Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

SOFTWARE/MODELWARE APPLICATION REQUIREMENTS (PANEL)

Session Chair

David Withers

Mead Data Central
PO Box 933
Dayton, OH 45401

Panelists

Phil Cohen

Artificial Intelligence Center/EK274
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025

Laura Giussani

Daxus Corporation
One Oliver Plaza 9th Floor
Pittsburgh, PA 15222

Col Tom Schuppe

Air Force Institute of Technology.
AFIT/ENS
Wright-Patterson AFB, OH 45433

Marvin Seppanen

Principal Consultant, Productive Systems
Route Five Box 46
Winona, MN 59987

ABSTRACT

This session is intended to provide a forum for software/modelware users to express new requirements for future software and modelware products. The panelists are all experienced application developers representing industry, academic, and military users. All exhibitors are invited to attend this session.

1 OVERVIEW

The paper is organized as follows: The guidelines for requirements are defined followed by an introduction of the panelists' backgrounds. The backgrounds are provided to aid the reader in understanding the panelist's perspective. The requirements are then defined and are grouped into several broad categories. A key phrase defining each requirement is in **boldface**. The paper concludes with a short summary.

2 GUIDELINES FOR PANELISTS

Requirements do not reference any current or planned product by name. The requirements are stated in terms of the functionality required, not in terms of changes or extensions to existing function/service.

Requirements cover the full spectrum of software/modelware products. They are presented in the following categories:

1. Hardware
2. Operating systems
3. Simulation languages
4. Special purpose simulation systems
5. Support for analysis of inputs or outputs
6. Validation and test tools.
7. Simulation systems

Requirements all conform to the following numbering scheme:

Initials.Category.Sequence number

3 BACKGROUND OF PANELISTS

(PRC) Dr. Philip R. Cohen has been concerned for nearly 10 years with the development of simulation systems that provide end-users with flexible tools for establishing and evaluating scenarios. He has developed multimodal user interfaces for simulation that integrate direct manipulation and natural language processing (keyboard and speech). These ideas have been implemented in the Shoptalk and Miltalk simulation systems for manufacturing and military command-and-control, respectively.

(LCG) Laura Giussani has been involved for several years with simulation and scheduling applications for clients in the metals, manufacturing, aerospace, and food/ pharmaceuticals industries. As a systems analyst for Daxus Corporation, Ms. Giussani's interests are integrating simulation and scheduling with other engineering disciplines (including systems design and systems emulation) and developing models to ensure that simulation and scheduling applications are both technically and culturally suitable to a client's needs. Daxus Corporation provides system development, hardware and software procurement, custom software development, prototyping, process simulations, quality assurance, training, and hardware and software maintenance.

(TFS) Col Schuppe currently directs three masters programs (operations research, strategic and tactical sciences, and space operations) and a PhD program (operations research) graduating approximately 60 MS and 2 PhD students per year. His department provides virtually all graduate level operations research and simulation education for United States Air Force officers. They also educate some Army and International officers. In addition, they provide research and consulting support throughout the Department of Defense. At the Institute, students and faculty use a full spectrum of simulation software and hardware.

(MSS) Dr. Seppanen is skilled in the development of special purpose simulators that are used by engineers with limited simulation experience to model and analyze manufacturing systems. Dr. Seppanen is experienced at viewing simulation from both the "Big Picture" perspective within an organization and at the "micro" level required to debug and validate the simulation code. This range of skills has been developed over more than 24 years of simulation practice in a variety of organizations using many kinds of computer hardware and software. Dr. Seppanen has taught simulation to both undergraduate and graduate students in engineering and business and to many continuing education and in-plant groups. He currently teaches at the University of St. Thomas. Productive Systems was founded in 1983 as an independent simulation consulting firm.

4 REQUIREMENTS FOR FUTURE PRODUCTS

4.1 Hardware

MSS.1.1 - Due to the sheer number available and their ever expanding capabilities, **personal computers** will continue to be the dominate simulation platform for manufacturing systems.

TFS.1.1 - Many large combat simulations are horrendously slow and **a need exists to significantly reduce run times**. Decision makers want more detail included in analyses and want to be able to assess the effects of changing input variables in large, theater-level combat models. When the necessary detail is added, mainframe run times in excess of 24 hours are not uncommon. These turn-around times are unacceptable when trying to provide quick answers in times of crisis. Parallel processing seems to hold much promise in this area, but there does not seem to be any workable hardware and software products on the market. Perhaps there are other alternatives to decrease run times on large models.

LCG.1.1 - **Simulation products need to utilize parallel processing and/or multi-processor architectures.** For instance, the software needs the capability of sensing what resources are available in its current environment and configure itself accordingly to allow the best possible performance.

4.2 Operating Systems

MSS.2.1 - A personal computer operating system must be found with the available software options and execution speed of MS-DOS but **without the 640K memory size limitation.** (Windows and OS/2 appear to slow down simulation execution.)

4.3 Simulation Languages

MSS.3.1 - Simulation languages should **support flexible model coding structures** such as, IF ... THEN ... ELSE and DO ... WHILE. This is equivalent to eliminating the GO TO or BRANCH statements.

MSS.3.2 - Simulation languages should be **designed for "paperless" model development.** All user defined names should be available for continuous display and sorted alphabetically by type. One step global name changes should be supported, for

example, change Q1 to Wait-for-Service.

MSS.3.3 - Simulation languages should include the **ability to merge models**. Potential name conflicts should be noted for user resolution as the merging model is loaded.

MSS.3.4 - Simulation languages must **include documentation for a range of users** from the novice to the expert with more experience than the vendor's technical support staff. A range of documents may be required to support different needs, i.e., overviews, college courses, tutorials, and references. On-line reference manuals may be appropriate.

MSS.3.5 - Simulation languages should **support flexible external data exchange.** Direct access to worksheets and databases should be supported. Direct reading of routing and schedule data structures should be supported.

TFS.3.1 - Not all simulation languages adequately **consider the teaching environment** with their documentation and books. Students need a single textbook, primarily for economic reasons. This text should cover all major topics and features of the language and be laid out in a logical teaching order. In addition to discussing features of the language, there should be many illustrated examples of modeling conventions employed to address common simulation situations. Many "texts" seem to be written by programmers, for programmers. Perhaps a textbook should be written by someone who has taught the language for some time. A textbook should also include a rich and varied set of problems for students exercises. It would also be helpful if a set of example problems could come bundled with the software. This would allow students to experiment with code that is known to work and can be used as a learning tool.

TFS.3.2 - **Make adding animation** to a simulation model an **easier** task. Currently, it seems that adding animation to a model can easily double model development time. This is difficult in a teaching environment, where student time must be allocated to many different topics. The documentation supporting animation must also be complete and detailed.

TFS.3.3 - Strengthen the **link between animation and the underlying code**. While animation has a great potential for increasing the validity of a simulation, it is possible to have the animation doing something inconsistent with the underlying code. This inconsistency can be inadvertent, or purposeful. Anything that would make the animation process more automatic and less prone to error should facilitate agreement between simulation code and the accompanying animation. Animation has been sold as an aid to validation. However, if misapplied, animation can actually complicate the validation process by introducing additional errors.

LCG.3.1 - There is a need for **simulation "templates," in the form of generic products with standard selections.** So far, major effort has focused on material handling; it needs to expand into less traditional ones (mixing, pumping, bottling lines) as well as into processing units.

LCG.3.2. - A capability is required to **integrate simulators with design software products for concurrent engineering** and to handle development of life cycle planning, beginning with concept definition and continuing through needs analysis and requirement definition, preliminary and detailed design, construction, integration, implementation and maintenance.

LCG.3.3 - Simulation products should **facilitate the development of integrated models for evaluating complex strategic business issues.** This will make the decision-making process of investment in manufacturing technology more understandable to the decision makers. For instance, a company should be able to examine its assumptions regarding the allocation of direct/indirect costs, explore those assumptions prior to developing new policies, and examine the cost of alternatives relative to each other and the profit implications of changes in the manufacturing process or equipment. Similarly, it should be able to evaluate the effect of introducing new products at various times in different locations, how assumptions regarding market penetration of the product affect decision making, and when a mature product should be phased out. Also, in addition to performing manufacturing analysis, it should "support" market and financial analysis.

4.4 Special Purpose Simulation Systems

MSS.4.1 - Special purpose simulation systems should provide means to **prevent users from outgrowing the tool as they become experienced.** It should not be necessary to start with a new tool or vendor when the focus of the simulation project changes or the required level of detail increases.

LCG.4.1A - A capability is needed to **handle mixed mode of operations** (batch, continuous, semi-continuous) with ease within one language **and to address the related issues of process simulation** (rule definition, equipment sizing, etc.) with systems other than "pure" ones.

4.5 Support for Analysis of Input and Output

MSS.5.1 - The support for the analysis of output should provide for **customized reports and graphs**. Comparison of multiple simulation runs or models should also be provided.

TFS.5.1 - Output analysis could be greatly facilitated by providing **better graphical output capabilities**. There are many high-quality graphics programs currently available that generate excellent quality plots. High-quality histograms, two dimensional plots, and three-dimensional response surfaces should be possible with today's technology. Even a simple scatterplot of X_i, X_{i-1} could be very helpful in an analysis. It would also be very helpful if these graphics programs could be bundled with the simulation software. If this is not possible, a direct link to existing commercial graphics packages (which could be purchased separately) would be very helpful.

TFS.5.2 - **Common statistical analysis capabilities should be included** within the simulation software. For example, it should be possible to implement batch means (both overlapping and non-overlapping) or the method of replications without having the analyst write all required code. Other common statistical analysis procedures, such as linear regression, ANOVA, autocorrelation and partial autocorrelation functions should be available in the software. If not directly available, it should be easy to link simulation output to popular commercial packages.

LCG.5.1 - We need a capability to **use simulation models in conjunction with generative procedures.** Simulation, an evaluative technique, provides a detailed performance estimate for a given set of decisions. It needs to be integrated with generative procedures that would produce an alternate set of decisions and choose optimal parameter levels at which to operate the system being simulated without full factorial experimentation.

PRC.5.1.1 Next-generation simulation systems need to offer substantially **more powerful user interfaces.** The interface to a simulation-based decision-support system should, at a minimum, assist the user in iteratively assessing the state of the complex system being studied; expressing, simulating, and comparing scenarios for altering that state; and, assessing the resulting state. Although modern simulation systems employ graphical user interfaces (GUI's), and hence benefit from their advantages, they are hindered by the absence of any provision by GUI's for end users to describe rather than merely select objects. This limitation is particularly severe for nontechnical users of decision-support or command-and-control systems, who would like to solve complex unstructured problems without delving into the intricacies of the underlying computer system.

Graphics and direct manipulation are effective interface technologies for many classes of problems, but they are limited in important ways. Specifically, they provide little support for identifying objects not on the screen, for specifying temporal relations, for identifying and operating on large sets and subsets of entities, and for using the context of interaction. Thus, the user who is trying to solve a problem when he does not know which objects, events, or time periods satisfy his constraints, can only point to entities on the screen, and attempt to determine their relevance. At times, this may be an effective strategy, but in many common circumstances, it can also be tedious or even completely ineffective.

For example, apart from the simple presentation of charts of tabulated variables, current simulation systems offer few tools for reviewing the history of the simulation. To support this aspect of problem-solving, **simulation systems need to record the events that occur**, and then provide sophisticated querying tools that assist a user's ad hoc exploration of that history. Because the user's questions will often refer to complex relationships among events, the query facility needs to offer the features of temporal logic. These requirements rule out the use of standard database query languages, such as SQL, as non-technical decision-makers should not have to learn the intricacies of underlying database structures. Moreover, virtually all of the present day query languages are weak in their handling of temporal relationships.

To overcome these limitations, simulation systems should provide multimodal interfaces, ones that allow users to employ direct manipulation when appropriate, but include natural language processing (spoken or typed) as necessary.

4.6 Validation and Test Tools

MSS.6.1 - Validation and testing tools should include the **ability to specify a confidence goal for a critical system statistic**. That statistic should be displayed during the simulation run and used to terminate the simulation run when the goal has been reached or appears to be impossible.

MSS.6.2 - Validation and testing tools should include the **ability to specify a set of test strategies or scenarios**. The tool should then execute the required simulation runs (preferably in the background) and provide a comparative statistical analysis of the results.

LCG.6.1 - **A capability is required to provide a**

variety of tools (analysis, knowledge based, database, etc.) and a framework to bring these different programs together. This framework should support the user in performing trade-offs, in deciding the critical factors of designing the system, and in studying the impact of different sets of requirements and constraints. Also, it needs to be able to process partial results and missing information and to provide "intelligent" defaults for initial factors.

LCG.6.2 - **A capability is needed to aid in subsequent functions,** such as selection (when a certain software is appropriate given a certain task), integration (when the different tools should talk to each other), adaptation (when to modify a tool depending on the information/task available) and documentation (be able to record how the tools have been impacted by previous functions).

4.7 Simulation Systems

MSS.7.1 - Simulation systems should **maintain a cumulative log file** which records each model change, key results of the simulation execution, and output analysis activity. These log entries should be automatically recorded. However, the user should be able to add notes or comments. Means should be provided to periodically summarize and print the log information for a particular project or timeframe.

MSS.7.2- Simulation systems should **support a hierarchical modeling structure**. Such a model development environment would start with a general system description. Additional detail would be added when and where required. The model structure should be goal based, i.e., equipment utilization, throughput capacity analysis, Kanban levels, or quality improvement.

LCG.7.1 - Tools are needed to **extend simulation modeling into actual start-up and operation phases of the systems.** One should be able to use simulation in an intermediate stage, where the model represents the "as-specified" mechanical plant and equipment. The model would contain sufficient detail to accurately test the control logic by direct connection with the controller. Similarly, one should be able to use simulation in the final stage, where the model represents the "as-built" environment, running in parallel to it and monitoring for discrepancies between real - operations and model activity. This is especially useful in debugging process control strategies and measuring actual improvements.

LCG.7.2 - **There needs to be a way of incorporating a feedback feature.** It would require software modeling that "learns" from feedback of actual process variances and attempts to adjust the model to a " best fit" based on the data acquired. It would also recommend where variables or rules might be altered by the developer. Ultimately, it should have self-diagnostic and correcting features.

PRC.7.1 To borrow a theme from Rothenberg (1986), simulation systems **need to go beyond "What-if".** For current systems, "What-if" means only setting new parameters, and rerunning the simulation. But, true "What-if" questions are far more complex, and answering them requires a more sophisticated underlying simulation framework (Cohen, 1991), one based on the satisfaction of conditions rather than just the scheduling (and unscheduling) of events at particular times.

Future simulation systems **should be built on a more flexible underlying computational framework than FORTRAN or even** object-oriented languages, namely upon Prolog. Prolog offers both an object and a relation-oriented view of simulation (Rothenberg, 1986; Narain, 1991), and is suitable for parallel simulations (Cleary, Unger and Li, 1988). Moreover, it incorporates a database, which can be mapped to industry-standard relational databases. The challenge is to build Prolog-based simulation systems that offer and improve upon the efficiency of present-day simulations, but that record history and allow for exploration and also problem-solving. In meeting this challenge, the importance of the user interface should not be underestimated, and the utility of graphics and animation should not be overestimated.

5 SUMMARY

This paper has discussed requirements for software to support discrete event simulation from the modelling practitioner perspective. Some of these requirements have already been implemented in a particular product, but the feature is not available in the product(s) used by the panelist. A consistent requirement is greater integration with other analysis tools and with tools for system design and generation. The summary message to vendors is that all of the best of the current functionality is needed as a base, and the new functions described here are additional to a solid language/system foundation.

REFERENCES

Cleary, J., Unger, B., and Li, Xining, 1989. "A distributed and-parallel backtracking algorithm using virtual time," *Distributed Simulation*, The Society for Computer Simulation, pp. 177-182.

Cohen, P. R., 1991. "Integrated Interfaces for Decision Support with Simulation", *Proceedings of the 1991 Winter Simulation Conference*, Nelson, B. L. , Kelton, W. D., and Clark, G M. (eds.) Phoenix, Arizona, pp. 1066-1071.

Narain, S. 1991. "An Axiomatic basis for general discrete-event modeling", *Proceedings of the 1991 Winter Simulation Conference*, Nelson, B. L., Kelton, W. D., and Clark, G M. (eds.) Phoenix, Arizona, pp. 1073-1082.

Rothenberg, J., 1986. "Object-oriented simulation: Where do we go from here?", *Proceedings of the 1986 Winter Simulation Conference*, Wilson, J. R., Henriksen, J. O., and Roberts, S. D. (eds.), Washington, DC, pp. 464-469.

AUTHOR BIOGRAPHIES

PHILIP R. COHEN is a senior computer scientist in the Artificial Intelligence Center at SRI International. He received his Ph. D. in computer science from the University of Toronto in 1978. His research interests include multimodal user interfaces, simulation, and applications to military and factory command-and-control. He is a Fellow of the American Association for Artificial Intelligence, and a member of the Manufacturing Systems Sciences Technical Advisory Board to the Semiconductor Research Corporation.

LAURA CHIARA GIUSSANI is a system analyst at Daxus Corporation. She holds a B.S.I.E and M.S.I.E from Purdue University as well as several minors in Foreign Languages. Her interests are in simulation and scheduling applications for diverse areas of process and discrete industries and in pursuing further studies in foreign languages. She is a member of Phi Beta Kappa, Alpha Pi Mu, Tau Beta Pi and IIE.

TOM SCHUPPE is Head of the Operational Sciences Department, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. He holds a PhD in operations research from The Ohio State University.

MARVIN S. SEPPANEN is the Principal Consultant for Productive Systems of Winona, Minnesota. He holds his BME, MSIE, and Ph.D. (Operations Research) degrees from the University of Minnesota. Before launching Productive Systems in 1983 he was an Associate Professor of Industrial Engineering at GMI Engineering & Management Institute and The University of Alabama. His consulting and research activities have been primarily in the area of manufacturing systems simulation. He is a Registered Professional Engineer; Senior Member, IIE; Member, ORSA, SCS, ASQC, and SME; and is certified at the Fellow Level by APICS.

DAVID H. WITHERS is the Director, Systems Evolution and Modeling, Mead Data Central, Dayton, Ohio. He holds a BS degree from the U.S. Coast Guard Academy and MS degrees in mathematics and computer science from Rensselaer Polytechnic Institute. His interests are performance assessment and capacity prediction for information processing systems. He is a member of TIMS, ORSA, TIMS CS, and ACM. Mead Data Central specializes in providing full text, on-line information via the LEXIS and NEXIS products.