Motors Canada Ltd.

Simulation has had only moderate acceptance within the manufacturing environment to date. Currently simulation is not considered to be an essential ingredient required for the manufacturing of a product and as such is used and accepted by only some in the manufacturing sector. Many hold the view that simulation is a cumbersome process requiring excessive time and skill and delivering little useful information. This view stems from a lack of understanding of simulation and the methodology that should be used in carrying out a sound simulation study.

Proper methodology is the critical success factor in any simulation study yet it is most often taken for granted. More emphasis is placed on choosing the appropriate modelling language than on using an appropriate methodology.

What is lacking is a close integration of simulation methodology with the simulation model. The scope of simulation software needs to be expanded to include all aspects of a simulation project. All languages provide a solid "modelling" framework. What they do not provide is a solid "methodology" framework.

Three areas for improving methodology will be discussed. They are the integration of plant floor information with simulation software, current hardware and software and training.

Integration of Plant Floor Systems

1. CAD information: CAD drawings should provide all coordinate data necessary for defining the animation and process model.

2. PLC Logic: Current PLC logic should be capable of being uploaded to a simulation model or downloaded from a simulation model.

3. Monitoring Systems: Current cycle time and downtime collection systems should readily provide input data and the software should assist in the analysis and distribution fitting of the data.

4. Experimental Design: Simulation software should assist in setting up experiments, validation and interpreting output.

5. Scheduling: the software should be capable of testing planned production schedules and assist in developing scheduling rules.

Current Hardware and Software

Restrictions on addressable memory space severely limits model size and execution speed is still too slow. Graphics and animation have brought simulation into the manufacturing arena but at best are crude. The industry should evolve to CAD standards. Much work is still left to be done in the creation of constructs specific to flexible manufacturing. Expert system technology needs to be incorporated in the software to enable the full integration of methodology with model creation.

Training

There is an excellent availability of training courses which are language specific. There is a definite lack of training courses which deal in methodology. Modelers require training in statistical analysis, validation techniques, experimental design, simulation theory and other operations research tools that could be used to complement simulation. Training in simulation methodology is difficult to find outside of undergraduate programs.

Building a solid methodology framework into simulation software will result in more successful simulation studies and shorter project life cycles resulting in a wider acceptance of simulation by the manufacturing industry.

3. THE IMPACT OF PARALLEL PROCESSING.

Brian Unger, Jade Simulations

Today, nearly all simulations take between five minutes and five hours to execute. Further, the applications of simulation to more and more complex problems continues to place simulation in the highly computation-intensive world. Simulation is thus till in the batch era of computing.

Execution of computationally-intensive simulations requires either very powerful computers or that the problem is over-simplified. In some cases, it is simply not feasible to use simulation to solve the problem at all. Thus, two of the significant limitations imposed upon simulation from a practical view are the requirements for long execution times and the noninteractivity that long execution times imply.

The recent emergence of extremely cost effective massive parallel computers offers the opportunity to remove these limitations. Today, a 200 MIP multicomputer can be purchased for about 1/100 of the cost of a 200 MIP supercomputer. This inexpensive computing power offers the potential to dramatically broaden the range of problems that can be simulated interactively on a parallel engineering workstation. The only barrier to achieving this goal is the software needed to synchronize parallel simulations.

Virtual Time is the breakthrough needed to realize the potential of parallel computers for simulation. A Time Warp software implementation of Virtual Time can synchronize a parallel simulation in a way which is transparent to the simulation developer. Evidence has emerged over the past three years that 50 to 90 % speedup efficiencies (for example 50-90 times speedup on a 100-node multicomputer) are achievable for a broad range of applications.

Truly interactive simulation would mean that alternative system designs could be evaluated within human response times. Interactive exploration of alternative architectures would both expand the areas of simulation applications and would replace the current need for carefully planned batch experiments. Interactive simulation would also have a significant impact on education and training applications. Many more training and education applications would exist if the current $20 million simulator could be built for $200,000.

4. ADVANCES IN STATISTICAL ANALYSIS
Peter Welch, IBM Corp.

As we move through the 1990's, most system simulation will be done on powerful, personal workstations. This will have a dramatic effect on the kinds of statistical studies that are performed, the level of statistical support software, and the output analysis interfaces that this software presents to the practitioner. Workstations will provide the processing power currently available only on the largest mainframes. Further, these MIP's will be personal and the experimenter will, in general, be accountable to no one after the initial purchase of the workstation. Hence, with multitasking he will be motivated to run simulation in the background at all times if necessary.

As one consequence of this surfeit of computing power there will be less interest in the small sample behavior of confidence interval procedures. They will be sequential in nature and have reliable large sample behavior with safeguards against generating an invalid small sample interval.

As a second consequence of the availability of computing power, there will be frequent large parametric studies: the characterization of model performance measures over a multidimensional parameter space. These parameters will be both continuous and discrete. The statistical software will provide a high level "intelligent" interface for such studies. It will provide and carry out for the user an appropriate experimental program. The experimenter will indicate the parametric region of interest and the software will plan and carry out a sensible statistical exploration of that region. Since simulation deals with the artificial world of the model there are not the usual practical problems and expense of experimentation involved. The interface will have a heavy emphasis on graphics and will, at any time, effectively display what is has learned. The experimenter can then refine his directions by narrowing or widening the boundaries of the parametric region, changing accuracy requirements, indicating an interest in a maximum or minimum, etc. The software may not initially obtain exactly the information the experimenter wants but it will create a dialog with the experimenter which will lead to that information.

The use of multiple microprocessors will fit naturally into the simulation application. The method of independent replications provides a natural parallelism for simple model evaluation and, in parametric studies, the multiple points in the parameter space can be evaluated in the obvious parallel fashion.

5. THE EMERGING ROLE OF GRAPHICS IN SIMULATION
Deborah Medeiros, Pennsylvania State University

Computer graphics is used in three general areas: model construction, data, analysis (input and output) and animation of simulated output. The model construction and data analysis areas have received far less attention than animation, and their graphical requirements are less demanding, consisting primarily of symbol placement and graphing. Therefore, the discussion will focus on animation of simulation models.

Animation is used to verify model correctness and to explain the model to others. These tasks become easier as the animation more closely resembles the actual system. However, it is also important that the complexity of the animation corresponds with the level of detail present in the model.

Present animation systems include 2D character cell based systems, 2D bitblt systems, and 3D wireframe systems. Many are tied to a specific computer or graphics system configuration. It is often difficult to achieve acceptable speed of operation for complex systems with many moving components. The level of realism available is limited as well. Portability, speed, and image quality are the major technological limitations.

These technological constraints limit the application of simulation. For example, one prevalent use of simulation is in the design of complex, computer-integrated manufacturing systems. Since these systems are capital-intensive, detailed study is justified. In the initial stages of design, high level models of several alternative configurations might be developed. As design decisions are made, the model becomes more detailed, and the requirements placed on an animation system increase. Control software must be developed for the final system configuration : creation and testing of this software can become a major bottleneck in the implementation phase. A sufficiently detailed simulation model could be used as graphical animation, and do so in real time. The user of simulation as a test bed is common in electrical engineering and computer science applications, but relatively rare in manufacturing.

CAD/CAE, scientific visualization, and image processing needs are driving development of more sophisticated graphics systems. Graphics capabilities of personal computers are increasing. Specific developments which will impact simulation animation include:

New graphics coprocessors. Devices such as the Intel 1860 will greatly reduce the time required to get a complex image on the screen by supporting more advanced graphics functions, thus offloading the host CPU.

High-performance graphics workstations. These systems are improving in speed of display and manipulation of color-shaded images, primarily to support solid modeling and image processing applications. Display technologies such as stereo imaging further enhance realism and human ability to understand a complex image.

Continued development of graphics standards. Standards such a X-Windows and PHIGS will allow increased portability of applications and an easier migration path to new technology. At present, sacrifices in speed often result from adherence to standard interfaces, limiting their use in high-performance systems. It is hoped that, as the standards become developed, hardware will be designed to optimize system performance under these standards, thus leading to increased use.

Capability to support combined video images and computer-generated graphics. With the development of a HDTV standard, video images will have sufficient resolution to be used as the background for an animation, enhancing realism and reducing animation development time.

These developments will result in faster display for current animation systems, but more importantly, will allow a new generation of simulation animation packages suitable for detailed and realistic modeling. This, in turn, will open up new application areas, such as control system design and testing, that are difficult at best using current technology.

6. THE PROMISE OF OBJECT-ORIENTED SIMULATION

Timothy Thomasma, University of Michigan, Dearborn

Simulation is used as a tool in the design and implementation of complex systems. There are stages in this activity, just as there are in design and production of machinery or design and construction of buildings. It has been suggested that different modeling tools be used at different stages of system design and implementation. For example, a rapid modeling tool could be used in early design stages, followed by a series of increasingly detailed simulation models. The purpose of these models is to evaluate design alternatives. When the final design is to be presented to management, a simulation language that supports animation might be used. Simulations can be used for other purposes as well: to train system operators, as specifications for engineers who will construct the system, and as testbeds for evaluating operational policies. As need for change in the system is encountered during its operational life, simulation can be used to evaluate proposed modifications.

Simulation has the potential for use as an analysis tool and means of communication at almost every stage in the life cycle of a complex system. In order to realize this potential, we need the ability to construct simulation programs that are capable of evolving throughout the system's life cycle. Object-oriented programming is meant to support incremental, evolutionary, exploratory programming, which is exactly what is needed if simulation is to keep pace with the activities of each stage in the design, implementation and operation of a complex system.

In an object-oriented programming language (OOPL), programming is done by sending messages to objects. An object is a collection of data, together with a list of messages that can invoke operations on that data. Each object is an instance of a class, which specifies the data and how the object is to respond to messages it receives. Classes are organized into hierarchies of subclasses by an inheritance mechanism. Objects, classes, and inheritance are essential if a language is to be considered object-oriented. In addition, typical OOPLs like Smalltalk and Objective-C provide data abstraction and late binding.

In an OOPL with data abstraction, the only way to gain access to an object's data is by means of the message that object can respond to. To use an object in a program, one need only know what the object does, not how it does it. As the software evolves, the internal details of an object's implementation can be changed without affecting any of the rest of the software.

Inheritance permits the programmer to create new classes by specifying only how the new classes differ from the old ones. This is similar to copying a piece of code and editing it to serve a new purpose, except that only one copy of the unchanged code is kept by the OOPL.

In compiled languages like FORTRAN and C, the data types of parameters for functions and subroutines must be specified when programs are compiled. This is termed early binding of data and operators. In Smalltalk this binding is done at run time; therefore it is termed late binding. One message in Smalltalk can be sent to instances of many different classes, each of which might respond in a different way to the message.

All these features of OOPLs encourage software reuse. Objects and data abstraction make it easy to understand a complex program and change it without creating software defects. Inheritance encourages the use of program templates and skeletons, as well as other forms of programming by exception. Late binding allows single implementations of algorithms to apply to many kinds of data, including new classes of objects that might be defined by the user after the algorithm is coded.

The use of OOPLs has led to the notion of an object-oriented simulation world review. Objects in an OOPL can be designed and named to match real-world objects. This makes programming much easier because it reduces the conceptual distance between the programmer's natural view of the world and the structure of the code being produced. Simulation languages and systems that present an object-oriented world-view to the user allow simulation models to be built more quickly and demand less computer expertise on the part of the analyst who is building them than languages that present other kinds of world-views to the user.

Simulation languages and systems are starting to more often present object-oriented world-views to the user. Most of the visual, interactive simulation languages for manufacturing do this. The presence of an object-oriented world-view is actually independent of the underlying software technology, which may have an entirely different underlying world-view, and which may not be implemented using an OOPL.

In the future, will any simulation languages but Simula provide objects, classes, inheritance, data abstraction, and late binding, and therefore themselves be true OOPLs? It is much easier to extend OOPLs to support simulation programming (several such extensions exist) then it is to make OOPLs out of existing simulation languages.

AUTHORS' BIOGRAPHIES

DR. C. DENNIS PEGDEN is the President of Systems Modeling Corporation. Dr. Pegden has taught Industrial Engineering at The Pennsylvania State University and the University of Alabama in Huntsville. During his tenure at the University of Alabama, he studied simulation language design and led in the development of the SLAM simulation language. After joining the faculty of the Pennsylvania State University, he continued his work in simulation language design and led in the development of the SIMAN simulation language. Dr. Pegden received his Ph.D. in 1976 in Industrial Engineering from Purdue University where he studied optimization. His current research interests include both optimization and simulation.

CINDY MOREY is Supervisor of Simulation with General Motors of Canada Ltd., Chevrolet-Pontiac-Canada Group. Ms. Morey held several Industrial Engineering positions before becoming supervisor of the Simulation Group in 1985. GM of Canada Simulation Group designated a Centre of Expertise for the Chevrolet-Pontiac-Canada Group of General Motors Corporation in March 1989. The Simulation group now provides assistance and technical support in simulation to 36 vehicle assembly, fabrication and engine plants located in Canada, U.S. and Mexico. Ms. Morey received her Bachelor of Industrial Engineering in 1983 from General Motors Institute in Flint, Michigan.

BRIAN UNGER is the President of Jade and a Professor of Computer Science at the University of Calgary, Alberta, Canada. Jade Simulations is a recent startup company that has been formed to provide object-oriented parallel simulation tools that implement Virtual Time. He has been working in the filed of simulation for the past 20 years, and specifically in the area of parallel simulation for the past 8 years. Dr. Unger received his Ph.D. in Computer Science from the University of California in San Diego in 1972 and has published widely in the field of simulation since that time.

PETER WELCH has been on the staff of the IBM T.J. Watson Research Center since 1956. From 1951 to 1956 he was with the Physical Science Laboratory of New Mexico State University, where he worked on radar signal processing. At IBM Research, he has worked in the areas of speech recognition, spectral estimation, queueing theory, seisimic signal processing, fast Fourier methodology, pattern recognition, computer system performance modeling, simulation output analysis, and graphic system design. He is currently Manager of Modeling and Analysis Software Systems. He received his Ph.D. in Mathematical Statistics from Columbia University in 1963. He is an Adjunct Professor of Statistics at Columbia University.

D.J. MEDEIROS is an Associate Professor of Industrial Engineering at the Pennsylvania State University. Her research interests include manufacturing systems analysis, CAD/CAM, and simulation. She is an Associate Editor of IIE Transactions, a past Director of the Production and Inventory Control Division of IIE and a member of the College-Industry Council on Material Handling Education. Dr. Medeiros holds a B.S.I.E. from the University of Massachusetts at Amherst and a M.S.I.E. and Ph.D. from Purdue University.

TIMOTHY THOMASMA has been a senior consultant for the Production Modeling Corporation since 1987. He is also on the faculty of the Department of Industrial and Systems Engineering of the University of Michigan-Dearborn, where he collaborates with Dr. Onur M. Ulgen in research on object-oriented simulation environments for manufacturing. Dr. Thomasma received the Ph.D. in Industrial and Operations Engineering in 1983 from the University of Michigan in Ann Arbor and is a member of IIE, SIAM, IEEE-CS, and a senior member of SME/CASA.