Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

SIMULATION: TECHNOLOGIES IN THE NEW MILLENNIUM

Wayne J. Davis

General Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, U.S.A.

ABSTRACT

This paper first addresses future simulations needs from a generic point of view and then from the viewpoint of major stakeholders within the simulation community. These stakeholders include the simulation software developer/vendor, the corporate end user, the government end-user, the researcher and the educator. We then describe or outline the set of capabilities that will be needed to design and manage future systems, and also the limitations of the current simulation tools in meeting these needs. Finally, we conjecture about the kind of simulation-based design and planning capabilities that might exist in future manufacturing systems.

1 INTRODUCTION

Most interested persons believe that there will continue to be major advancements in simulation technologies. Already, we have seen new capabilities such as virtual reality being developed and applied to situations ranging from floating in space aboard the Hubbell telescope to riding an earthmover up a steep incline. Unfortunately, there is not space here to explore all of these advancements. Rather, this paper will address future changes in discrete-event simulation, the principal interest for most attendees here at the Winter Simulation Conference. Still, one should not completely ignore the advancements in simulation areas beyond discrete-event simulation. For example, it is likely that future simulation tools will provide virtual reality capabilities in order to animate simulations.

Clearly, there are new technologies that should provide significant opportunity for further development and application into the next millennium. Two of these areas are distributed and web-based simulation. Another area that has received only limited attention in the past, but should become a major area for future investigation, is on-line simulation.

Before discussing these new areas, however, we must ask a more fundamental question pertaining to the benefits that are to be derived from advancing research and development in these areas. In order to answer this question, we first discuss three basic types of simulation studies. We then discuss the future simulation needs and advancements with respect to each type of study from the perspective of each stakeholder in the simulation community. I will highlight exciting advancements. However, I will also discuss potential concerns that might effect a given stakeholder or the community at large. Finally, this paper will discuss potential new initiatives.

2 THE ESSENTIAL SIMULATION NEEDS

Simulation models and analyses derive their utility from being incorporated into other, more comprehensive studies. That is, a stand-alone simulation model or study is of limited utility. The primary reason we simulate a system is because we cannot provide an analytical expression for the predicted response of the system while it operates under a specified set of inputs and values for critical design (system) parameters. If we could specify an analytical solution, then most system analysts would not simulate. However, we do not have worry that there will be no need for simulation in the immediate future because a broad collection of systems exists for which one cannot specify an analytical solution. Moreover, in many cases where analytical expressions have been specified to project the system performance, there are usually simplifying assumptions that significantly degrade the accuracy of the model. The true benefits of the proposed solution can be tested only through rigorous simulation analyses.

In general, simulation analyses can be divided into three major types of studies: system design, system management and training. By far, the greatest application of discrete-event simulation to date is the design of systems. Because design usually implies the analysis of a system to be implemented or the operation of an existing system in a new manner, most design-oriented simulation studies are considered when they are off-line. The goal of these analyses is to specify the ideal values (not necessarily optimum) for the set of design parameters, which is to be implemented when the system is brought into operation. Once a system is made operational, there may be little need for the model. Indeed, many models employed during the design process are not validated because no physical system exists to provide the essential operational performance data for the validation process.

The use of simulation models to support on-line system management needs is in an embryonic development. There is a critical need for a validated model in order to make realistic projections of the future system performance. Furthermore, since most complex systems are time-variant, the need for model updating and validation is constant. Because the simulation analyses employed in system management require the on-line projection of the system operating under one or more alternative strategies, the simulation models must be executed much faster than real time. In addition, the employed model must be initiated to the current state of the system and the models must explicitly consider the same control inputs that are used by the manager in order to influence the operation of the actual system.

The training application attempts to bring the human into the loop. That is, people must be able to interact with simulations in a manner that closely replicates the way that they will interface with actual systems. Like system management applications, the models employed must explicitly consider the control inputs that can be managed by a person in order to influence the behavior of the actual systems. Simulation models employed in training applications typically operate at real-time. Furthermore, there is usually a need to effectively render the environment in which the managed system operates in order to provide the external stimulus to which the trainee must respond. Given the real-time nature of the simulation, training simulations are often hybrid in nature, containing both continuous state and discrete-event elements within the model. In many cases, these training simulations must also interface with physical devices.

The question remains as to whether a single simulation model can be employed in all three applications. Even if it is possible to do so, there is uncertainty as to whether the use of a single model represents the ideal approach. The future will provide more research and development in each of these application regimes. The question is: what directions will this future research and development take? In order to answer this question, we must consider the role and the needs of each of the stakeholders. We will limit our discussion to five stakeholders: the simulation software developer/vendor, the corporate end-user, the government end-user, the researcher and the educator.

3 THE NEEDS FROM THE STAKEHOLDER'S PERSPECTIVE

3.1 The Software Vendor/Developer

Perhaps the most important stakeholder in terms of providing a viable environment for the advancement of simulation is the simulation software developer/vendor. In general, these developer/vendors play two roles. The first role is to transfer the theoretical developments derived from simulation research into useful software that will allow these technologies to be implemented by an end-user. The developer/vendor's second role is to assist the end-users in modeling their systems. Sometimes the software developer/vendor also engages in research. However, the developer/vendor's research contributions are typically secondary to its contributions as a software provider and consultant.

Because most developer/vendors are for-profit corporations, they must achieve two primary goals: to remain viable and to show a profit. Given limited resources, the developer/vendor always faces tradeoffs between the allocation of resources for software development and application consulting. The usual compromise is to provide a simulation tool with sufficient unique features in order to allow the developer/vendor to capture a share of the market in simulation software. Given the market, the vendor may then provide consulting assistance on the software to the customer. It is important to note that developing software represents an up-front investment. Once the software is generated, it can be marketed for a profit and the developer/vendor's personnel can take on a consulting role that also generates revenue and profit. Hence, investments in software development must be justified economically in terms of the marginal increases of revenue that will emerge after the software is generated. Although most developer/vendors employ the same modeling paradigm for the discrete-event system, (i.e. the stochastic queuing network), there is a desire on the developer/vendor's part to capture and retain a customer. Therefore, each developer/vendor employs its own proprietary modeling elements that usually prevent a simulation model developed under its simulation language from being executed with another developer/vendor's simulation package. Furthermore, it is essential that a developer/vendor insure that a new product can execute models that were developed with its old product. If the customer must reprogram its models to employ a new product, then the customer might easily consider another developer/vendor.

Few software developer/vendors are willing to develop software at the leading (or bleeding) edge of technological development. Given that software development is an investment, it is almost impossible for the developer/vendor to insure that a profitable market will exist for the finished product. The development of experimental simulation tools resides with the researcher. In many cases, the development of a new software capability by a research group results in the group becoming a developer/vendor for its new product. Over time, the other developer/vendors will incorporate these new capabilities as they update their products. However, when a new developer/vendor first comes on the scene, the developer/vendor will likely employ its own proprietary modeling elements in order to capture and retain a share of the market.

Most commercial simulation software addresses the system design function. Some developer/vendors have made limited attempts to provide auxiliary planning capabilities such as scheduling tools that can assist in management functions. These auxiliary tools have limited on-line analysis capability, however, and their planning capability can be best described as primitive (see Davis 1998).

Most simulation software developer/vendors do receive some income from the sales of product to small corporations, government agencies and educational institutions. However, the majority portion of the developer/vendor's income is usually derived from interactions with large corporations that can afford many copies of the software and which are likely to employ the developer/vendor's consulting services. In the past decade, a new type of software developer/vendor has begun to market advanced planning tools for demand and supply chain management. Many of these developer/vendors employ their own proprietary definitions and algorithms. Some vendors even provide their own database systems. The goal of these developer/vendors is, once again, to become the exclusive software provider for each customer. The problem here is that most of these planning tool developer/vendors are much bigger than the typical simulation software developer/vendor. This implies that the simulation software/developer must usually compete against a much larger competitor for a given corporation's business. Usually, the investments that a large corporation is likely to make with such planning tool developer/vendors are usually orders of magnitude greater than the investment made with a simulation software developer/vendor.

It does not now appear that the developer/vendors of management software have realized the expanded capabilities that on-line simulation can provide for their planning and control software. In truth, there is uncertainty about how they are addressing planning and control problems because their algorithms are typically regarded as proprietary information. When they do realize the potential that on-line simulation can provide in solving management problems, we can expect that the management software developer/vendors will become major players in the development of the simulation software. They may simply purchase a current simulation software developer/vendor in order to secure a simulation capability.

The concern here is the mismatch in size between the typical developer/vendor of management software and the developer/vendor of simulation software. Both are using proprietary definitions for system variables and both rely primarily upon the large corporate customer in order to remain profitable. If the developer/vendor of management software can provide an effective simulation capability, (which most can easily do), then the developer/vendor of simulation software could easily be prevented from selling to the management software's client. Why would the client want to buy a separate simulation software package when the developer/vendor of the management software package that the client has adopted already provides a simulation solution that will fully integrate with the other management software modules and databases?

Can the simulation software developer/vendor survive when its access to the large corporate market is reduced? If not, what are the consequences upon other stakeholders?

3.2 The Corporate End-User

With the corporate end-user, the primary use of simulation has been in the design of manufacturing, logistic and business systems. In this design role, the corporate end-user typically relies upon the commercial simulation software developer/vendor to provide both the modeling tool and consulting assistance. Given the proprietary nature of the simulation tools, most companies are committed to one or a few developer/vendors. The simulation models employed in the design process are seldom validated. Furthermore, since there is limited use for these simulation models in managing the systems, these "throw-away" models are usually set aside after the system is placed into operation.

Unfortunately, there is often a major discrepancy between the projected performance and the performance achieved when the system is placed into production. This has caused many managers to distrust simulations. One particular area of concern for managers has been flexible manufacturing. Although most commercial simulation tools were designed to model such systems, it is not uncommon for these tools to severely overestimate the production characteristics of these systems. When the systems were placed into operation, many failed to achieve their designed performance. The mismatch between the simulated and the realized performance was a major contributor to the demise of flexible manufacturing.

Several reasons can be cited with regard to the inability of the current simulation tools to accurately project the performance of these manufacturing systems (see Davis 1998). Ironically, developer/vendor focus upon design rather than the management of these systems is probably the greatest contributor. The conventional modeling paradigm that employs stochastic queuing networks simply cannot address the management concerns associated with the operation of these complex systems. In particular, this paradigm does not account for the consequences that the control architectures (needed to manage these systems) have upon the system. More importantly, simulation tools employing this paradigm cannot interact with this control architecture. Furthermore, most available simulation tools cannot even be easily initiated to a current system state because the proprietary definition for system's state variables employed by the modeling tool are inaccessible to the modeler. Even if these were made accessible, in many cases, the model's variables do not conform to actual system state variables.

Because the current simulation tools provide little assistance to the management of the systems, large corporations are now turning to other developer/vendors for management software tools to manage their systems. Another concern for all corporations is the integration of different subsystems in order to coordinate the entire enterprise. Once again, the current simulation tools and modeling paradigms are woefully inadequate. In most cases, it is impossible to integrate models for two or more distinct subsystems into a single model for the composite system. The reasons for this limitation are discussed in Davis (1998). Available simulation tools simply do not meet the present and future corporate needs.

3.3 The Government End-User

The government end-user also employs conventional simulation tools to model many of its systems. However, the major investments in simulation at the government level occur in developing defense simulation capabilities and supporting basic simulation research. Due to their specialized nature, defense simulations are usually coded as individual programs by a contracted system consultant. The government end-users are pushing the technological envelope to advance the training capabilities that simulation can provide. Government's desire is to allow separate models to be integrated into more complex systems has led to the development of the High Level Architecture (HLA).

Recently, the HLA was adopted by the Object Modeling Group (who were responsible for the development of CORBA) as the standard for distributed simulations. Hence, the HLA will likely have an impact upon the entire simulation environment. More importantly, the Pentagon has mandated that all future simulation models must be HLA-compliant. There are some real concerns here, particularly as the military launches broad new simulation initiatives such as Simulation-Based Acquisition. It is unclear, for example, whether the HLA will support comprehensive modeling activities. There are also concerns relating to the potential of implementing on-line simulations within the HLA framework. Davis (1999) discusses many of these concerns in detail.

The government is also the primary sponsor of simulation research. We can expect that the government will continue to support the basic research in the more conventional simulation areas such as input and output analyses. Government funding will be critical to the advancement of new simulation technologies such as web-based simulation, distributed simulation and on-line simulation. At some point in the future, it is also likely that simulation research will be integrated with other ongoing efforts to develop distributed intelligent control architectures for complex large-scale systems. For example, the National Science Foundation is already funding major programs in knowledge-based, distributed intelligence systems. This research addresses many of the same systems now being considered by simulation researchers. Nevertheless, only a few collaborations exist between simulation researchers and other system researchers.

3.4 The Researcher

We now consider the fourth stakeholder, the researcher. The researcher provides the future direction for the development of the simulation area as a whole. Today, most of the funded and published research continues to relate to simulation needs for system design and training.

Perhaps the newest technologies are web-based and distributed simulations are focused on these two simulation problems. Web-based simulations currently address three primary areas. First, some applications provide an interface that will allow the remote user to interact with a simulation that is executed at the server for the web-site. Such simulation environments can be especially useful in a training situation. Second, other applications allow one to download a pre-programmed simulation model as a Java applet, which can then be executed at the remote web site. The next logical step is to download a Java-based simulation language, which the remote user can then use to program simulation model. For an up-to-date summary of web-based simulation, the reader is referred to the July 1999 (Vol. 73, No. 1) issue of *Simulation*, which is dedicated to this topic.

The other major, more established research thrust addresses distributed simulation. Distributed simulation technologies were initially developed as a means to execute extremely large models using concurrent processing technologies. This parallels pa the development of decomposition algorithms for solving large-scale mathematical programming problems in Sixties and Seventies. These technologies were very important when the memory and processing speed of the available computers were more limited. However, as computers became larger and faster and as new numerical analysis procedures were developed for performing numerical procedures upon large-data structures, the need for decomposition algorithms declined.

The same pattern is true for distributed simulations. Over a decade of research has been devoted to this topic. Today's personal computers are as powerful as yesterday's supercomputers. One giga-hertz personal computers should soon appear. Like decomposition procedures in mathematical programming, the increases in computational capabilities have lessened the need for distributed simulation. There will always be problems that will demand distributed simulation techniques, but problems that previously could not be simulated efficiently on a single computer can now be simulated on a personal computer.

After computational advances lessened the need for decomposition algorithms to solve large mathematical programming problems, new efforts were initiated to apply these decomposition algorithms to model distributed (hierarchical) planning situations. For several reasons that are beyond the scope of this paper, these efforts were not successful. See BenAfia and Davis (1986). In a parallel evolutionary pattern, distributed simulation techniques are now being employed as a means to integrate distinct simulation models into more comprehensive models. For example, a primary goal for the development of the HLA by the Department of Defense was to allow individual simulation models to operate together. In particular, it was desirable to allow simulation models that had been written under old programming paradigms (e.g. structured programs in FORTRAN) to interact with models that were programmed under a new paradigm (e.g. object-oriented programs in C++). Although the HLA has met the original design goals, it still does not provide a good framework for modeling distributed planning and control systems. (See Davis [1999] for an expanded discussion of HLA's limitations). Remember that the HLA paradigm was developed to save existing models, not as the ideal framework for future simulation needs. That is, the HLA has saved the past, but does not promote the future. For this reason, I have substantial concerns regarding the adoption of the HLA as the standard for distributed simulations.

Unfortunately, the inertia of the past pervades the current simulation research arena. Please note that I am not saying that these traditional topics should not be considered, but my concern is that authorities in these traditional areas are not bringing their expertise to bear upon evolving research needs. As stated earlier, there have been recent funding initiatives by the National Science Foundation and others to consider large-scale knowledge-based and distributed-intelligence systems. The systems considered here are comprised of many of the same systems that simulationists have considered for decades. These are discrete-event or hybrid systems. The proposed research must rely upon simulation technologies to model and analyze the systems. Unfortunately, most of our current modeling paradigms and off-line simulation technologies are not appropriate for the distributed on-line planning and control needs that must be considered to truly manage these systems. (See Davis 1998). New modeling paradigms and new on-line simulation technologies must be and will be developed. The question is whether the traditional simulation community participate or whether a whole new generation of simulationists/system analysts will emerge. The utility of simulation is derived for solving system design and management problems for the real world. The researchers who address the evolving system needs will necessarily direct future developments in simulation.

3.5 The Educator and Student

The final stakeholder is the educator and student. Again, most simulation textbooks have adopted a similar format. Most focus either upon the use of a given simulation tool or upon the teaching ofthe basic principles associated with constructing a simulation model under the classical stochastic queuing network paradigm and the statistical issues associated with doing off-line simulation analyses. Most major universities also offer graduate courses on specialized simulation topics such as distributed simulation, web-base simulation or interfacing issues. New course topics are needed. Fortunately, the new computer/networking technologies do permit educators to establish course development environments that were not possible a decade ago. Today, professors from different specialties, including areas outside the traditional simulation community, and institutions can collaborate to develop courses upon the web that all can use. Such courses can benefit the academic community, the practitioner and the researcher.

4 FUTURE NEEDS AND DIRECTIONS

The advancements in computer and networking technologies will provide new simulation opportunities and capabilities that few can imagine. Who thought about web-based simulation ten years ago? The technology landscape is changing fast, and simulation will exploit these advancements. I believe that these future developments in simulation must address the following concerns:

- Few systems operate as independent systems. Most systems are subsystems within larger systems. In many cases, a given subsystem also represents a system of subsystems.
- Most complex systems evolve through the execution of tasks. High level tasks are defined by most aggregate subsystems. These tasks are then redefined using a task decomposition process into more detailed subtasks that are executed at the subordinate subsystems within a given system.
- The decomposition process can be employed recursively across several layers of subsystems within other subsystems.
- Multi-resolutional architectures naturally evolve where subsystems within a given subsystem address subtasks in more detail over a shorter time horizon.

- Each subsystem executes its assigned task in the following manner:

  - First, it allocates a subsystem to execute each decomposed subtask,
  - Then, it schedules the task for execution at the assigned subsystem,
  - Also, it insures that the additional resources and entities that are needed to execute the subtask are positioned at the assigned subsystem at the scheduled execution time, and
  - Finally, it monitors the progress of the assigned subsystem as the subtasks are executed.

- The primary exception to the above situation occurs when a subsystem cannot be further decomposed into other subsystems. In this case, the subsystem executes its assigned task itself.
- The system-of-systems nature associated with the recursive task decomposition/execution process inherently implies that planning and control must be distributed. Additionally, it is essential that each subsystem address its planning, control and execution functions in an integrated fashion.
- On-line planning and control functions for each subsystem usually require human intervention. This need is likely to extend into the future.

Unfortunately, the following limitations currently prevent one from meeting the aforementioned needs:

- No effective modeling paradigm has been implemented that can capture the multi-resolutional nature of these systems as they implement the task decomposition/execution processes.
- No effective algorithms exist to distribute the planning and control functions among the included subsystems.
- The existing planning and control procedures are ideal for task-oriented problem formulations.
- Planning and control are still considered as separate technologies and have not truly been considered in an integrated fashion.
- Current simulation analysis techniques address an off-line planning and control environment, not the on-line environment where real-time management decisions are made.
- No established procedures exist that will allow people to effectively interface with the real-time data that will be generated by these on-line management systems, and the most effective roles for humans in the execution of these systems are not known.

These needs will be addressed in the future, and their satisfaction will govern the future advancements in simulation, planning and control. However, this research is too large for researchers in a single technological area to address. A multi-disciplinary approach is essential. Simulationists who look beyond today's technologies are critical participants for these teams. At my advanced tutorial, I plan to generate several new technologies that are being developed in order to address the above concerns. These employ web-bases and distributed simulation technologies and provide direct interfaces for more comprehensive planning and control needs.

Based upon our discoveries in doing this research and upon my interactions with the manufacturing community, I will conclude with my beliefs about that which can be achieved within the next decade with respect to modeling and analyses of manufacturing systems. Here are my projections (assuming the above needs can be met):

- A single modeling paradigm will be adopted to capture the task-execution and multi-resolutional nature of complex manufacturing systems.
- Through the use of this modeling paradigm, each vendor of manufacturing-related equipment will maintain a current model for each piece of equipment that it markets. This model will be written in a language such as Java and placed upon the web.
- Simulationists, manufacturers and equipment vendors will collaborate to define a set of virtual manufacturing processes. A common instruction language, like Java, will be provided for specifying the tasks that can be executed upon these virtual processes.
- Each equipment vendor will provide a means to allow its equipment to execute tasks programmed in this common instruction language. In addition, the vendor will provide virtual processing software for each piece of equipment that animates the equipment's actions as it implements a set of specified processing instructions. The virtual processing tool will be able to take a solid CAD model for each physical part inputted in process. The CAD model for the processed part will then be generated and compared against the

desired outcome as specified by the design engineer. The virtual processing tool will interface with other planning tools and will assist the manufacturing engineer to improve the processing plan and to select the proper tools, end effectors and fixtures.

- The vendors of simulation tools will provide a design environment that will allow the designer to download models for the desired equipment from the equipment vendor's web site. The simulation tool will then allow the user to do the facility layout with the selected equipment. Direct interfaces to facility planning tools will be provided.
- The new simulation software will immediately assist the system designer in specifying the control modules that will be needed to coordinate operations among the included equipment. The generated control modules will immediately provide an object-oriented model that will support simulation analyses as well as manage the real-time equipment operations.
- The model will also represent a programmed object that can be included within a more comprehensive model for the manufacturing facility. The physical layout for the new system will be incorporated directly into the layout for an entire manufacturing facility within which the designed system will reside. The control structure for the system will also interface directly with higher level controllers that are responsible for the coordination of the various manufacturing systems that comprise the overall manufacturing facility.
- Processing plans developed under the common instruction language can be included directly into the plant's database. Typically, the process plans will be hierarchically classified into more generic workstation instructions, some of which may be used upon more than one part type. A set of available equipment for executing each workstation instruction will also be specified.
- With a consideration of the equipment that is employed to transport materials among the processing equipment, the simulation tool will then generate additional instructions that are needed to support the processing tasks that occur at the defined workstations.
- The entire set of models and supporting databases will be easy to maintain and update. The system designer will be able to change equipment configurations as well as to modify and to enter new process plans with ease.
- The generated system/control models will interface immediately with on-line scheduling and management tools. Workstation and system managers can immediately interact. These same on-line management tools will further interface with other tools within the management execution system, including manufacturing resource planners, long-term production planners, and supply chain managers.
- The interfaces to the planning tools will permit one to switch from one planning tool to another with little programming effort.
- These management tools will continue to evolve as new planning and control technologies that can better incorporate evolving on-line simulation capabilities as they are developed.

REFERENCES

BenAfia, K. and Davis, W. J. 1986. The Hierarchical Search Model: An Experimental Tool for Programming Decision-Making Hierarchies. *Journal for Large Scale Systems*, 11(3), 243-265.

Davis, W. J. 1998. On-Line Simulation: Need and Evolving Research Requirements. In *Handbook of Simulation*, ed. J. Banks, 465-516. New York: John Wiley and Sons, Inc.

Davis, W, J. 1998. Looking into the Future of Simulation. *IIE Solutions* (special issue on simulation), 30(5), 24-30.

Davis, W. J. and G. Moeller. 1999. The High Level Architecture: Is There a Better Way? To appear in the *Proc. of the 1999 Winter Simulation Conference*.

AUTHOR BIOGRAPHY

WAYNE J. DAVIS is a professor of General Engineering at the University of Illinois at Urbana-Champaign. His research addresses the distributed intelligent control architectures for complex systems. To support this development, he has developed several new modeling paradigms and on-line simulation approaches.