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

WHAT DOES INDUSTRY NEED FROM SIMULATION VENDORS IN Y2K AND AFTER? A PANEL DISCUSSION

CHAIR

Jerry Banks

AutoSimulations, Inc.
Marietta, GA 30067 U.S.A

ABSTRACT

Panelists respond to the question, "What does industry need from simulation vendors in Y2k and after?" The panelists include software vendors, simulation modelers from industry, simulation consulting, and academia.

1 INTRODUCTION

For the past two years, a panel similar to this one was convened with the objective of forecasting the future of simulation software in five areas selected by the Chair. The panelists were all simulation vendors. However, the Chair's ability to see even one year in the future of simulation software was only about 40% accurate.

So, it was decided to change from a push system (with vendors indicating what was coming in the designated areas) to a pull system (where the panel indicates what is needed in simulation software). Another difference in the panel is its makeup. Two panelists are software vendors, two are from industry, one is a consultant, and one is an academic.

2 JAMES O. HENRIKSEN, Wolverine Software Corporation

This panel session provides a forum for users of simulation software to express what they'd like to see in the future from simulation software vendors. I am participating in this session as an invited vendor. Accordingly, I expect to be more on the receiving end than on the giving end in the discussions this session will generate. Nevertheless, I am a user of my own software, so I have a wish list of my own. Furthermore, over the years, I've talked with many users of, and perhaps more importantly, many *potential* users of, my software. Thus, I have a pretty good feel for the kinds of questions people ask when they consider the merits of various simulation software packages. The paragraphs which follow present my own biased opinion of the factors simulation software users should consider when selecting a new package or seeking improvements in existing software.

2.1 90% Syndrome

Beware of the 90% syndrome defined as follows: You have an application, and you select software that seems ideally suited to the task. You start using the software and make rapid progress to the point where your model is 90% complete. At this point, you discover a few of your requirements for which there is no built-in, higher-level functionality, so you have to use the features which do exist in unusual combinations to achieve the desired effects. This may get you to the 92-93% point. As time goes on, you discover more and more requirements that are not fulfilled by the software. Eventually, you may conclude that a 100% solution falls outside the capabilities of the software, in which case you must either (1) change tools, or (2) live with a 95% solution.

Simulation software should provide straightforward recourse when its "normal" building blocks are inadequate. This is not a new problem. For example, over thirty years ago, the GPSS language provided the HELP block for calling Fortran and/or assembly language programs to perform functions not easily carried out in GPSS. The major problems with this approach are that (1) cross-language interfaces are tricky, and (2) if you have to manipulate simulation package data from outside the package, in a "foreign" language, you must understand implementation details, and you run the risk of corrupting the run-time environment. Tools that provide recourse to lower levels within the same package are preferable to those that do not.

2.2 Realistic and Long Term

Take a realistic, long-term view of ease of use. Software ease of use is not constant over the course of a project.

The ease of use of one package may be high in the early stages of a project, but degrade as the project goes on. Consider, for example, software that offers only a graphical model-building paradigm. With such a tool, one should be able to construct a model comprising 50 building blocks fairly easily. All the blocks can fit on a single screen, and a pictorial representation is easy to understand. On the other hand, if the model eventually will require 5,000 building blocks, they can't all fit on the screen, so one must group the blocks into hierarchies, revealing and hiding detail, as appropriate. A large amount of time can be spent panning and zooming through the model representation. As model size increases, this becomes more and more of a problem. The robustness of internal algorithms can also contribute to size-related degradation. A package with naive event list management may work fine in simple queueing models with only 50 simultaneous activities; however, they can completely collapse when used to model telephone switching equipment, where at any given time, there may be 500,000 simultaneously active calls. Other packages may be harder to use in the beginning, but provide a payoff once its techniques are mastered.

The key to evaluating ease-of-use is project duration. If you're doing a 3-week project, software with high day one ease-of-use is preferable. Packages which are harder to learn may have more than a 3-week learning curve. Conversely, if you're doing a 6-month project, or if you're doing the same kinds of projects repeatedly, a package's learning curve can be amortized over a longer period. You may be able to devote some time to building your own toolkit that can be reused.

2.3 One Vendor?

Don't expect to get everything from one vendor. Look for packages that are good at performing their specialties and embrace industry standards for communicating with other software for performing functions outside their area of specialty. For example, Wolverine software offers a library version (a Windows DLL) of its Proof Animation package. The library contains about a half-dozen procedures, two of which are predominantly used. Any software that can call a DLL can be used to drive the library version of Proof. This enables the straightforward addition of concurrent, or even real-time animation to simulation software that lacks this capability. Other vendors have gone to great lengths to provide built-in interfaces to tools such as Visual Basic, greatly extending their "reach."

2.4 Software Pricing

Have realistic expectations about software pricing; you get what you pay for. The advent of the personal computer has placed demands on vendors for simultaneously increasing functionality and decreasing prices of their software. If this trend continues unabated, the result will be that simulation software will be provided only by firms which are large enough to amortize development costs over a large number of sales. A traditional strength of the simulation community has been that its software has been produced by small companies who are in touch with their users. Do you think it would be a good idea if Microsoft takes over the simulation software marketplace? This may be where we are headed.

2.5 Talk to Vendors

Talk to your software vendors. Let them know what you want, what you like, and what you dislike. Most vendors maintain a "wish list." You should be able to get your requests on their wish list. However, be patient. An idea that is allowed to percolate for six months and to be amplified and amended by requests from other users will yield a better solution than that provided by an immediate response to a single user.

3 RICKI G. INGALLS, Compaq Computer Corporation

When I was asked to participate in this forum, I thought that it was great that I would be asked to provide a vision for the future of simulation. However, you would think that people who create the software would be the ones casting the vision for the future. To have the vendors conduct this session would be more like Bill Gates talking about the future of operating systems or Bill Clinton talking about the future of the government or Jack Welch talking about the future of business. What I have been asked to do is more like being a journalist or critic. It would be like Spencer F. Katt talking about the future of operating systems or Matt Drudge talking about the future of government or Scott Adams talking about the future of business. If Gates, Clinton, and Welch are wrong, they ruin their careers. If Katt, Drudge, or Adams are wrong, they simply make some excuse like "I guess I blew that one" and life goes on. Still, it does not keep the latter group from talking about the future and they still make a decent living. Honestly, it is not a bad group to be in.

So as an involved outsider, I believe that the discrete-event simulation (DES) industry has a bright future if it can put its past behind it. In a nutshell, the DES industry has been primarily based on languages or systems that have been based on the general problem of discrete-event simulation. Some have specialized in certain activities, such as manufacturing, but no one has been able to capitalize on a major business problem. In a related industry, optimization, the fabulous growth of i2 shows what can be done if an OR technology is properly deployed to critical business issues. DES has a special problem, in that the model would not give a single answer to the user, but I believe the issue can be overcome if the output is properly packaged and it addresses corporate financial issues.

On the languages, systems, and user interaction front, the DES industry must migrate to systems that have a clean, natural interface with the primary business software used for decision making. On the desktop software side, a seamless interface with Microsoft Excel would be a wonderful addition to any simulation package. On the systems software side, interfaces with SAP, Oracle Manufacturing, Aspen, i2, and/or Manugistics would speed the introduction of simulation to large-scale problems. With input and output data, simulation systems should be able to read and write data directly to databases that can easily be queried by the user. For too long, simulation companies have considered themselves independent to the point of not wanting to integrate their software with that of other companies. These suggestions are small changes that would break the isolationist attitude that has hurt the growth of simulation as a viable analytical tool in business.

It would serve the DES industry well to become less visible. There are many analytical applications of simulation where the user would never need to know that a simulation was ever executed. This already happens in simulation-based scheduling and it needs to be extended to other types of analysis. The strength of current simulation systems does not lie in the user interface or data analysis capability. It does lie in its ability to model variance. The DES vendors should take full advantage of that capability and embed their software inside other companies software.

These are the future developments that will serve customers of simulation well. Furthermore, I believe that the future of simulation will include these changes because it is the natural way to move simulation into the business mainstream. Some companies will embrace this change and grow tremendously. Others will not embrace this change and they will limit their potential growth.

4 THOMAS JEFFERSON, Intel Corporation

The practice of using modeling methods, especially discrete-event simulation, is becoming more commonplace in industry today, partially as a result of the continuous improvement efforts of simulation vendors. To continue to infuse simulation analysis into decision-making processes as we approach the next millennium, the rate of product enhancements must, at a minimum, continue at its current pace. Generic areas for improvement – speed, flexibility, ease of use, and accuracy – must continue to improve in order for more detailed enhancements to be possible. Although critical, these traits are the obvious areas for improvement in simulation packages, indeed in any computing or analysis product. For simulation tools to realize their potential as decision-making tools, there are two specific development which need to occur; the concept of end-user interfaces, and the concept of interoperability and integration between multiple simulation tools and/or other machinery from multiple vendor sources.

Today, most simulation analysis is done in a local environment; that is, a stand-alone (non-networked) workstation which contains the simulation 'brain' and which also functions as the user-interface, where the model builder is also most likely the model user. Unfortunately, the modeler is usually not the person who a) best knows and understands the system being modeled, and b) is most affected by the results of the model. The true value of modeling will come when the people most knowledgeable and impacted by the system being modeled are able to interact with the model directly.

This concept is achievable with the creation of a networked system where a central (core) model resides with the model developer, and remote end user interfaces where people can provide inputs, run, and receive outputs from the core model in standardized templates and reports. To make such a system feasible financially, 'dumb' end user interfaces would be required. These devices would be devoid of any simulating capacity, as they would serve only as data input stations and would then relay the request for results to the core model. See Figure 1 for a graphical description of this concept. The scope of the expert modeler does not change in this scenario – accurate models will still need to be created, verified, and validated. However, the onus of generating output shifts to the end user (customer), allowing the modeling expert more time to focus on additional model creation.

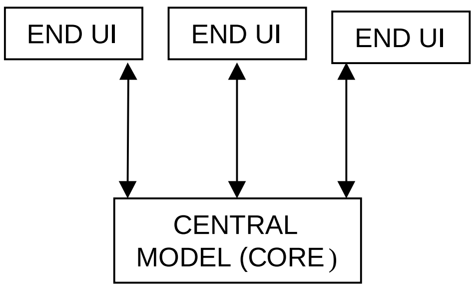

Figure 1: Graphical Example of a System with Remote User Interface

Today's market for simulation software packages is very diverse, with each product on the market having specific strengths and weaknesses. While a few general 'catch all' products do exist, most vendor offerings seem to be designed for specific modeling projects such as material handling, scheduling, process machinery, ergonomics, and even power plant operation, just to name a few. The issue with all-inclusive type packages is that usually they are not all-inclusive, lacking in one or more application areas;

conversely, the issue with application specific packages is that they do not give the user the ability to expand beyond a specific application if required.

The likeliness of one package being able to model any and all situations accurately seems remote, so the need for application-specific products should continue to exist. To maximize the effectiveness of these products, systems of interoperability, where multiple models will be linked to leverage the strengths of each specific product are needed. For example, a package whose strength is ergonomic modeling could be fused with a product whose strength is material handling to capture all important interaction effects in these two areas. Figure 2 gives a graphical description of this concept, which would be combined with the concept of multiple end user interfaces where experts in all areas being modeled could use the combined 'super model' for analysis purposes. As an extension of this concept, linking modeling engines directly to process equipment or other objects to be modeled should also be possible. A second advantage of such a system would be that simulation users would be able to select among all application specific packages available to best model a specific scenario.

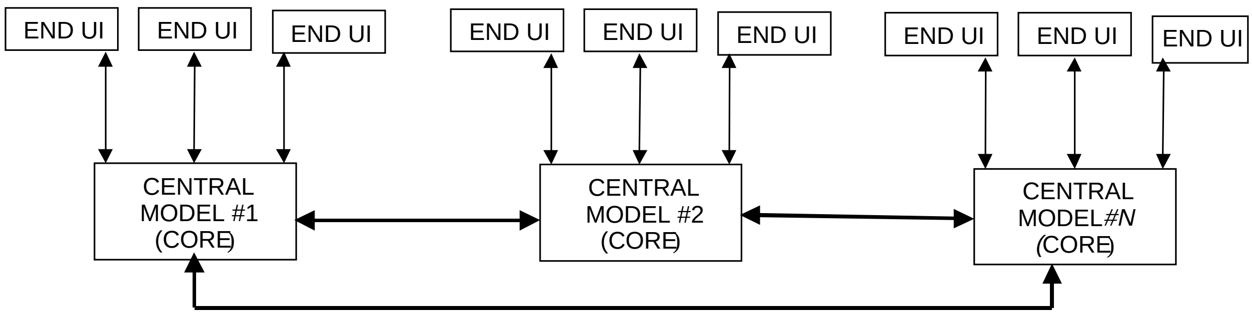

Figure 2: Graphical Depiction of Interoperable Modeling System

A first step towards the vision of a simulation system consisting of integrated central core models with multiple end-user interfaces will be the creation of common protocols for intra-product communication.

Reaching this state will require simulation vendors and companies using simulation to work together to leverage best practices and develop industry standards. The development of standards in the semiconductor industry has been hugely successful in a variety of specific areas including facilities, process equipment, software systems, and automation, resulting in lower costs for both equipment vendors and semiconductor manufacturers. In this vision, standard communication protocols will need to be developed for data input and output, transfer of data between modeling applications, and coordinating analysis between multiple packages.

In addition to the development of applicable standards, the cost of simulation tools will need to decrease for this concept to be feasible. Most simulation users will not be able to afford the capital and maintenance costs for multiple simulation engines, so the challenge to simulation vendors will be to develop innovative methods and processes to reduce cost and remain competitive. The vision proposed will provide model builders and end users with a highly efficient tool for decision-making in the 21st century. However, the technical roadblocks preventing improvement in the basic areas discussed above – model speed, flexibility, accuracy, and ease of use – must first be overcome in order to make such visions technically feasible.

5 KHALED MABROUK, The Model Builders, Inc.

The simulation industry has enjoyed an acceptable growth rate in the last five years. Even though the number of simulation users has increased tremendously from ten years ago, I believe that the recent growth rate is significantly lower than ten years ago. I believe that this is tied closely to the concept of "Crossing the Chasm" presented by Geoffrey A. Moore (1991).

Mr. Moore separates most technological innovations into those which are categorized as "continuous" innovation and those which are categorized as "discontinuous" innovation. Continuous innovation occurs for technological tools that have been widely accepted as useful and practical. Examples of continuous innovation technologies include cellular phones, microwaves, video cassette recorders, and personal computers. The buyer of a continuous innovation tool is most concerned with features and price.

Discontinuous innovation occurs for technological tools that are not widely accepted as useful and practical. Some people may believe that these tools are useful and practical, but most people don't. Most people consider these tools risky and unproven. Examples of discontinuous innovation technologies include DVD players, Betamax video cassette recorders, and palm pilots. For most people, these tools are unproven as a practical tool. Even though there are many supporters of simulation, discrete-event simulation is still a discontinuous innovation technology. There are many reason for this, and I would like to focus, for now, on how simulation vendors can help simulation "cross the chasm" from a discontinuous innovation to a continuous innovation.

First, let us review the Technology Adoption Life Cycle and how it relates to simulation. This understanding is crucial for allowing us to have an impact on simulation's ability to cross this chasm. In his book, Mr. Moore argues that with respect to most technologies, we each fall into one of five technology adoption categories. These categories are the "Technology Enthusiast," the "Visionary," the "Pragmatist," the "Conservative," and the "Laggard."

Technology Enthusiasts are crucial for getting a technology started. These are the people who jump right in every time a new technology tool becomes available. They are comfortable with the software crashing on a regular basis, as long as they can feel that they are on the leading edge of technology. The simulation industry currently has many such enthusiasts. They are constantly pushing and asking for *many* improvements to their simulation software. Thus, a simulation software vendor who is focused on pleasing this group, comes up with new feature releases on an annual basis; if not more often.

Visionaries are crucial for getting a technology established. In contrast with Technology Enthusiasts, these people tend to control significant corporate budgets. In addition, they are looking for a high risk/high reward situation to use simulation to gain a competitive advantage. They tend to want more out of the simulation software than it currently offers. Thus, the simulation software vendor who is focused on pleasing this group is constantly winning big contracts that require it to modify its software to fit the needs of these enthusiasts. Eventually, these modified versions of their software become incorporated into the basic product, or they become new independent products.

Most of the current user base of discrete event simulation tends to fall into one of these two categories. The interesting thing is that these two categories tend to represent only two and a half percent of the total possible market for a technology. The next two categories represent close to ninety-five percent of the total possible market for a technology.

Pragmatists have an IT mentality. They are looking for new technologies both to be consistently effective and to have a marginal impact on the success of their organization. They are definitely not high risk/high reward people. The technological tool must have proven itself to be effective most of, if not all of, the time. Unluckily, due to a number of reasons, simulation can not claim to have proven itself to be an effective tool most of the time. There are many reasons for this, only on some of which the simulation software vendors have an effect.

Conservatives hate to use new technology. For these people to use a new technology, it must appear as a continuous innovation. Thus, for these people to utilize simulation, it needs to be incorporated as part of another tool. For example, a scheduling tool that has discrete-event simulation embedded in it would serve this purpose. These people don't care if they are using simulation or not. They just want a tool to solve their business problem that is effective, simple, and straight forward to use.

Laggards do not use new technology. These people represent two and a half percent of the total possible market for a technology and are best left alone.

For a technology to cross the chasm, the market leaders must abandon their current user base of enthusiasts and visionaries, then change their approach so that it best fits the Pragmatists' needs. Creating a large number of success stories amongst its user base, and working hard to minimizing shelfware is a crucial step forward for any software vendor who chooses to cross the chasm. This, though, will not be enough.

A very crucial requirement for a technology to cross the chasm is that the industry settles down to two or three major players. This is necessary, from a Pragmatist perspective, since it eliminates the risk (for the Pragmatist) of choosing the wrong software vendor. Being practical people, Pragmatists do not want to choose a software product, build the infrastructure to support it, then have to change to a different product a few years later. This is too expensive in their eyes. Thus, if there are too many choices that appear to be acceptable, these people would rather not choose at all.

In no way am I advocating that only two or three software vendors remain in the market while all the other ones get bought out or go out of business. This might work, but it will not happen. On the other hand, it would make a lot of sense for the simulation software vendors to segment the simulation market out into a set of vertical niches. For each of these niches, only two or three vendors should dominate. This will make it easier for the pragmatists to make a commitment to a specific simulation tool, and, with that, the market for the simulation software vendors would increase significantly.

Even though I am confident that this philosophy will be very effective, it will be a difficult decision for most, if not all, vendors. They will need to determine not only how to best divide the market into vertical niches, but they must also choose the niche where their software will best fit.

6 GERALD T. MACKULAK, Arizona State University

Simulation languages have improved immensely over the past 20 years along with the hardware on which they run.

Unfortunately, with the increase in features has come a corresponding decrease in execution speed and increase in complexity. This has led to what I call the "magic bullet syndrome." If you buy and use 'my' magic bullet your simulation problems will forever be solved. I postulated twelve years ago that simulation was on the verge of greatly increased popularity due to ease of use improvements, cost reductions, and improved PC platforms. I now think I was very wrong. I remember seeing the annual survey of simulation software vendors ten years ago and noticing over 200 different entries. Realistically, we are now down to a few handfuls. Why?

The simulation software vendors have added features, (some extremely useful) that have led to increases in run times and in complexity. In trying to make languages easier to use, the vendors have made modeling less efficient. Fifteen years ago when a simulation model was being built, many intricate logic concepts were excluded since the languages did not support their inclusion. As a result models were built that were appropriate given the data and time available. As the languages improved in the complexity of their features and interfaces they tempt the model builder into adding more into the model than they have logic or data to support. The model builder forgets that the goal is the analysis, not the creation of the model itself.

As an academic, I want a language that is easy to learn, useful (realistic), supportive of statistical output analysis, comes with a textbook, that still lets me customize (get in there and create basic changes to the way things are modeled), and is available free off the internet. As a consultant I want a language that accurately models the components of the system that I am analyzing, lets me build an accurate model in four hours or less, has great animation, interfaces with everything Microsoft sells, runs in minutes, supports the analysis of designed experiments, and is available free off the internet! My conclusion is that the language that supports both of my lists has yet to be created. So what is realistic to expect (request?) in the future?

The software of the future needs to (in no particular order of importance):

1. Use a GUI that looks and feels like the Microsoft stuff. Most people are exposed to computers using Windows so simulation should use the same approach.
2. Aid in the accuracy of predicting system performance. The software should support statistical analysis for both input and output, DOE support, transient response prediction (How many languages just request you to input when to truncate? Does a novice really know?), run length determination, and logic accuracy (My xyz stamping machine can be modeled by selecting block type XXX.). This accuracy concept needs to occur even if the model builder doesn't realize it.
3. Reduce model run time by supporting distributed model execution without significant model builder assistance. (Most offices have multiple PC's. Why not tap that capacity if you want to run large models quickly?)
4. Interface with CAD as a dynamic system. (If you are modeling a factory, the CAD group will have located any possible equipment changes while you are beginning your analysis. Why can't this be used as input to the model and relationships, including dimensional accuracy and be automatically created within the model? It means that the language needs to support a physical system interface rather than just be logical.)
5. Embrace more of the software developments coming out of the internet support environment. Why can't intelligent agents determine the type of models that you have been building and when you next log on to build a new model give you a starting point that is 75% of the way there?
6. Why can't a model be self debugging? MS Word checks spelling and grammar. Why can't software do something similar for model builders?
7. Since design groups are possibly in different physical locations, can the language support virtual creation via internet interfaces?

7 C. DENNIS PEGDEN, Systems Modeling Corporation

Simulation is continuing to change and expand at an amazing pace. The technology is poised to move beyond a narrowly deployed tool used by highly technical and sophisticated analysts, to a widely deployed tool within the enterprise.

One of the important strengths of simulation has been its ability to adapt and be applied to a broad cross section of problems within the enterprise. As simulation expands throughout the enterprise, the range of applications is also expanding. Although manufacturing still remains a critical area of application, there are many new areas where significant investments are being made, and simulation can be used to help manage these changes. In particular, both business processes and supply chain systems are going through significant changes within many enterprises, and simulation is an ideal technology for understanding these changes. Models are used to study the entire supply chain from suppliers to final customer delivery. These are complex systems with many interacting and random elements. Users need modeling tools that can be used to model the entire supply chain within in enterprise at multiple levels of detail. These applications place significant demands on the simulation in terms of size and speed of the models.

In the past decades the focus within the simulation community has been on making it possible to model a wide range of systems. This has led to the development of very rich and powerful modeling tools. However, rich and powerful tools are by their nature complex and difficult to learn. What users need are tools that are powerful and flexible, yet very easy to learn and use. Without a doubt, the number one barrier to the broad deployment of simulation technology is the complexity of the technology. Reducing the complexity, while keeping the flexibility to accurately model a wide range of systems, remains the number one challenge from the user to the industry.

As the number and types of users expand within the enterprise, the need for scaleable simulation tools increases. The basic concept here is the notion of having products that expose only the specific functionality that a user needs for the model that is being developed. For example, if a user is modeling a simple business process for processing orders, there is no need to expose functionality to the user that is related to modeling complex manufacturing processes or for 3D visualization. On the other hand, there may be a desire to later expand the model to include the manufacturing process, and these more advanced features may be needed. The challenge is to provide a system that has all the modeling power available when needed, but only exposes the necessary features for the current modeling project.

As the number and size of simulation models increase, there will be new demands placed on simulation tools to make it easier for people to share models across the enterprise, and also collaborate on the development and maintenance of models. The Internet will clearly play a significant role in this evolution. The Internet is changing the entire information technology field, and simulation is no exception. The Internet will play an important part in building and viewing simulation models. In the future, an enterprise will maintain a knowledge base of their systems, process, and products that can be accessed across the Internet. The processes will be defined in terms of animated, simulation models that can be executed by any individual within the enterprise that has access to the system. Simulation will emerge as the preferred way of documenting and communicating processes within the enterprise.

During the past forty years, simulation has been a tool used by a small group of trained experts to model complex and expensive systems. In the future, analysts throughout the enterprise will routinely use this technology. To support this new class of user, the tools will become easier to buy, learn, and implement.

8 CONCLUSION

A wide variety of answers have been given by the panelists. Jim Henriksen gave considerations when selecting software. Ricki Ingalls indicated that success in the simulation software industry will only come when a vendor capitalizes on a major business problem. Tom Jefferson argued for the creation of an interoperable modeling system. Kal Mabrouk says that simulation is still a discontinuous innovation and that success will only occur when the technology "crosses the chasm" to continuous innovation. Jerry Mackulak gives seven criteria for software of the future. Finally, Dennis Pegden says that simulation tools will become easier to buy, learn, and implement, supporting a new class of user.

REFERENCE

Moore, G.A. 1991. *Crossing the Chasm: Marketing and selling high-tech products to mainstream customers*, HarperBusiness, New York.

AUTHOR BIOGRAPHY

JERRY BANKS is Director of Simulation Technology at AutoSimulations, Inc. in their Marietta, Georgia office. He retired from Georgia Tech as Professor of Industrial and Systems Engineering in July, 1999. He was a member of the Board of the Winter Simulation Conference for eight years and served as Chair of that body.

PANELIST BIOGRAPHIES

JAMES O. HENRIKSEN is the president of Wolverine Software Corporation. He was the chief developer of the first version of GPSS/H, of Proof Animation, and of SLX. He is a frequent contributor to the literature on simulation and has presented many papers at the Winter Simulation Conference. Mr. Henriksen has served as the Business Chair and General Chair of past Winter Simulation Conferences. He has also served on the Board of Directors of the conference as the ACM/SIGSIM representative.

RICKI G. INGALLS is a senior manager in the Global Integrated Logistics Group at Compaq Computer Corporation. He has been involved in the application and development of operational modeling tools and techniques in the electronics industry for over 15 years. In his current position, he is responsible for the strategic planning of Compaq's Global Supply Chain Operations, which includes extensive logistical and financial supply chain modeling. He has a B.S. in Mathematics from East Texas

Baptist College, a M.S. in Industrial Engineering from Texas A&M University and a Ph.D. in Management Science from The University of Texas at Austin. Prior to re-joining Compaq, he was on the technical staff of the Operational Modeling Group at SEMATECH, Manager of Operations Analysis at Compaq Computer Corporation, a consultant with the Electronics Automation Application Center of General Electric Co. and an Industrial Engineer with Motorola, Inc.

THOMAS JEFFERSON is an Industrial Engineer with the Automated Material Handling Systems Group at Intel Corporation, responsible for the project management, design, layout, and modeling of automated material handling systems. He received a B.S. in Industrial Engineering from the Rochester Institute of Technology in 1994 and joined Intel in 1995, after the completion of research at SEMATECH. Current simulation interests include material handling concepts for next-generation manufacturing, cost-effective automated material handling solutions, and semiconductor manufacturing operational methods and labor modeling.

KHALED MABROUK has been a Simulation Strategist for The Model Builders since December, 1991. He has been actively involved in the simulation industry since 1987 when he worked for Systems Modeling Corporation. His current responsibilities include assisting corporations in developing a strategy for implementing simulation for decision support, and managing simulation modeling and analysis projects. He has authored eighteen articles on the subject of simulation, is a co-founder of the Michigan Simulation User Group, and speaks often on the subject of simulation.

GERALD T. MACKULAK is an Associate Professor in the Department of Industrial Engineering at Arizona State University. His research interests center on improvements to simulation methodology, material handling modeling, and production scheduling in high technology manufacturing. He holds B.S.I.E., M.S.I.E. and Ph.D. degrees from Purdue University.

C. DENNIS PEGDEN received his bachelors in Aeronautics, Astronautics, and Engineering Sciences from Purdue University in 1970. He worked in the aerospace industry at the National Aeronautics and Space Administration and the Matrix Corporation. He returned to Purdue in 1973 and received his Ph.D. in mathematical optimization from the Industrial Engineering Department in 1976. After graduation, he taught at the University of Alabama in Huntsville where he began his work in simulation and led in the development of the SLAM simulation language. In 1979, he joined the faculty at the Pennsylvania State University where he completed the development of the SIMAN simulation language. He is currently CEO of Systems Modeling Corporation which markets SIMAN and Arena simulation products; the Tempo scheduling product; and vertical market products in the areas of call centers, business processing, manufacturing, high-speed processing, and real-time control.