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

PANEL: STRATEGIC DIRECTIONS IN SIMULATION RESEARCH

Ernest H. Page
(Moderator)

The MITRE Corporation
1820 Dolley Madison Boulevard
McLean, VA 22102, U.S.A.

David M. Nicol

Department of Computer Science
6211 Sudikoff Laboratory
Dartmouth College
Hanover, NH 03755, U.S.A.

Osman Balci

Department of Computer Science
Virginia Polytechnical Institute
& State University
Blacksburg, VA 24061, U.S.A.

Richard M. Fujimoto

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, U.S.A.

Paul A. Fishwick

Department of Computer &
Information Science and Engineering
University of Florida
CSE 301
Gainesville, FL 32611, U.S.A.

Pierre L'Ecuyer

Département d'Informatique et de
Recherche Opérationelle
Université de Montréal
C.P. 6128, Succ. Centre-Ville
Montréal, Québec, CANADA H3C 3J7

Roger Smith

BTG Inc.
3481 Woodley Park Place
Oviedo, FL 32765, U.S.A.

ABSTRACT

We consider the future directions of simulation research.

1 INTRODUCTION

To mark the 50th anniversary of the Association for Computing Machinery, Volume 28, Number 4 of *ACM Computing Surveys* was released entitled, "Strategic Directions in Computing Research." Notable—to attendees of Winter Simulation Conference at least—among the topics not covered was computer simulation.

One may reasonably ask why this is so. Are there no unanswered questions remaining in computer simulation? Is computer simulation an unimportant topic? Does simulation not have relevance as a computing discipline? Should it rather be considered solely in terms of operations research, statistics or mathematics?

Arguably computer simulation is quite relevant to technological advance in many arenas. Modeling and simulation are playing key roles within industry, academia and the government. The papers appearing in these *Proceedings* bear significant witness to that fact. The study of computer simulation as a computing discipline also seems quite appropriate and potentially valuable. Modeling—fundamental to simulation—is growing ever more central to the art and science of modern programming.

Thus, we believe strategic directions in simulation research are worthy of formulation and will provide a useful roadmap as the simulation community moves into the twenty-first century. This paper represents a collection of thoughts from a few distinguished simulationists toward the generation of such a roadmap.

2 PROBLEMS OF SCALE (DAVID NICOL)

We in the discrete-simulation world need only look over the fence at what has happened in continuous simulation to see some important issues in our future. The continuous simulation world has, without ceasing, devoured every available computing cycle to solve their models. As more and more computing power became available, their models grew in size and complexity. And—importantly—the gap between the size and scale of problems they wanted to solve with the size and scale of problems they were able to solve forced a focus on new, more efficient solution techniques.

The discrete-event world is different of course, and has historically placed emphasis on different sorts of problems. But commodity technology applied to discrete-event simulation is already able to represent models that are vastly larger and more complex than in the past, and this capability continues to grow with increasing capabilities of computer hardware. This raw horsepower opens the way for simulation to be used in new domains, e.g., control of very large systems where real-time decisions are made as a result of forecasting (through simulation) the results of various decision options. How can we effectively harness such capabilities? The issues we have to deal with include (i) how do we express a huge model, (ii) how do we validate a huge model, (iii) how do we solve a huge model, (iv) how do we trace and understand the output of a huge model?

Issues of modeling a huge system, and understanding a huge system model loom large. A fairly standard technique now is to represent a model hierarchically, with spatial aggregation (e.g., a set of gates aggregated into a component, a set of components aggregated into a device, a set of devices aggregated into a system). At one level the mechanics are easy; this sort of organization can be expressed in languages such as VHDL, C++, and Java. However this is an organization optimized for a compiler. It is not at all easy for a human to assimilate a model expressed this way, nor is it easy by looking at a model so expressed to understand subsystem interactions or relationships between events at differing time-scales. The fundamental problem is that modelers are able to manage a limited amount of complexity when developing or studying a model. The graphical modeling tools any one of a number of vendors is happy to sell you aren't up to a model of, say, the Internet. We need breakthroughs on expressing and navigating huge models. Furthermore, the increased level of complexity only exacerbates an open sore in our world, the issue of validation.

To a first approximation, the speed of a CPU and the size of a memory are increasing at the same rate, one that has been dictated by Moore's law for the last 20 years (a doubling of capacity every 18 months). There is a hidden complication however, that the amount of computation required to solve a model increases by more than a linear factor in model size. This phenomena is well-understood and much cursed in the continuous world. A simple example is the simulation of many bodies in a gravitational field. The naive solution given N bodies requires work in proportion to N^2. Even fiendishly clever techniques manage to knock the workload requirement down to something proportional to $N \log N$. Increasing memory sizes may whet our appetite for larger models, but the more-than-proportional increase in computation becomes the bottleneck. Speaking as a researcher in parallel simulation, I can say with confidence that parallel processing does not solve this problem, at best it merely delays its onset—using P processors increases computational ability only by a factor of P. Just as continuous models approach the issue with techniques of multi-resolution (in time and space), so must we. The computational advantages are tremendous. In my own experience I've used a fluid methodology to represent network traffic, and seen thousand-fold acceleration in solution time, with relatively small "error" relative to a packet-based simulation. What happens here is that the simulation is conducted at a much slower time scale, but on a model formalism that still captures certain salient features. Just as this example aggregates behavior in time, so also do modelers aggregate in space. Multi-resolution modeling clearly has huge potential to accelerate solution speed, but at present is an ad-hoc endevour. Questions remain on how to do it, how to anticipate, handle, control, or bound the difference between the results it computes and those of a detailed model, how to integrate state represented at different spatial and/or temporal resolution.

The problem of seeing and understanding the results of a large scale simulation are even more important. A simulation of millions of interacting entities will give rise to interesting behavior at levels of spatial and temporal scale. How do we detect what is "interesting"? How do we visualize so much data, at so many scales?

All of the other viewpoints expressed in this panel shed light on various challenges facing simulationists. To each and every viewpoint we can add, "This challenge becomes all the more formidable with increasing model size."

3 STRATEGIC DIRECTIONS FOR VV&A OF M&S APPLICATIONS (OSMAN BALCI)

Verification, Validation, and Accreditation (VV&A) of Modeling and Simulation (M&S) applications pose significant technical challenges for researchers, practitioners, and managers. This position statement describes some recent findings on the state of VV&A technology and presents some recommendations.

3.1 Findings on the State of VV&A Technology

1. Automation is critically needed to effectively and affordably conduct M&S VV&A. Software tools to be developed to provide the much needed automation specifically for M&S VV&A pose a significant potential for effectively and affordably conducting M&S VV&A.
2. Component-based development is an effective and affordable way of creating M&S applications and conducting M&S VV&A. A verified and validated model component can substantially decrease the M&S VV&A effort when reused thereby significantly decreasing the time and cost of development. Component-based M&S

development technology, when created, may be the "silver bullet" for effective and affordable M&S VV&A.

3. Success in the application of the VV&A technology is very much affected by employment of skilled personnel. Although billions of dollars are spent by the government annually in the use of the M&S technologies, we lack sufficient education and training in the area of M&S especially in VV&A. The availability of better educated VV&A personnel should result in more successful applications of the M&S VV&A technology.
4. Terminology continues to be a serious problem in communication. Due to the diversity of areas within the M&S community, the definitions of terms are interpreted differently, incorrectly, liberally, or loosely. Inconsistent use of M&S VV&A terminology adversely affects the development, use, and dissemination of the VV&A technology.
5. M&S VV&A technology heavily relies on the software verification, validation, and testing (VV&T) technology. Advancements in the M&S VV&A area mostly come from the software engineering discipline. This requires the M&S VV&A professionals to be knowledgeable about the software VV&T technology and the associated Computer-Aided Software Engineering tools.
6. The advancement and employment of the M&S VV&A technology are adversely affected by the lack of peer-reviewed publications. There does not seem to be any incentive or encouragement for peer-reviewed publication within the government and industry sectors. Lack of journal-quality peer-reviewed publications poses a serious problem for the advancement and dissemination of the M&S VV&A technology.
7. The advancement of the VV&A technology specifically for M&S application development can only be achieved by substantial funding provided by government agencies such as NSF, DARPA, ONR, and DMSO.

3.2 Recommendations

1. There should be substantial funding for development of software tools that provide computer-aided assistance specifically for M&S VV&A. Such assistance is critically needed to effectively and affordably conduct M&S VV&A.
2. Organizations such as DMSO, MORS, and SCS should bring awareness to the component-based development technology and advise proper government agencies to establish funding in this area. This effort is currently led by NIST and is a very hot topic in the software engineering field. NIST has an advanced technology program (ATP) on Component-Based Software (see http://www.nist.gov) that provides minimum $2 million of funding per award for the creation of component-based software development technologies. The major objective of ATP is to make software development effective and affordable. (See the ATP program objectives on the web site). Similarly, if we are to make the M&S development and VV&A activities effective and affordable, component-based development technologies for M&S must be created.
3. Organizations such as DMSO, MORS, and SCS should lead/encourage the development of educational and training programs in the area of VV&A.
4. Organizations such as DMSO, MORS, and SCS should promote the preparation and publication of case studies reporting on software tools used, techniques employed, experiences gained, lessons learned, cost data, recommendations for future work, etc.
5. Government agencies such as NSF, DARPA, ONR, and DMSO should provide substantial funding for advancing the current M&S VV&A technology. This funding should be allocated primary for basic research.

4 PARALLEL AND DISTRIBUTED SIMULATION (RICHARD FUJIMOTO)

Parallel and distributed simulation technology has reached a crossroads. In past years, researchers have long lamented the limited impact of this technology in the general modeling and simulation community, e.g., see (Fujimoto 1993). This situation has changed considerably in recent years. The latter half of the 1990's has seen parallel and distributed simulation technology flourish. Perhaps most important is its inclusion in the High Level Architecture (HLA) that has been adopted both by the U.S. Department of Defense and NATO, and is undergoing commercial standardization. Further, parallel simulation systems are used to model commercial air traffic and has impacted the design and management of air transportation systems in the U.S and Asia (Wieland 1997). Speedes (Steinman 1992), a commercial simulation system originally developed at JPL and enhanced and supported by Metron, is playing a central role in several large-scale defense simulation projects. The Aggregate Level Simulation Protocol, a fore-runner to the HLA continues to utilize algorithms originating from the parallel simulation community (Wilson and Weatherly 1994).

Plotting strategic directions for parallel and distributed simulation technology to advance as we enter the 21st century requires an understanding of the reasons behind its recent growth and adoption. We highlight two important reasons below:

- Model and software reuse. The driving force behind efforts such as the HLA is the possibility to reuse rather than reinventing large-scale simulations. It is important to note that reduced model execution time, traditionally the driving force behind academic research in parallel simulation, has not been the driving force. This is not to say performance isn't important. Indeed, scalability is often critical. However, one must keep in mind the context in which performance requirements arise.

- Transparency. Key to the success of efforts such as the HLA is the fact that simulation modelers need not be intimately familiar with details of the technology in order to utilize it. For example, the synchronization algorithm can be hidden from the modeler by a suitable application program interface (API). Transparency is essential to achieve widespread exploitation of the technology. While the HLA is not completely transparent (e.g., simulators must specify and adhere to lookahead constraints), it provides a sufficiently simple interface for modelers without parallel simulation expertise to use.

A few important research directions for this technology are briefly discussed next.

4.1 Synchronization

Synchronization will remain a core research area in the future. Synchronization algorithms can be broadly classified as conservative or optimistic (Fujimoto 1999). A key challenge in exploiting optimistic execution is maintaining transparency. Introducing rollback to an existing simulator requires a major re-engineering effort to incorporate state saving mechanisms. In addition, special primitives must be used for I/O and memory allocation to ensure that their execution can be properly rolled back. Automation of the process of exploiting optimistic execution presents an important challenge. A related problem is the complexity and amount of tuning that is often required to effectively exploit optimistic processing.

Conservative execution largely avoids many of these difficulties. However, a fundamental problem that has yet to be solved is the reliance on lookahead to achieve scalable performance. Many have attacked this problem, but only limited success has been achieved. This suggests an altogether new approach may be needed. One such approach is to relax ordering constraints. New ordering semantics and realization of scalable distributed algorithms to implement them may offer a viable solution.

Past work in synchronization has focused on execution on multiprocessor platforms. As a practical matter, platforms using standard networking hardware will continue to dominate the marketplace. While prior work in parallel and distributed simulation traditionally treats the network as a black box, significant advantage can be realized by exposing network characteristics to the simulation executive. Realization in geographically distributed computing environments such as the global Internet where large communication latencies cannot be avoided also presents new technical challenges.

4.2 Converging Application Domains

Parallel and distributed simulation research has traditionally focused on analytic simulation applications. Distributed virtual environments (DVEs) for training and entertainment have emerged as an important domain where this technology may also be applied. Historically, research in DVEs has evolved largely independent of the parallel and distributed simulation community, coming largely from the Internet, computer gaming, and the military training communities. Efforts such as the HLA highlight the utility of supporting interoperability of training and analytic simulations. But different paradigms are typically used in these domains. Approaches to unify these domains are needed.

The confluence of analysis and virtual environment simulations presents new challenges. Training and entertainment have different requirements than analytic simulations. For example, maintaining precise time stamp order processing of events is often not essential because humans may not be able to perceive "incorrect" orderings of events. On the other hand, real-time execution of the simulation is essential. Relatively little attention to date has been paid in the parallel simulation community to the problem of ensuring timely delivery of results. Parallel and distributed simulation techniques have much to offer in addressing problems such as repeatability and correct ordering of causally related events that arise in DVEs.

4.3 Heterogeneous Distributed Simulations

Most of the work thus far in the parallel simulation community has focused on homogeneous simulations where the entire simulation is built from scratch using a parallel simulation language or library. Federated simulations composed of a heterogeneous collection of models and software present new challenges. For example, publication/subscription mechanisms are used to route data because the publisher cannot easily determine what other simulators should receive its messages. By contrast, traditional parallel simulators typically assume the sender is responsible for enumerating all destinations that are to receive each event.

More generally, fundamental issues concerning composability have yet to be addressed by the modeling and simulation community. For example, how should one develop a new simulation now in order to facilitate its later reuse in another simulations that cannot be foreseen today? How much standardization is required to achieve "plug 'n play" composability? What does composability mean, given that different degrees of composability are sufficient for different applications? There are just a few of many questions that have yet to be resolved.

5 FROM MODELING TO PROGRAMMING (PAUL FISHWICK)

Modeling represents a way to communicate to your colleagues and to the public at large. There are many types of models, from scale models to mathematical models and each type has its benefits and drawbacks. In the brief amount of space that I am allotted, I hope to convince you that what we (as a community) do in simulation modeling will lay the groundwork for the future of computation. The simulation community, instead of being viewed as an entity on the outskirts of just about every discipline and "a method of last resort," is about to become the keystone for computer programming and computing at large. It is not that I am trying to demonstrate this breakthrough as a complete paradigm shift, for it is not. We are already moving along the path that leads us to this convergence, and as a technical community, we need to start using binoculars so that we can achieve better path planning.

It all begins with a brief survey of the landscape of computing and programming languages. It is only during the past decade, despite the early pioneering work done on the Simula Project, that software engineering has become significantly object-oriented. Simula most likely did not achieve widespread prominence due to the lack of an accompanying graphical language. While this may seem trivial at first, the graphical aspect is important since it allows humans to better interface to programs. Software already assumes the appearance of models if one studies the Unified Modeling Language (UML) and its predecessors. The average simulationist may be delightfully shocked to discover models of banks, libraries, ATM machines and air conditioning systems in the more recent software engineering books. There are two main reasons for this movement of program to model: 1) with CPUs inhabiting the cracks and crevices of every physical device, software is forced to behave as a model: there are separate components and communication among components; and 2) the object-oriented paradigm suggests a more physical approach to programming where physical items, with their attributes and behaviors, are surfaced into the language. The use of physical metaphors, as with Simulated Annealing, Neural Networks, and Genetic Programs also suggests a convergence: a program becomes a model of a hypothetical physical construct. The metaphor is turned into the program.

If we can agree that programming is beginning to converge to modeling, we have to question our role as simulationists in this convergence. Do the software engineers need or want what we can deliver? We know how to model. Although we may not always agree on what a model constitutes, we know that there are several levels of abstraction and many ways to build a model. Beyond that, we tend to treat modeling as an art. It is an art, but we are only beginning to realize this in any of our formulations. In a typical computer simulation conference, such as the Winter Simulation Conference (WSC), we see many examples of manufacturing assembly line process models. To be more accurate, we do not see the dynamic models, but we do see the shape or geometry of units rolling along belts and being operated upon by avatars and numerically-controlled machines. Let's ponder this commonly found simulation application, the task of modeling, and the task of programming to see what comes out of the mix. What if the manufacturing floor model were to become a program? We would recognize that a program has to be constructed to create the 3D animated assembly line, but this needs to be turned on its head to see the other possibility where the assembly line is used to design a program. What if we used the assembly lines to move "program data" around and machines to "process data?" This sort of mapping requires a strong interest in the application of metaphor, and an interest in aesthetics. Modeling is more of an art than may be readily apparent.

At the University of Florida, we have begun an investigation into the use of modeling in programming by first beginning with the Virtual Reality Modeling Language (VRML) as a starting place. VRML is an open specification and most of the software connected with VRML is now open-source. Even if we consider an ordinary model type such as a Finite State Machine or Petri Net, we can reuse 3D components from the web in order to create these models, and then place the models in the same physical space as the objects being modeled. A model browser, utilizing a parser generator in Java, is being used to permit easy browsing of objects and their models. The use of the prototype construct and Java/Javascript script nodes in VRML are essential in making the vision a reality. The transition from using these sorts of 3D models to "programming in 3D" is a smooth one involving the creative use of metaphor where it is needed. With our present indoctrination of programs as glorified mathematical expressions, it may take some time to build the correct infrastructure to make this sort of programming efficient. There are many serious issues to be addressed, and the 3D approach is not without its faults, but we feel that we are wandering into unknown, fresh territory with many possibilities for strengthening the uses of simulation in programming.

6 RANDOM NUMBERS (PIERRE L'ECUYER)

6.1 Random Number Generation

Statistical analyses of stochastic simulations are based on the assumption that the software can generate streams of independent random variables with specified distributions.

However, no simulation software does that. The random number generators (RNGs) are all *fake*; they are simple deterministic programs made to deceive the users. So what are the statistical meanings of simulation results? How do we know if the RNG has passed "enough tests" to provide reliable results for all our problems? The answer to the last question is: *We never know.* So how do we measure the quality of RNGs? The definition of quality criteria must be subjective and heuristic to some extent. Then, analyzing specific RNGs with respect to these criteria requires powerful mathematical tools.

Current research on RNGs attempts to address these issues, at least partially, both from a purely theoretical viewpoint, e.g., via asymptotic analysis and complexity theory, and from the pragmatic perspective, by proposing concrete solutions for today. Building an RNG that passes all statistical tests (or that gives the correct output distribution for all simulation problems, which is equivalent) is known to be impossible. In fact, one can show that all RNGs pass the same number of tests of a given size. The difference between the good and bad RNGs is simply that the good ones fail only fairly complicated tests, which correspond to models that are very unlikely to occur in practice. Research is still needed to better understand the following (interrelated) questions, among others:

1. How should we define, concretely, the quality criteria that RNGs must satisfy to make sure that they do not fail "too simple" tests, and under the constraint that the corresponding figures of merit must be easy to compute?
2. For the popular classes of RNGs used in practice (e.g., linear congruential, multiple recursive, shift-register, etc.), what should be the criteria for choosing the parameters and what should be the minimal period length to make sure that the structure of the RNG is not detected by simple tests? What kind of structure in a simulation problem could dangerously interact with the structure of these RNG?
3. Certain types of nonlinear RNGs tend to do better in the tests than the linear ones, but are significantly slower. However, combining a large linear recurrence with a small nonlinear one could provide the best of both worlds, if done properly. This area needs investigation.
4. Additional issues arise when several RNGs (or random number streams) are needed for parallel computations, or for different parts of a simulation model on a single computer. What is the best way to provide such streams, how should we measure the dependence between them, and should this affect the selection criteria for RNG?
5. Concrete random number packages, based on well-designed and well-tested backbone RNGs, must be made available for all major programming languages and software environments, in order to replace the cheap, simplistic, and bad RNGs which can still be found all over the place.

The reader who would like to examine these issues in greater detail can look at (Knuth 1997; L'Ecuyer 1994; L'Ecuyer 1998; L'Ecuyer and Hellekalek 1998) and the references given there. A reliable RNG is a basic building block on which all stochastic simulations depend; without it, everything else collapses.

6.2 Quasi-Monte Carlo

Most stochastic simulations are performed to estimate a mathematical expectation, which can always be expressed as an integral over the t-dimensional unit hypercube, say

$$\mu = \int_{[0,1)^t} f(\boldsymbol{u}) d\boldsymbol{u}, \qquad (1)$$

where t can be large (sometimes infinite). To see why, it suffices to recall that the RNGs feeding the simulations produce (an imitation of) a stream of independent $U(0,1)$ random variables, and that the simulation output can be written as a function of these numbers. The usual Monte Carlo (MC) method estimates μ by an average of $f(\boldsymbol{u}_1), \ldots, f(\boldsymbol{u}_n)$, where the $\boldsymbol{u}_i$ are independent random points over $[0,1)^t$. The idea of *quasi-Monte Carlo* (QMC) is to replace the $\boldsymbol{u}_i$ by points that are more evenly distributed over $[0,1)^t$ than random points. In theory, asymptotically as $n \to \infty$, QMC beats MC, in the sense that error bounds (given, e.g., by the Koksma-Hlawka inequality and its generalizations for QMC) converge to 0 at a faster rate than the usual $O(n^{-1/2})$ associated with MC. But in practice, these bounds turn out to be impractical because they are almost impossible to compute and because they get reasonably small only for huge values of n, as soon as t exceeds 10 or so. Nevertheless, QMC seems to work quite well in practice, and sometimes provides huge error reductions compared with MC, even for high-dimensional problems. Research questions on this important topic include:

1. Studying ways of constructing good high-dimensional QMC point sets and sequences for which the important projections over lower-dimensional subspaces are well-behaved, and which are easy to implement and fast. So far, (t, m, s)-nets and lattice rules have been the two leading contenders (Niederreiter 1992).

Polynomial lattice rules are now joining the race (L'Ecuyer and Lemieux 1999).

2. Developing practical ways of estimating the error in QMC, e.g., by transforming QMC into a variance reduction technique via clever randomizations.
3. Understanding how and why QMC works so well for medium/high-dimensional problems.
4. Defining, studying, and comparing figures of merit to measure the uniformity of QMC point sets. These criteria can be based on error or variance expressions.
5. Developing techniques for transforming the models or the estimators (e.g., by reducing the effective dimension) so that QMC works better.
6. Developing adaptive QMC techniques, where the QMC point sets adapt dynamically to the function f of the model considered.

For a glimpse at current research, see, e.g., (Fox 1999; Hellekalek and Larcher 1998; L'Ecuyer and Lemieux 1999; Hickernell at al. 1999; Niederreiter and Spanier 1999).

6.3 Efficiency Improvement

Simulation is often used to compute statistical estimates *on-line*. Reasonable answers are then required very quickly. Examples include option pricing in finance (Boyle, Broadie and Glasserman 1997), short-term production scheduling in a stochastic environment, etc. In these situations, achieving the required precision by simulating the model naively often takes too much time. In other contexts (e.g., in reliability, telecommunications, finance, etc.), important *rare events* are involved, so that excessively long simulations (e.g., several years of CPU time) would be required to obtain reasonable estimators by simulating the model in a straightforward way. Parallel simulation can speed up things to some extent, but greater efficiency improvements can often be achieved via variance reduction techniques. These methods must be adapted or tailored to specific classes of problems. This activity is partly science, partly an art. There is a large number of important classes of problems for which efficiency improvement methods are needed, and for which the effective use of certain general methods (such as importance sampling, splitting, stratification, etc.) is yet to be worked out. For references, see, e.g., (Fishman 1996; Heidelberger 1995; L'Ecuyer 1994).

7 STRATEGIC DIRECTIONS IN DISTRIBUTED SIMULATION (ROGER SMITH)

Here we consider strategic directions and research challenges in distributed simulation. In searching for these technologies the author polled several prominent members of the simulation community and reviewed recent publications that characterized many areas of simulation:

- *Proceedings of the 1998 Winter Simulation Conference* (Medeiros, Watson, Carson, and Manivannan 1998),
- *Proceedings of the Twelfth Workshop on Parallel and Distributed Simulation* (Unger and Ferscha 1998),
- *Proceedings of the 1999 Game Developers Conference* (Yu 1999), and
- *Digital Illusions: Entertaining the Future with High Technology* (Dodsworth 1998).

Distributed simulations are those applications that span multiple computer devices, executables, or geographic areas. These include what are often referred to as parallel and distributed simulations (PADS) and distributed interactive simulations (DIS) (Fujimoto 1999). These communities vary widely in their techniques for implementing a distributed simulation, but they both fall under the general category of distributed simulation.

Distributed simulation is widely applied in military training systems in which computers and executables have been joined together through techniques like the Distributed Interactive Simulation (DIS) protocol, Aggregate Level Simulation Protocol (ALSP), and the High Level Architecture (HLA). It is also used in analytical models in which networked and parallel computers divide a problem into smaller pieces that can be solved more rapidly. The entertainment community has applied distributed simulation ideas in attractions like the Battle Tech Entertainment Center and the Internet-based Virtual Worlds environment. Most computer games also contain a distributed simulation mode that allows them to interoperate with other people playing the game on the Internet. Games like Quake II, Rainbow Six, Command & Conquer, and the entire Star Wars line are well known and well sold for this capability.

7.1 Strategic Directions

The strategic directions are areas in which simulation can be applied immediately, but where we have not taken full advantage of the technology that is available. These include:

- Systems operations and management,
- Real-time decision making,
- Persistent virtual worlds, and
- Virtual verisimilitude.

7.1.1 Systems Operations and Management

It is possible to embed simulation modes in the operating systems of computer systems. These systems can feed data about their operations into a data store that is accessible to simulation processes. Periodic execution of these would evaluate this performance data and identify the operational trends in the data. This can then be used to optimize the system for its most characteristic applications.

The PC is a general-purpose computer that is put to specific tasks once it is in the hands of the user. If the operating system contained a simulation kernel it would be able to evaluate the uses to which each machine was being put and optimize that machine for those applications. The simulation would need a database of application characteristics such as word processing, accounting, databases, graphic art, sound editing, games, web serving, web surfing, telephone management, and hundreds of others.

The advantage of simulation-based adaptation is that the user need not be an expert in configuring the machine and the simulation can re-optimize the machine when it is applied to a different function. Since most systems are used for more than one application, the simulation would also be able to adjust the configuration to best satisfy two or three applications—a task beyond the abilities of most PC users.

7.1.2 Real-Time Decision Making

The world is filled with opportunities to apply computer simulations to assist in real-time decision making. Any place that information is available in a digital form and humans are evaluating that information to making decisions based on that information, there is an opportunity to support the human with a simulation.

These opportunities occur in thousands of fields, only a few of which will be described here.

Combat Consultant. Large military organizations are migrating their communication and decision-making tools to computer systems. This provides combat information in a form that can be accessed and evaluated by a simulation. We are lucky to live in an age in which our citizens are not faced with life-and-death combat decisions on a daily basis. As a result, soldiers that encounter this kind of event are relatively inexperienced at dealing with it. Military organizations mitigate this through extensive training activities (some of which also involve simulations), but there is no substitute for experience. A Combat Consultant is a simulation mode embedded inside of the command, control, communications, computers, and intelligence systems being used by the soldiers (these are commonly referred to as C4I Systems). The simulation is equipped with the expertise of previous commanders and the real-time expertise of other commanders currently using similar systems on the network. The Combat Consultant can monitor the information in the system and suggest alternatives that may be successful under the current situation.

Though this may begin as an expert system, it also includes the real-time experience of other commanders solving similar problems at this moment in time and a simulation engine to project this situation into the future. The system searches for the best strategies for handling each combat situation in real time.

The term C4I evolved from C2 over the last two decades to more accurately describe the operations performed by commanders and their decision support systems. It is time to add *simulation* to the acronym: C4IS.

Aircraft Navigation. The Federal Aviation Administration is planning to change the mechanism for controlling commercial air traffic across the country. Under the new method, entitled "Free Flight," pilots will have unprecedented decision making authority in selecting their flight paths and adjusting them throughout the trip. Simulations can assist these pilots by evaluating environmental data, aircraft status, data received from sensors, and data transmitted from the ground. The simulation can constantly study the current situation, looking for the optimum solution for reaching a destination. Perhaps more importantly, the simulation can also generate customized plans for use in an emergency. When the unexpected happens, a plan is ready and the flight consultant is there to support a pilot who is confused, scared, and unable to make decisions.

Crowd Management. All large cities face the problem of managing the flow of people trying to accomplish their own objectives. These people may be driving in rush hour traffic, searching the mall for a sale, or rushing to the best rides in a theme park. In all of these cases, we could optimize operations by directing this traffic. Using traffic flow sensors we can measure the location and density of people in the system, feed this information to a simulation, and look for solutions in real time.

In the case of the theme park, entertainment events could be scheduled by the simulation in patterns that push and pull the guests in specific directions. Good theme parks are designed to direct the flow of traffic from the time you enter the main gates until you finish your tour of the attractions. These designs would be assisted in real-time by simulations that recognize overcrowding in one area and schedule activities to pull part of the crowd to another area. The "pull" mechanism may be the appearance of a costumed character down a side path; beginning a computerized entertainer directly behind the accumulating mass; or the sounds of a roaring dinosaur in a different direction. These tactics are designed to redirect the crowd in a manner that is non-intrusive and that appears to be of the guest's own volition. Events may also be scheduled to direct the guest's attention away from the fact that they are waiting in line.

Market Prediction. Banks and financial institutions are already using simulation and gaming techniques to analyze past performance and predict future activities. These simulations influence commodity trades, stock speculation, and currency exchanges. They provide an edge over competitors that can result in millions of dollars in additional profits. Simulations of this type can be embedded into many forms of stock selection and advice software, including those used by your stock broker, internet stock trading web page, and personal asset management package (e.g. MS Excel, Quicken, MS Money). These are also useful tools for teaching a novice how markets work and what to watch for in future investments.

7.1.3 Persistent Virtual Worlds

The networked world is a natural host for a persistent virtual world that is accessible to all users. We need to create virtual environments that are persistent over many years and that form the foundation for specific studies, training, and entertainment that will be conducted within them. The gaming community has already accomplished this with online persistent virtual worlds like Ultima Online, Diablo, and Everquest. These provide persistent fantasy worlds that evolve as the users interact with them.

Similar virtual worlds need to be created by high level sponsors of studies and training events. These would be the seeds from which scenarios are drawn and the environment in which distributed interactions occur. Organizations like the Office of the Secretary of Defense, the Defense Modeling and Simulation Office, the Central Intelligence Agency, the National Air and Space Administration, and others need to become the hosts for persistent virtual worlds that support the needs of their entire customer base. It should be possible for globally distributed customers to enter these worlds at any time and explore solutions to current problems.

Commercial versions of this can be used to track the activities of specific individuals in the population. The popularization of cell phones and pagers has placed electronic tracking devices on the belts of a demographic of people that we are most interested in tracking. These tagged people can serve as a sample of the general population, allowing us to see customer density in airports, malls, highways, and large entertainment events.

This could be used to identify potential witnesses to crimes based on their presence in the area and predictions of the path they were likely to have followed while in the area. It may even be possible to identify the perpetrator of the crime using this technique.

7.1.4 Virtual Verisimilitude

In the simulation business we strive to create virtual worlds that accurately represent the real world. This always involves a high degree of abstraction to help us experiment with systems that are far too detailed to fit into any model. However, we have been so conditioned by our lack of computational power and seduced by our skills at abstraction that we sometimes avoid extending our simulations when we have the tools to do so.

There are few simulations that portray a really convincing virtual world. With all of the computational power now available and the increasing maturity of software tools to build models and virtual worlds, we need to explore a new level of representation. It is time for the next big advance in modeling detail and the richness of virtual environments.

Statistically accurate simulations are excellent for many applications, but we need to begin equipping ourselves with models that accurately represent individual objects, events, and interactions without relying on actuarial effects to make them correct.

7.2 Research Challenges

The research challenges are those technologies that are essential for the progress of the field of distributed simulation. Though there are many areas of valuable research, the four listed here are broad enough and essential enough to be listed as research challenges. These are:

- Human behavior modeling,
- Simulation domain architectures,
- Abundant network bandwidth, and
- Practical event management techniques.

7.2.1 Human Behavior Modeling

Many simulations are driven by statistical distributions that characterize the average behavior of a system, but do not claim accuracy for individual events or small time intervals. These distributions represent the activities of machinery, the population growth rates of animals, and human performance of specific tasks. However, they do not model instantaneous behaviors of intelligent or reactive beings in the virtual world. We are in dire need of techniques for inserting intelligent, reactive, unique human behavior in the virtual world.

Military training simulations and computer games require interactions between human operators and automated virtual humans. In the past, this has been accomplished through techniques like Finite State Machines that encode specific behaviors and define the transition conditions from one behavior to another. However, we are discovering the limitations imposed by this technique. These are very difficult systems to create and maintain. Human operators that interact with them regularly identify their limitations and take advantage of them. The entities controlled by these techniques do not exhibit realistic behaviors, rather they exhibit correct behavior—“by-the-book,” robotic actions. We need to discover and create techniques for representing the behavior of human leaders, followers, and groups that give them the ability to appear "live" or "real" to the humans interacting with them. Both the military and the gaming communities are augmenting their robotic methods with "softer" models that include human emotion, training, and fatigue. These result in objects that are all slightly different in spite of being driven by the same software.

The distributed simulation community needs a set of behavior libraries that can be linked into a simulation in the same way we currently link in statistical distributions. This will require the definition of a set of categories of behavior and API's that are necessary to stimulate those categories.

7.2.2 Domain Architectures

Within the U.S. Department of Defense we have been developing standard protocols for joining multiple, previously independent, simulations. These methods have included the DIS protocol, the Generic Data System (GDS), ALSP, and most recently, the HLA. With HLA we have begun to identify simulation functionality that is generally necessary for all systems and which should be included in an infrastructure rather than within specific simulation models. This approach allows a simulation development team to reuse some of the essential functionality that is included in the general infrastructure. It has also encouraged us to question the uniqueness of every simulation system. We recognize that simulations fall into domain areas in which the degree of commonality is much higher than it is across all simulation systems. We begin to imagine a layered view of simulation uniqueness similar to network protocol layers. Higher layers become more specific until they narrow to a single application.

It should be possible to develop an architecture that supports an entire domain of simulation systems, providing a large common pool of functionality. These architectures may include a general interoperability standard like the HLA, but would go further by defining a set of domain tools for operating the simulations, common interfaces to connect to external systems, and object base-classes from which to extend unique object instances.

7.2.3 Information Bandwidth

Distributed simulations *cannot* exist without sufficient reliable communications bandwidth for delivering events and synchronizing execution of the entire system. This bandwidth is currently one of the limiting factors on the size of a distributed simulation. Luckily, bandwidth is also a limiting factor for all applications using the Internet. This has attracted millions of commercial research and development dollars to the problem. That work can and will be applied directly to simulation applications. The global communications industry will discover methods for providing abundant information transfer. These will include methods for configuring the physical medium of delivery and efficient protocols for transferring data. We may productively put our efforts into simulation-specific communications protocols that are not addressed by other communities.

7.2.4 PDES Management

For twenty years we have been involved in research to discover techniques for practical and efficient synchronization of distributed simulation processes. This has resulted in some very clever and powerful ideas for addressing this problem. However, these ideas have been embraced by few industrial and government applications. The constructive wargaming community has adopted Conservative Time Management, but Optimistic Time Management is still searching for an ideal application.

We must identify applications that are well served by the different methods of event management. To justify further study, our research in this area needs to find a practical and valuable home in commercial, government, or military simulation systems. By 2010 we should be able to apply the appropriate synchronization technique to a distributed simulation by analyzing the problem, setting configuration variables, and attaching the event management engine to our simulation. Trial-and-error and fine-tuning of the engine must become standardized such that a simulation professional can perform these operations, rather than a PDES specialist.

7.3 Conclusion

The strategic directions and research challenges presented in this section emphasize two different aspects of the future of distributed simulation. The first is the need for additional development and imagination in applying the technologies we already have. The second is the need for additional research and innovation in areas that will allow us to advance the state-of-the-art. It is the author's opinion that, while the research challenges provide stimulating problems, the strategic directions for applications are much more urgent at this time. The world is in the middle of an information, communication, and computational explosion. Thousands of advanced applications are being fielded every year and many of these could be improved through the inclusion of existing simulation technologies. However, these opportunities are being lost or the technology reinvented by others because of the lack of communication, marketing, and proselytization by members of the "core" simulation community.

8 SUMMARY

This panel represents initial steps in the formulation of a strategic vision for simulation research. We believe that the formulation of such a vision could provide valuable guidance and assistance with respect to decisions involving the generation and allocation of future research funding.

In this paper, we have addressed problems involving: (1) the size and complexity of models; (2) verification, validation and accreditation; (3) the modeling methodological and model execution implications of parallel and distributed simulation; (4) the centrality of modeling to the discipline of computer science; and (5) random number generation and execution efficiency improvements through quasi-Monte Carlo, and variance reduction. Obviously, the practical limitations a single panel imply that many important topics have not been addressed. We hope that this treatment is the beginning of a dialogue—one that will serve to stimulate an assessment of the strategic research needs spanning the breadth of simulation as a discipline.

REFERENCES

Boyle, P., Broadie, M. and Glasserman, P. 1997. Monte Carlo methods for security pricing, *Journal of Economic Dynamics and Control*, **21**:1267-1321.

Digital Arts and Sciences Programs, http://www.cise.ufl.edu/fdwi

Dodsworth, C. 1998. *Digital Illusion: Entertaining the Future with High Technology*, ACM Press. New York, NY.

Ferren, Bran. 1999. Some Brief Observations on the Future of Army Simulation, Army RD&A Magazine. May-June 1999.

Fishman, G. S. 1996. *Monte Carlo: Concepts, Algorithms, and Applications*, Springer Series in Operations Research, New York: Springer-Verlag.

Fishwick, P. 1999. A Modeling Strategy for the NASA Intelligent Synthesis Environment, to appear in: *Journal of Space Mission Architecture*, June.

Fishwick, P. 1995. *Simulation Model Design and Execution: Building Digital Worlds*, Prentice Hall.

Fox, B. L. 1999. *Strategies for Quasi-Monte Carlo*, Boston, MA: Kluwer Academic.

Fujimoto, R. M. 1993. Parallel Discrete Event Simulation: Will the Field Survive? *ORSA Journal on Computing*, **5**(3): 213-230.

Fujimoto, R. M. 1999. *Parallel and Distributed Simulation Systems*, Wiley Interscience.

Heidelberger, P. 1995. Fast simulation of rare events in queueing and reliability models, *ACM Transactions on Modeling and Computer Simulation*, **5**(1):43–85.

Hellekalek, P. and Larcher, G., Eds. 1998. *Random and Quasi-Random Point Sets*, Volume 138 of *Lecture Notes in Statistics*. New York: Springer.

Hickernell, F. J., Hong, H. S., L'Ecuyer, P. and Lemieux, C. 1999. Extensible lattice sequences for quasi-Monte Carlo quadrature, submitted.

Knuth, D.E. 1997. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Third ed. Reading, Mass.: Addison-Wesley.

L'Ecuyer, P. 1994a. Efficiency improvement via variance reduction, In: *Proceedings of the 1994 Winter Simulation Conference*, 122-132.

L'Ecuyer, P. 1998. Uniform random number generators, In: *Proceedings of the 1998 Winter Simulation Conference*, 97-104.

L'Ecuyer, P., and P. Hellekalek. 1998. Random number generators: Selection criteria and testing, In: *Random and Quasi-Random Point Sets*, P. Hellekalek and G. Larcher, Eds. Volume 138 of *Lecture Notes in Statistics*, 223-265, New York: Springer.

L'Ecuyer, P., and Lemieux, C. 1999. Quasi-Monte Carlo via linear shift-register sequences, To appear in: *Proceedings of the 1999 Winter Simulation Conference*.

Medeiros, D.J., Watson, E.F., Carson, J.S., and Manivannan, M.S. Eds. 1998. *Proceedings of the 1998 Winter Simulation Conference*, Washington D.C.

Niederreiter, H. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*, Volume 63 of *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*. Philadelphia: SIAM.

Niederreiter, H. and Spanier, J. Eds. 1999. *Monte Carlo and Quasi-Monte Carlo Methods 1998*, Lecture Notes in Computational Science and Engineering, New York: Springer-Verlag, to appear.

Steinman, J. S. 1992. SPEEDES: A Multiple- Synchronization Environment for Parallel Discrete Event Simulation. *International Journal on Computer Simulation*, 251-286.

Unger, B. and Ferscha, A. Eds. 1998. *Proceedings of the Twelfth Workshop on Parallel and Distributed Simulation*, Banff, Alberta.

Wieland, F. 1997. Limits to Growth: Results from the Detailed Policy Assessment Tool. In *Proceedings of the the 16th Annual IEEE Digital Avionics Systems Conference*, Irvine, CA.

Yu, A. Ed. 1999. *Proceedings of the 1999 Game Developers Conference*, San Jose, California.

AUTHOR BIOGRAPHIES

ERNEST H. PAGE is a Lead Scientist in modeling and simulation with The MITRE Corporation. He received the Ph.D. in Computer Science from Virginia Tech in 1994. He is Chairman of the Association for Computing Machinery

(ACM) Special Interest Group on Simulation (SIGSIM), an Associate Editor for the *ACM Transactions on Modeling and Computer Simulation*, and co-coordinator for the Military Applications track within the 1999 Winter Simulation Conference.

DAVID M. NICOL is Professor of Computer Science at Dartmouth College. He has served on the Editorial Board of ACM TOMACS from its inception, and is current the Editor-in-Chief. He has published extensively on topics in parallel processing and performance analysis, particularly in the area of parallel simulation. He received a B.A. in mathematics from Carleton College in 1979, and a Ph.D. in computer science from the University of Virginia in 1985.

OSMAN BALCI is Professor of Computer Science at Virginia Tech and President of Orca Computer, Inc. He received B.S. and M.S. degrees from Bogazici University in 1975 and 1977, and M.S. and Ph.D. degrees from Syracuse University in 1978 and 1981. Dr. Balci is the Editor-in-Chief of two international journals: *Annals of Software Engineering* and *World Wide Web*. He also serves as the Verification, Validation and Accreditation (VV&A) Area Editor of *ACM Transactions on Modeling and Computer Simulation* (TOMACS) and Simulation and Modeling Category Editor of *ACM Computing Reviews.* He is a Director-at-Large for the Society for Computer Simulation International (SCS) and is a member of the Winter Simulation Conference Board of Directors representing SCS. Dr. Balci is a member of Alpha Pi Mu, Sigma Xi, Upsilon Pi Epsilon, ACM, IEEE CS, INFORMS, and SCS.

RICHARD M. FUJIMOTO is a Professor in the College of Computing at the Georgia Institute of Technology. He received the Ph.D. and M.S. degrees from the University of California (Berkeley) in 1980 and 1983 (Computer Science and Electrical Engineering) and B.S. degrees from the University of Illinois (Urbana) in 1977 and 1978 (Computer Science and Computer Engineering). He has been an active researcher in the parallel and distributed simulation community since 1985 and has published over 100 conference and journal papers on this subject. He has given several tutorials on parallel and distributed simulation at leading conferences. He has co-authored a book on parallel processing and recently completed a second on parallel and distributed simulation. He served as the technical lead in defining the time management services for the DoD High Level Architecture (HLA). Fujimoto is an Area Editor for *ACM Transactions on Modeling and Computer Simulation.* He also served as chair of the steering committee for the Workshop on Parallel and Distributed Simulation, (PADS) from 1990 to 1998 as well as the conference committee for the Simulation Interoperability Workshop (1996-97).

PAUL A. FISHWICK is Professor of Computer and Information Science and Engineering at the University of Florida. He received the Ph.D. in Computer and Information Science from the University of Pennsylvania in 1986. His research interests are in computer simulation, modeling, and animation, and he is a Fellow of the Society for Computer Simulation (SCS). Dr. Fishwick will serve as General Chair for WSC00 in Orlando, Florida. He has authored one textbook, co-edited three books and published over 100 technical papers.

PIERRE L'ECUYER is a professor within the Département d'Informatique et de Recherche Opérationnelle, at the University of Montréal. He received a Ph.D. in operations research in 1983, from the University of Montréal. He obtained the *E. W. R. Steacie* grant from the Natural Sciences and Engineering Research Council of Canada for the period 1995–97. His main research interests are random number generation, efficiency improvement via variance reduction, sensitivity analysis and optimization of discrete-event stochastic systems, and discrete-event simulation in general. He is an Area Editor for the *ACM Transactions on Modeling and Computer Simulation*. More details at: `http://www.iro.umontreal.ca/~lecuyer`, where his recent research articles are available on-line.

ROGER SMITH is the Technical Director for BTG Inc., Chief Scientist for ModelBenders Inc., and an Adjunct Professor at the Florida Institute of Technology. He is actively involved in designing, developing, and fielding constructive and virtual simulations for the Department of Defense. He is the Area Editor for Distributed Simulation for *ACM Transactions on Modeling and Computer Simulation* and has just completed his term as Chair of the ACM Special Interest Group on Simulation.