

1-1-2002

Perspectives on the Evolution of Simulation

Richard E. Nance

Virginia Tech, Department of Computer Science and Systems Research Center, nance@vt.edu

Robert G. Sargent

Syracuse University, Department of Electrical Engineering and Computer Science, rsargent@syr.edu

Follow this and additional works at: <http://surface.syr.edu/eecs>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Nance, Richard E. and Sargent, Robert G., "Perspectives on the Evolution of Simulation" (2002). *Electrical Engineering and Computer Science*. Paper 100.

<http://surface.syr.edu/eecs/100>

This Article is brought to you for free and open access by the L.C. Smith College of Engineering and Computer Science at SURFACE. It has been accepted for inclusion in Electrical Engineering and Computer Science by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

PERSPECTIVES ON THE EVOLUTION OF SIMULATION

RICHARD E. NANCE

Department of Computer Science and Systems Research Center, Virginia Tech, Blacksburg, Virginia 24060, nance@vt.edu

ROBERT G. SARGENT

Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, New York 13244, rsargent@syr.edu

Simulation is introduced in terms of its different forms and uses, but the focus on discrete event modeling for systems analysis is dominant as it has been during the evolution of the technique within operations research and the management sciences. This evolutionary trace of over almost fifty years notes the importance of bidirectional influences with computer science, probability and statistics, and mathematics. No area within the scope of operations research and the management sciences has been affected more by advances in computing technology than simulation. This assertion is affirmed in the review of progress in those technical areas that collectively define the art and science of simulation. A holistic description of the field must include the roles of professional societies, conferences and symposia, and publications. The closing citation of a scientific value judgment from over 30 years in the past hopefully provides a stimulus for contemplating what lies ahead in the next 50 years.

1. INTRODUCTION

1.1. Background

In this paper we present joint perspectives on the development and evolution of simulation over approximately 50 years, recalling a few personal experiences during this period. We preface these remarks with a brief sketch of our introduction to simulation as graduate students during the early and mid-1960s. During the intervening period, both of us have been active in research and the teaching of simulation, heavily involved in service to the simulation community, and engaged with practitioners through consulting and sabbatical appointments.

RGS wrote his first simulation program in 1961 using the Michigan Algorithmic Decoder (MAD) language while doing his graduate work at The University of Michigan. His principal appointment at Syracuse University was in industrial engineering and operations research, with additional appointments in computer and information science and in electrical and computer engineering. His research, sponsored primarily by the U.S. Air Force over some 25 years, focused on modeling and simulation.

REN wrote his first simulation program in FORTRAN in 1965. He used simulation to study the machine interference problem for his M.S. research at North Carolina State University, which led to his interest in time flow mechanisms and the influence of model representation on execution efficiency. His Ph.D. dissertation at Purdue University is believed to be the first attempt to apply system dynamics to not-for-profit service organizations (university libraries). His permanent academic appointments have been in computer science/operations research and in computer science,

at Southern Methodist University from 1968 to 1973 and since then at Virginia Tech. He directs a research center established in 1983 by the U.S. Navy and has personally conducted projects in modeling and simulation, software engineering, and computer networking.

1.2. Categorizing Simulation

An understanding of the evolution of simulation is assisted by applying categorizations according to various criteria. One such categorization is based on the objectives of the simulation study. By far, the early work in simulation and that which has been dominant in management science and operations research over the history is *system analysis*, where the intent is to mimic behavior to understand or improve system performance. A second objective is *education and training*, where the former addresses the broader understanding of concepts and the latter, more specific behavior in the application of concepts. A third objective is *acquisition and system acceptance*, where the simulation model is intended to answer questions related to “Does the system meet the requirement?” or, “Does a subsystem contribute significantly to the improvement of the larger system performance?” A fourth objective relates to *research* which can involve the creation of an artificial environment. In such an environment, systems components can be tested or the behaviors of an individual or groups can be compared, contrasted, or categorized. *Entertainment* is the most recent objective: using a simulation model in a real-time interactive mode to derive pleasure and enjoyment.

A second categorization relates to the representation of time and state in a simulation model. A Monte Carlo model requires state sequencing but no explicit representation

Subject classifications: Professional: comments on. Simulation: discrete event simulation.
Area of review: ANNIVERSARY ISSUE (SPECIAL).

of time. Discrete event models specify state changes at discrete points in time. Continuous simulation portrays state changes as continuous over time, and discretized approximate solutions of differential equations are the most common examples. Combined discrete event and continuous models enable both techniques to be applied within the same study. Hybrid simulation models generally incorporate an analytical submodel within a discrete event model (Shanthikumar and Sargent 1983).

Related to simulation models are games and gaming, a topic of considerable interest in the early history. Stimulated by the entertainment objective, games are experiencing a strong resurgence, but the earlier batch mode of play is now replaced by real-time interaction with human players. (Games pitting computer programs against each other have also been staged.)

The use of simulation precedes computers, either analog or digital. Described by some authors as “artificial sampling,” a manual Monte Carlo method was employed by Buffon to estimate π in a study documented in 1777 (Jansson 1966). Hammersley and Handscomb (1964, p. 7) identify “Student” (a pseudonym) as using artificial sampling to calculate exact expressions for the distribution of the sample correlation coefficient and to derive what is now called Student’s t-statistic.

Computer simulation began during World War II in the case of the continuous and Monte Carlo models. Discrete event simulation probably originated in the late 1940s; however, we have no evidence as to the exact date. During the remainder of this paper we focus on computer simulation with discrete event models and simply use the term “computer simulation” or “simulation.”

2. THE EARLY DAYS

Simulation books published in the 1960s present a rather uniform set of steps for conducting a study: problem formulation, system data collection and conceptual model formulation, validation of the conceptual model, construction of the simulation program, execution of the simulation program, operational (results) validation, experimental design, output data analysis, and documentation. To give a feel of how the earliest steps were accomplished, we examine the form of input, the content of the input, the execution of the program (running the model), and the output.

The early simulation program consisted of a model description and an auxiliary set of simulation functions, including a random number generator, random variate generators, list processing routines for queue insertion and deletion, a time flow mechanism, forms of model data collection and analysis, and a report generator. A Simulation Programming Language (SPL) representation of the model would include a library that the translator would access to provide these functions. If programmed in a General Purpose Language (GPL), the simulation modeler would have to rely on a GPL library routine or include a program to perform each necessary function.

The content of the model, referred to as the model specification, requires a world view (*Weltansicht*), as Lackner (1962) notes. (A referee notes that *Weltanschauung* is the correct term.) The early works of Lackner (1962) and Krasnow (1967) identify differences in world views that distinguish SPLs. Kiviat (1969) explores the similarities and differences in the first work to explain fully the SPL implementations of the world views.

The implementation step (programming) of the model relied on simple techniques for creating uncertainty (random number and random variate production), about which little in terms of randomness properties was actually known. Arrays were used for simple list processing, and variable- or fixed-time incrementing was used in the time flow mechanism. Data collection methods used simple statistical accumulations, again relying on the array data structure. A deck of 80-column (also called “IBM”) cards represented both program and data, and RGS recalls that at Michigan a typical turnaround time for computer jobs submitted by graduate students was one to two days. Debugging of programs was often tedious and not infrequently required the decoding of “core dumps” with values in octal or hexadecimal representation. Storage limitations and costly processor time required conservation on both fronts.

Output analysis often took the form of multiple replications with different random number streams or simply reliance on an estimation of the mean without concern for variance estimation. Report generation was limited to a rather restricted set of output variables; however, early versions of the SPLs did supply some rudimentary forms of dynamic error checking.

The teaching of simulation in the 1960s was inhibited by the lack of textbooks. Often based on experience from practice, instructors used techniques that had no identifying source, and students were expected to know (or to learn) basic fundamentals of computer programming, list processing, and statistics as needed. RGS notes that simulation was introduced as part of his course on data processing, required of both undergraduates and M.S. students in industrial engineering at The University of Michigan. A graduate course on simulation was also available, and typical techniques included model specification by flow charts, an event scheduling world view representation, and implementation in the MAD language.

With instructional responsibility for simulation at Syracuse University in 1967, RGS used Naylor et al. (1966) as the text and taught both the event scheduling world view (using FORTRAN) and the transaction world view (using GPSS). In teaching his first course in 1968, REN exercised the same choices in languages and text. Programming simulations in the early period, irrespective of language, required the use of a manual provided by the language vendor.

Excitement and expectation characterized both the academic and industrial sectors of the simulation community in the 1960s. SPL developers were interested in sharing ideas and understanding different approaches, and this interest

is reflected in a number of conferences and papers comparing simulation languages. The first book on simulation appeared (Tocher 1963), and a number of others followed thereafter. REN and RGS believe that a healthy tension existed between research and practice, and the methods and techniques for modeling and simulation created during that period have had a lasting influence.

3. TECHNICAL FACTORS IN SIMULATION DEVELOPMENT

Examining the first few pages of a contemporary simulation book in comparison with one published 30 years ago, one notices strong similarities in the steps described for performing a simulation study. Essentially, *what* characterizes the modeling and simulation activities today seems little different; however, *how* those steps are performed differs considerably. We have chosen to separate factors marking the evolution of simulation into two categories: (1) *external*—those emanating from computing technology that set directions or shaped the progress of simulation research and practice; and (2) *internal*—those generated by the communities of simulation researchers and practitioners. Within each factor below, the development is described in a loose chronological order. We readily admit that selection of these factors is subject to personal biases; others might identify influences felt to be more significant and could present strong arguments for their choices.

3.1. External Factors Shaping the Evolution

The early pervading view of simulation as a problem-solving technique stems from the development of Monte Carlo techniques well before the appearance of either analog or digital computers. Monte Carlo computations performed on electromechanical calculators by a host of operators was the common solution procedure in the 1950s for numerical models (approximate solutions of differential equations). Yet simulation, and in particular discrete event simulation, could never have been a major problem-solving technique without the emergence and rapid development of the digital computer. Consequently, our view is that the external influences on simulation are dominated by those associated with digital computing technology.

3.1.1. The Revolution in Computer Hardware. Youthful faces and sprightly gaits aside, we are made even more conscious of the huge gaps in computing history separating us from our students when we realize that most today do not recognize the terms: “mainframe computer,” “core memory,” or “keypunch.” Since the ENIAC in the late 1940s, progress in computer hardware has advanced at a revolutionary pace. Processor speeds and storage sizes (both primary and peripheral) have increased by several orders of magnitude, while component size has decreased to a like degree as the succession of hardware technologies has transitioned from the mainframes of the 1960s to the minicomputers of the 1970s, the parallel processors of the

1980s, networks of processors in the 1990s, and the desktops and laptops joined by wireless connections of today. While the time intervals stipulated above are imprecise, the impact on modeling and simulation has been pervasive.

For the vast majority of the OR community, recognition of the hardware influence needs little justification, but we believe that the effects are more pronounced for simulation than for most areas. That claim aside, the incredible advances in computer hardware must be acknowledged as making simulation a viable problem-solving technique for some and the preferred technique for many.

3.1.2. Advances in Computer Software. The machine language representations of the early 1950s gave way to the assembly language of the mid-1950s. The improved representational capability, supporting the list processing and functional library organization needs for simulation, made a significant contribution to development of the first packaged simulator: the General Simulation Program (GSP) of Tocher and Owen (1960). By the late 1950s FORTRAN had extended the semantics helpful for understanding model representations. While FORTRAN emerged as the dominant language for engineers and scientists in the United States throughout the next two decades, its limited data structures (the array) in the early versions had some lasting effects. FORTRAN simulation packages such as GASP and MILITRAN provided functional capabilities, but in general inhibited the acceptance and widespread use of SPLs that were appearing in the mid-1960s.

In Europe ALGOL was the dominant language, and its failure to achieve widespread acceptance in the United States was in part due to a hardware influence: the domination of IBM in the mainframe market at the time. SIMULA 67, as an extension of ALGOL, ushered in the object-oriented programming style. Popularized by Smalltalk in the 1980s, object-oriented programming would become the dominant software methodology in the 1990s.

A burgeoning interest in SPLs stimulated a number of representational issues related to specification and abstraction in model development. Graphical representations were prominent in some early languages, notably the flow chart symbology in GPSS and the Activity-Cycle Diagram (ACD) (or wheel chart) in the Control and Simulation Language (CSL), popular in the United Kingdom. The history of simulation programming language development is described briefly in Crain et al. (1992) and by Hixson in Araten et al. (1992). A more detailed description of the history of SPLs from 1955 to 1985 is given in Nance (1996).

The recognition of software engineering as an area of study had its own effects on simulation in the late 1960s and early 1970s. Model documentation, stimulated by issues in program documentation, became an important concern as did the life-cycle perspective and the user involvement in model development. Two government reports (U.S. General Accounting Office 1973, 1976), identified major deficiencies in “computerized models” (many

were simulation studies). Sensitivity to good software engineering practices became a requirement for major simulation modeling efforts. Arguments were advanced that model representation should generate model documentation, and that the common consideration of documentation as an after-the-fact “activity” was a major detriment to effective use of the model (Nance 1979).

3.1.3. Influences by Other Computing Technologies.

The influences cited in this section derive from technical areas that do not fit within either hardware or software. A brief description serves to support the assertion that these technical areas have had notable influence on modeling and simulation.

3.1.3.1. Computer Graphics. Utilizing the capabilities to discriminate based on color, perspective, and motion, advances in computer graphics have led to the use of animation for model output and increased the credibility of simulation results. Interestingly, the early emphasis on graphical input (the flowchart symbols in GPSS and the Activity-Cycle Diagrams) did not persist with the major advances in the 1980s. An early development that never reached commercial use was the RAND Tablet that transformed the drawing of GPSS flow-chart symbols automatically into a GPSS program. GPSS/NORDEN (Reitman et al. 1970) demonstrated the use of output animation using vector graphics. The NORDEN version also provided a graphical depiction of transaction queueing in the GPSS flow diagram to assist in program debugging, which, coupled with the capability for user interrupts, permitted some interactive corrections and changes.

Graphical interests in the late 1970s and early 1980s centered on output animation. Color and motion were prominent in depicting product transformations during execution of the simulation model. Visual Interactive Simulation (VIS) became a prominent technology in the mid-1980s, and the claim was made by O’Keefe (1986) that the United Kingdom was ahead of the United States at that time in this technology. Bell and O’Keefe (1994) describe VIS in methodological terms, contrasting the active and passive forms in model development and experimentation. Current work in graphics is pushing the development of three-dimensional displays of output behavior.

3.1.3.2. Human-Computer Interaction. Significant developments in human-computer interaction (HCI) would not have been possible without hardware and software advances that enabled time-sharing operating systems and interactive programming. HCI has the goal of making interactive software efficient, effective, safe, and satisfying in its use (Hartson 1998). A major consequence of the conjunction of HCI with other advances is an ever-increasing user relief from the requirement to have detailed knowledge of the underlying computing technology. The result has greatly expanded the population of productive users of the ubiquitous digital technology. However, a concomitant result is that, unless the user forces revealing actions, the modeling software hides how the function is performed.

An unsettling consequence is that simulation model *users* need not be those who *developed* the model, and users are likely to have little understanding of how the model results are being produced. Furthermore, model *developers* sometimes lack a sufficient understanding of the internal logic of SPLs to enable the recognition of erroneous results produced by incorrect models.

3.1.3.3. Computer Networks. The dumb terminal interface enabled by time-sharing systems of the 1970s, and first used for simulation purposes by the OPS project at MIT in the 1960s, was displaced by the networking of terminal interfaces in the 1980s. Several terminals connected to a minicomputer, supplemented by networked communications among several other minicomputers, was a typical architecture. As the minicomputers gave way to the microprocessor workstations, the growth of networks accelerated. The National Bureau of Standards (now National Institute of Standards and Technology) was a major factor through its leadership in the international arena that led to the creation of local area and metropolitan area networking standards. The preponderance of networked computing, coupled with the emergence of the Internet, provided the enabling factors for distributed interactive simulation.

3.1.3.4. The World Wide Web. In this past decade, network computing has expanded to a global level. Web-based simulation is now an implementation issue rather than a research concept. The potential in web-based simulation is for a model to be constructed and provided as a commodity. Users can define a set of parameter values and select alternate structures internal to the model in configuring an experiment. Remote execution is invoked to produce the simulation output. The maintenance and modifications are left with the model producer, and the simulation activity assumes the role of receiving service from a utility. The responsibilities of model producer and simulation experimenter are clearly distinguished.

3.2. Internal Factors

Selecting the key factors in over 40 years of research and experience in simulation is a challenge. We offer our suggestions in the brief description that follows.

3.2.1. Modeling. As the SPLs of the early 1960s emerged, each offered a conceptual framework derived from an application area, the influence of a GPL, or some combination of influences. Amidst the din of claims from the language disciples, a few such as H. S. Krasnow (1967) sought to fathom the SPL differences in more fundamental language-independent terms. He categorized world views for continuous, discrete, and combined (discrete and continuous) simulation and described ways of representing systems for discrete simulation in a paper presented at the 1965 NATO Conference on Digital Simulation in Operational Research. The representations for discrete event simulation included the event, activity, and process representations. A detailed comparative analysis of the differences

in these three, and the transactional, world views are contained in Kiviat (1969).

Graphical assistance in model specification accompanied the introduction of the languages for the transaction (GPSS) and activity (CSL) world views. For the latter, the activity-cycle diagram served several SPLs and was the basis for the interactive program generation work in the United Kingdom in the 1970s. Event Graphs were introduced by Schruben (1983) to assist in model building using the event world view and further developed by Sargent (1988) and Som and Sargent (1989). (We note that a representation similar to event graphs was introduced in Evans et al. 1967, but apparently not further developed.)

A graphical model representation for the process world view, called Control Flow Graphs, was developed by Cota and Sargent (1990) (see also Cota et al. 1994) and later extended to Hierarchical Control Flow Graphs by Fritz and Sargent (1995) to aid in the control of representational complexity.

Model development environments that were research subjects in the 1980s, see Nance (1983), have become the practitioner's initial modeling tool in the 1990s. In 1990, the renaming of the tutorial track "Software" to "Software and Modelware" in the program of the Winter Simulation Conference reflected the expansion of modeling tools beyond the SPL level. Today, visual interactive modeling employs icons, graphical depictions, or actual pictures of system elements imported to provide a more recognizable association with the system counterpart. *Modeling methodology*, which includes events list management, automated and semi-automated modeling techniques (diagnosis, agent-based approaches), time flow mechanisms, and validation and verification, is recognized as a primary research area. The transition to environments has relieved most practitioners from direct involvement with modeling methodology issues, relegating them to provided functions. This indirect involvement does exact a price: the inability of model users to recognize potential errors (see §3.1.3.2 above). Moreover, these simulation functions remain as essential contributors to the success of a simulation study.

3.2.2. Simulation Functions.

3.2.2.1. Random Number and Random Variate Generators. Random number generation (RNG), tests for randomness, and transformation techniques (random variate generators) have been active research topics since the advent of the digital computer. The congruential (or Lehmer) generators (Lehmer 1951), whose behavior could be based on number-theoretic properties, displaced the ad hoc techniques in the 1960s, but various unsubstantiated methods for achieving better randomness properties can be found in publications into the 1980s. Knuth (1997) provides a comprehensive description of RNG techniques and tests for randomness, and descriptions of current research findings are provided by L'Ecuyer (1998, 2001). The field of random variate generation gained maturity in the 1980s after much research in the prior decades. An extensive treatment of transformation techniques is given in Devroye (1986).

3.2.2.2. Time Flow Mechanisms and Event List Management. Early simulation programs used either a fixed-time increment (FTI) or variable-time increment (VTI) method as the basis for control of time. The developer of an SPL, influenced by application area or perception of run-time efficiency often chose between FTI and VTI inevitably creating the world view implemented in the language. Each world view promotes a particular way of characterizing the relationships among model objects and their attributes depicting time and state (Nance 1981).

The event view imposes the implementation of time passing because events occur, and event list management (insertion, reordering, and removal) determines the execution time for a simulation model. A contentious issue in the 1960s was the comparative performance of FTI and VTI methods. Nance (1971) showed that universal superiority could not be claimed by either method.

During the late 1970s and early 1980s research in data structures for event list management received major attention. Personally involved in this work, RGS recalls the excitement on discovery that efficient list processing algorithms could reduce the run times of some simulations up to 30 or even 40% (McCormack and Sargent 1981).

The three-phase extension to the activity scan method, attributed to Tocher, dominated in the United Kingdom and the SIMULA co-routine implementation of process interaction had major influence in Europe and a few United States locations. Both of these methods can transition between resembling a next event or an activity scan method. The process view treats the object as primary, but the (process) transactional view characterizes only the dynamic objects as processes. (Hence, the term "active resource" is also used to distinguish the "pure" process view from the transaction view.)

3.2.3. Verification and Validation. The importance of basing a decision on results from a valid model is underscored in an early text, Naylor et al. (1966), that quotes an earlier paper containing a definition of simulation proposed by C. West Churchman (1963):

"X simulates Y" is true if, and only if, (a) X and Y are formal systems, (b) Y is taken to be the real system, (c) X is taken to be an approximation to the real system and (d) the rules of validity in X are non-error-free.

Thomas Naylor, a coauthor of the book cited above, deserves credit for drawing major attention to the validation issue in the 1960s: Is the model actually representing the *truthful* behavior of the referent system? His work, above and in later publications (Naylor 1971, Naylor and Finger 1967), exerted a major influence in framing validation within different philosophical perspectives. Numerous techniques that can be used were identified or developed. While the issues of both verification and validation were of concern from the early days of simulation, often no clear distinction was made between the two terms.

In the late 1970s Sargent (1979, 1981) and Balci and Sargent (1980, 1981) raised the visibility and understanding of verification in contrast with validation and placed the

latter within a sound statistical framework. With the strong interest in verification from the software engineering community, this contrasting but complementary explanation of the term was quite important. The effort to place validation in a cost-risk framework moved the concept from a philosophical explanation in earlier works to a form more useable for simulation practitioners.

Current views hold verification and validation to be separate processes, each employing techniques appropriate to the differing objectives. Informally, verification focuses on the activities in developing the model (“producing the model correctly”) and validation focuses on comparison of the model with the referent system (“producing the correct model”). Formal statistical tests developed for model validation are difficult to apply in practice because of the required assumptions and/or availability of system data. In practice, both verification and validation are often performed using subjective (inspection) approaches, with the validation being given the greater attention.

3.2.4. Analysis Methodology. Richard Conway (1963) initiated a research area that became characterized as *analysis methodology*. This paper was the first to take a holistic approach to simulation experimentation, identifying the two phases as “strategic planning” and “tactical planning.” While this paper concentrates on tactical planning, in particular the difficulties inherent in steady-state parameter estimation, it also discusses the use of variance-reduction techniques and different statistical approaches for comparisons of alternatives systems (or operating policies) using simulation. The analysis methodology area has been and continues to be an extremely active research area with papers on the subject numbering in the hundreds. (See Law and Kelton 2000 for a detailed discussion of current analysis techniques.)

3.2.4.1. Output Analysis. The analysis of simulation output is divided into two system classes: steady-state and terminating. Systems such as banks and many retail outlets can be modeled as terminating simulations if replications of a defined operating period can be assumed to constitute an independent and identically distributed random sample. The classical statistical analysis techniques can then be employed.

Systems modeled for steady-state analysis introduce the complexities of: (1) removal of the bias of the imposed initial model state and (2) definition of a sample that admits an accepted estimate of sample variance, which is needed to determine the precision of estimates of steady-state parameters. Removal of initial state bias (also called the initial transient problem), despite some innovative approaches, remains an unresolved problem (unless the regenerative process technique is employed). Estimation of sample variance by imposing assumptions to apply the method of replications or the batch means method (dividing one long series of output values into batches with the autocorrelation among them included in the variance estimate) are both addressed by Conway (1963). Theoretical and experimental research since that time has significantly improved the

understanding of the behavioral properties of both methods. Developing procedures to determine appropriate batch sizes remains an active research topic (see papers in recent issues of the *Proceedings of the Winter Simulation Conference*).

Research in other approaches for variance estimation include spectral analysis, autoregressive models, regenerative processes, overlapping batch means, standardized time series, and combinations of different methods. Alexopoulos and Seila (2000) provide an instructive description. Comparative behaviors have been investigated using theoretical and experimental approaches (for example, Sargent et al. 1992 employs both), and each method has proponents. Generalized Semi-Markov Processes (GSMPs) have been proposed as a foundation for steady-state output analysis (Glynn 1989, Haas 1999). Variance estimation for steady-state analysis remains an actively investigated problem.

While the discussion above dwells on the estimation of the mean and associated confidence interval for a single simulation model parameter, active research continues on other output analysis techniques. Included in this group are quantile estimation, multiple joint measures, the use of fixed-sample-size versus sequential-sample-size procedures, Bayesian statistics (Chick 2000), jackknife and bootstrap sampling (Efron 1982, Efron and Tibshirani 1993). Variance-reduction techniques, described below, have also received much attention.

3.2.4.2. Experimental Design and Comparison of Alternatives. From the early 1960s until today, considerable research has dealt with the use of classical design of experiments in simulation for such applications as comparison of alternatives, metamodeling, optimization, sensitivity analysis, and validation (see Kleijnen 1975). Conway, in his 1963 paper, suggests that the newly proposed (at that time) ranking and selection (R&S) procedures were more appropriate for comparison of alternatives than techniques derived from the classical design of experiments. In the interim, considerable research has been conducted in R&S procedures in general, see Bechhofer et al. (1995); and specifically for simulation, see Goldsman and Nelson (1998). Software enabling the use of R&S procedures in simulation studies is now included in several commercial simulation products. The use of variance-reduction techniques with the design of experiments and with R&S procedures is the subject of numerous articles.

3.2.4.3. Metamodels and Optimization. Metamodels—(simple) mathematical models of the output response surface of a simulation model—have been studied in terms of both the metamodeling role (Kleijnen and Sargent 2000), and the types of models that can be used (Barton 1998).

Optimization of simulation model output has a number of complexities: The number of model variables is often large, and a variable (or parameter) can take on a large or infinite number of values. Sometimes the response surface is multimodal. Various approaches have been suggested over the years, including the use of gradient-based optimization

methods, response surface methodologies (including meta-models), pattern search methods, and random search. (With a finite number of alternatives specified, the R&S procedures discussed above are candidates.) Much of the attention to these methods involves the convergence to a local or global optimum. The sophistication and computational intensity of these methods limit their use in practice. More information on optimization can be found in recent issues of the *Proceedings of the Winter Simulation Conference*.

More recent research (1990s) has taken a different tack towards optimization: expressing the objective so as to obtain a “good” but not necessarily optimal solution. These approaches use some type of metaheuristics such as tabu search or a genetic algorithm. Several commercial simulation software systems today contain “optimization” packages based on one of these approaches.

3.2.4.4. Variance-Reduction Techniques. Variance-reduction (or reducing) techniques (VRTs) received much attention in the early days of simulation since computer time was extremely expensive, and reductions in run time represented valuable savings. The extensive use of VRTs in Monte Carlo studies suggested similar efficiencies in (discrete event) simulation. Using VRTs in the collection and analysis of data has been a focus. While a few VRTs are simple to use, most VRTs are sophisticated and model dependent, which limits their general use. A counterexample is common random numbers, often found in commercial simulation products because they are easy to understand and simple to apply. The two cases where (sophisticated) VRTs are justified: (1) simulation models investigating rare events and (2) simulation models that exact excessive computation time.

3.2.5. Theory of Simulation. As early as 1964 Lackner had proposed the use of system theory as a basis for simulation modeling. First with a journal paper in 1972, then with his book in 1976, Zeigler built an explanatory theory of simulation based on systems-theoretic concepts. This work had a major impact on those who sought to separate the expression of simulation concepts from their implementation in SPLs. The theoretical structure applied to discrete event, continuous, and combined models provided a linkage that heretofore was difficult for many to conceive.

3.2.6. Factors Contributing Jointly to Simulation and Computer Science. Although originating in simulation, at least three concepts have had a major influence in areas of computer science and the advancement of computing technology.

3.2.6.1. The Process Concept. Embodied restrictively in the GPSS transaction representation and expanded more elegantly in the SIMULA process interaction world view, the process concept is a lasting contribution to both simulation and operating systems. Its influence in simulation was to provide the realization of an entity whose dynamic behavior it sought to mimic. In the operating systems context the process presented a quasi-independent program

segment in execution, and served as a major concept underlying computational models. The process concept within the co-routine execution environment provided by SIMULA provided a powerful mechanism for expanding the ability to represent complex systems.

3.2.6.2. The Entity/Attribute/Set Concept. Introduced by Kiviat et al. (1968) in SIMSCRIPT II, this modeling perspective provided a rigorous basis for describing the static relationships among objects. Entities could be members of and owners of sets, yet each was described individually by its own attributes. Coupled with the recognition of relationships among entities, captured so expressively by Mealy (1967), the underlying concepts of the entity-relational model of data were actually present in simulation for almost 10 years before their recognition by the database community.

3.2.6.3. Object-Oriented Programming. The revision to SIMULA I, known as SIMULA 67, introduced the object-oriented paradigm (OOP) with concepts of abstract data types, encapsulation, inheritance, and message passing. The co-routine concept from the earlier version, enhanced by the OOP capabilities, promoted a very powerful style of simulation programming, so powerful that after two decades the OOP became the predominant style for programming in general. This particular factor has exerted an effect far beyond simulation, substantiated by the fact that four of the eight most significant languages selected for the 1993 History of Programming Languages II Conference (HOPL II) (see Bergin and Gibson 1996) traced their major roots to SIMULA (Ada, C++, CLU, and Smalltalk). Despite its impact on programming languages in general, one of the SIMULA co-developers (Nygaard 1978) remarked to REN that only those who had used SIMULA as a simulation language truly understood its power.

3.2.7. Combined Simulation. The evolution of GASP enabled combined continuous and discrete modeling in GASP IV through the work of Pritsker (1974). Pritsker and his students worked out the detailed transitional relationships between the continuous and discrete computations by adding the necessary subroutines to the earlier version of the language (GASP II) and providing an alternative definition of the term “event.” (An interesting trivia note is that some early copies of the book had the title “The GASP IV Language Simulation” printed on the spine and the correct title on the front of the book jacket.) (Pritsker’s numerous contributions to simulation extended over a major part of its history (Wilson and Goldsman 2001).)

3.2.8. Parallel Simulation. Initiated by research in the late 1970s and early 1980s by Bryant (1977), Peacock et al. (1979), Chandy and Misra (1979), and Jefferson and Sowizral (1982), parallel simulation became a major area of research in the middle to late 1980s and extending into the 1990s. Fueled by the time warp concept formulated by Jefferson and Sowizral (1982), research using this optimistic protocol in which checkpointing with rollback and recovery is required, was contrasted with the conservative protocol of Bryant (1977) and Chandy and Misra (1979), where

no events were executed unless correct temporal ordering was guaranteed. A series of conferences bearing the name PADS (Parallel and Distributed Simulation) began in 1985, and the *Proceedings* contains much of this research. Strong tensions between the proponents of the two protocols marked the early conferences, but now have all but disappeared. The intense early interest in the subject has also waned considerably.

A special issue of the *ORSA Journal on Computing* in 1993, guest edited by Richard Fujimoto and devoted to parallel discrete event simulation, raised the issue of why parallel simulation has not been accepted in the broader domain of simulation practice. Numerous answers were offered, both within and outside the PADS community, but no consensus has formed.

3.2.9. Distributed Interactive Simulation. Enabled by network computing advances, the concept of distributed interactive simulation originated in the military domain for training, and the visionary concept is generally attributed to Thorpe, circa 1978 (see Cosby 1995). Major funding by ARPA/DARPA permitted the demonstration that remotely executing simulation models could communicate, although major questions still remain regarding the correct representation of temporal causality and trade-offs between the level of model fidelity versus the cost of the training experiment. The early Distributed Interactive Simulation (DIS) and Aggregate Level Simulation Protocol (ALSP) are being supplanted by the High Level Architecture (HLA) protocol, intended to enable interoperability among DOD simulation models.

4. ORGANIZATIONAL FACTORS

4.1. Conferences and Symposia

Interest in the new field of computer simulation and excitement about future prospects was high in the late 1950s. Meetings provided the prime venue for communications. One of the first was the System Simulation Symposium held in 1957, to be followed closely by the Second Symposium on System Simulation in 1959. Another of note was the IBM Scientific Computing Symposium on Simulation Models and Gaming held in December 1964 with 175 attendees. Workshops on simulation languages were held at Stanford University in 1964 and at the University of Pennsylvania in 1966, the latter having 110 attendees. The NATO-sponsored conference on digital simulation held in Hamburg in 1965 had 180 attendees. A symposium on "The Design of Computer Simulation Experiments," sponsored by the The Institute of Management Sciences (TIMS) College on Simulation and Gaming, was held at Duke University in 1968 (RGS attended along with 250 others). IBM's SHARE User Group set up a System Simulation Project that held meetings where changes to GPSS were discussed. Three members of this group, H. Hixson, A. Ockene, and J. Reitman, feeling the need for a national conference on the applications of simulation, organized the "Conference on the Applications of Simulation Using GPSS" in November

1967 in New York City. A planned attendance of 225 and an actual of 401 encouraged a successor, held in December 1968. The follow-up conference was called "The Second Conference on the Applications of Simulation" (note the removal of GPSS); a proceedings was issued; and the attendance numbered 856.

The two application-oriented conferences above began what is now called the Winter Simulation Conference (WSC). See Crain et al. (1992) and Araten et al. (1992) for a history of the WSC through its 25th anniversary in 1992 and (<http://www.wintersim.org/article.htm>) for a regularly updated overview of the conference and its more recent history. The WSC, held each December, is the premier conference in simulation, attracting international attendees drawn from researchers, practitioners, and simulation software vendors. Considered a "model" conference, the WSC is sponsored by several societies, including INFORMS, and is run by volunteers. The conference attracts high quality papers, publishes electronic and hard-copy proceedings, and offers exhibits by vendors.

The Annual Simulation Symposium, initiated in 1968 by Ira Kay, is a single-track conference. Both REN, who presented a paper at the 1969 meeting, and RGS, who attended in 1979, recall the very smooth and efficient operation of the conference. Operating now under Society for Modeling and Simulation International (SCS) sponsorship, the symposium lays claim to being the longest continuously running simulation conference.

In response to growing interest in the modeling of computer systems for performance evaluation in the 1970s, the Association for Computing Machinery (ACM) Special Interest Group on SIMulation (SIGSIM) and the National Bureau of Standards cosponsored a series of symposia (with proceedings) and workshops on the topic. RGS still recalls the enthusiastic response by attendees to his tutorial paper on the statistical analysis of simulation output data at the 1976 symposium. Specialized languages for computer systems simulation were developed during this period, and a Federal agency, FEDSIM with Philip Kiviat as director, was established for computer system performance improvement.

Two conferences on simulation research were held by SICSIM of NYC (Special Interest Committee for SIMulation of New York City) under Nabil Adam's leadership. Papers from the first led to the publication of a book, Adam and Dogramaci (1979), and those from the second to a special issue of *Communications of the ACM* (April 1981). Adam, REN, and RGS organized a follow-up conference sponsored by ORSA and SICSIM of NYC that led to a special issue of *Operations Research* (November–December 1983).

Tuncer Ören was extremely active in the organization of meetings in Europe, notably a 1979 workshop on the standardization of simulation languages in St. Agata, Italy. REN remembers observing Harry Markowitz reading Zeigler's *Theory of Modelling and Simulation* during a conference break. Discussing the book later, Markowitz

dryly remarked as to how unthinking of people to change the field on him during the past 15 years.

François Cellier (1982) renewed the European model of the conference with the papers appearing in a book (as had the NATO conference proceedings earlier). Ören, Maurice Elzas, and Bernard Zeigler organized a series of four conferences following this model. RGS and REN, participants in some of these conferences, recall that several outstanding papers were presented, leading to interesting and lively discussions.

Described above as areas of intense research activity, parallel and distributed simulation spawned a conference bearing the acronym "PADS" that since the mid-1980s has occurred (with proceedings) almost annually. Technically related to PADS is the Simulation Interoperability Workshop (SIW), a semiannual meeting encompassing a broad range of modeling and simulation issues, applications, and communities (see <http://siso.sc.ist.ucf.edu/siw/>).

A dozen or more conferences with simulation in the title or featuring simulation as a major topic are offered annually, sponsored by numerous organizations. The Summer Computer Simulation Conference (SCSC) is a complementary conference to the WSC. Sponsored by SCS and originally limited to continuous simulation, the scope in recent years has expanded to include discrete event simulation. Several multiconferences are sponsored or cosponsored by SCS, with a typical format of concurrent one- and two-day miniconferences involving different technical and applications topics.

4.2. Professional Organizations

Simulation draws its professional lifeblood from special interest groups within larger societies. The principal group today is the INFORMS College on Simulation (CS), founded in 1963 as the College of Simulation and Gaming (CSG) within TIMS. Another group is SIGSIM of ACM, formed in 1967 and extremely active in the 1970s and 1980s. Preceding both TIMS/CSG and ACM/SIGSIM was SCS, originally founded as Simulation Councils Inc. in 1952 under the leadership of John McLeod. Originally limited to continuous simulation, SCS now includes all types and forms of simulation. Within the Institute of Electrical and Electronic Engineers (IEEE) are the Computer Society and the Systems, Man, and Cybernetics Society. Both have strong interests in simulation, and the former has a subgroup with the title Technical Committee on Simulation (TCSIM). Simulation is also an area of technical and publication interest for the Institute of Industrial Engineers (IIE).

4.3. Simulation Coverage in Journals

Various professional journals established departments to handle simulation papers in the mid-to-late 1970s: *AIIE Transactions* in 1976 with REN as editor, *Management Science* in 1978 with George Fishman as editor, *Operations*

Research in 1978 with REN as editor, and the *Communications of ACM (CACM)* in 1980 with RGS as editor. (*CACM* transitioned from a focus on research papers to informative articles in the late 1980s.) When the *ORSA* (now *INFORMS*) *Journal on Computing* was established in 1989, it contained an area on simulation with REN as its editor.

SCS established an archival journal in 1984 called *Transactions of the Society for Computer Simulation* to handle both continuous and discrete event simulation. The journal *Transactions on Modeling and Computer Simulation (TOMACS)*, devoted primarily to discrete event simulation, was established by ACM in 1990 with REN as editor-in-chief. The current publication venue offers numerous archival journals where simulation papers can be published.

From 1988 to 1994 Paul Fishwick maintained *Simulation Digest*, the first online publication devoted explicitly to simulation interests. Another online publication bearing the same name was launched during the 1988–1990 time frame jointly by Fishwick (Chair of TCSIM) and Stephen Roberts (Chair of SIGSIM) to serve as a joint organizational newsletter.

5. CONCLUDING SUMMARY

In drawing this trace of the evolution of simulation to a close, comments on three aspects are offered. The first concerns the breadth and extent of simulation applications; the second relates to the differences in views of scholarly depth of simulation research; and the third pertains to the future of simulation.

5.1. Applications: What Cannot Be Simulated?

From the beginning, the ingenuity and innovation with which simulation is applied has proved impressive. The adoption of simulation in numerous fields has created a disturbing side effect as well: The term is often inappropriately used. An early book, Shapiro and Rogers (1967), provides a fascinating collection of papers on self-reproducing systems, the use of graphics in studying dynamic system instability, associative processor design, simulation of the human aorta for studying artificial blood pumps, design of a parallel network computer, torpedo performance analysis, and so forth. A common misconception was, and still persists, that any computational process produces a "simulation." Nevertheless, proper applications of the technique have abounded.

A collection of papers on simulation in the social sciences, economics, business, and public administration, with a foreword by Herbert Simon, is given in Guetzkow et al. (1972). Models ranging from neurotic processes of an individual to the interaction of nations in producing the events leading to World War I are described. Bonini's (1963) aggregative model of a hypothetical firm is a stark contrast with the systems (nee industrial) dynamics models of Forrester (1961). Cognitive process simulation is described in the early artificial intelligence classic by Feigenbaum

and Feldman (1963). Psychosocial models of individual and group behavior for performance analysis and prediction are described in Siegel and Wolf (1969). These citations are but a meager sample.

5.2. Simulation As a Scholarly Activity

Until the late-1960s, in the perceptions of RGS and REN, fundamental developments in simulation were readily accepted in the scientific literature. About this time an attitude of scholarly disrespect seemed to emerge. Many professionals in management science and operations research cast simulation as the “method of last resort” and expressed the view that “anyone could do it.” Unfortunately, the belief that simulation was simply a programming exercise led to that conviction becoming widespread among those who understood neither simulation nor computer programming.

Did this pejorative view arise from the misuse of the term “simulation,” the far-reaching applications in so many diverse fields, or the preoccupation of OR and MS with mathematical sophistication? Or did it stem from a combination of these factors? The answer is not obvious, but the emergence of simulation departments (or areas) in the archival journals in the late 1970s gave evidence of a reputation regained. An increasing number of simulation researchers were finding outlets for quality publications.

5.3. The Future of Simulation

Computer graphics, virtual reality, and virtual environments are defining new vistas for simulation, but at the same time creating threats to overwhelm it. Entertainment uses and extensions of the technique offer financial inducements that are mind-boggling. Real-time and web-based models can expand and extend the impact far beyond its current level. At the same time, simulation-based acquisition and medical training applications impose requirements that appear daunting. Perhaps we are on the verge of achieving that which J. C. R. Licklider (1967) predicted some 35 years ago:

In their dynamic form, however, computer-program models appeal to the recipient's understanding directly through his perception of dynamic behavior. That mode of appeal is beyond the reach of ordinary documents. When we have learned how to take good advantage of it, it may—indeed, I believe it will—be the greatest boon to scientific and technical communication, and to the teaching and learning of science and technology, since the invention of writing on a flat surface.

ACKNOWLEDGMENTS

The authors wish to thank the referees for their many suggestions that have proved very beneficial to the authors and considerably improved this paper.

REFERENCES

Adam, N., A. Dogramaci, eds. 1979. *Current Issues in Simulation*. Academic Press, New York.

- Alexopoulos, C., A. F. Seila. 2000. Output analysis for simulations. J. A. Joines, R. R. Barton, K. Kang, P. A. Fishwick, eds. *Proc. 2000 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 101–108.
- Araten, M., H. G. Hixson, A. C. Hoggatt, P. J. Kiviat, M. F. Morris, A. Ockene, J. Reitman, J. M. Sussman, J. R. Wilson. 1992. The winter simulation conference: Perspectives of the founding fathers. J. Swain, D. Goldsman, R. Crain, J. Wilson, eds. *Proc. 1992 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 37–62.
- Balci, O., R. G. Sargent. 1980. Bibliography of validation of simulation models. *Newsletter—TIMS College on Simulation and Gaming* 4(2) 11–15.
- , ———. 1981. A methodology for cost-risk analysis in the statistical validation of simulation models. *Comm. ACM* 24(4) 190–197.
- Barton, R. R. 1998. Simulation metamodels. D. J. Medeiros, E. F. Watson, J. S. Carson, M. S. Manivannan, eds. *Proc. 1998 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 167–174.
- Bechhofer, R. E., T. J. Santner, D. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*. John Wiley, New York.
- Bell, P. C., R. M. O'Keefe. 1994. Visual Interactive Simulation: A methodological perspective. O. Balci, ed. *Annals of Operations Research, Volume 53: Simulation and Modeling*, J. C. Baltzer AG, Science Publishers, Basel, Switzerland.
- Bergin, Jr., T. J., R. G. Gibson, Jr., eds. 1996. *History of Programming Languages—II*. ACM Press, New York.
- Bonini, Charles P. 1963. *Simulation of Information and Decision Systems in the Firm*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Bryant, R. E. 1977. Simulation of packet communication architecture computer systems. M.S. thesis. MIT Lab of Computer Science, Technical Report MIT-LCS-TR-188. Massachusetts Institute of Technology, Cambridge, MA.
- Cellier, F. E. ed. 1982. *Progress in Modelling and Simulation*. Academic Press, London, U.K.
- Chandy, K. M., J. Misra. 1979. Distributed simulation: A case study in design and verification of distributed programs. *IEEE Trans. Software Engrg.* SE-5(5) 440–452.
- Chick, S. E. 2000. Bayesian methods for simulation. J. A. Joines, R. R. Barton, K. Kang, P. A. Fishwick, eds. *Proc. 2000 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 109–118.
- Churchman, C. West. 1963. An analysis of the concept of simulation. A. C. Hoggatt, F. E. Balderston, eds. *Symposium on Simulation Models*. South-Western Publishing Co., Cincinnati, OH.
- Conway, R. W. 1963. Some tactical problems in digital simulation. *Management Sci.* 10(1) 47–61.
- Cosby, L. N. 1995. SIMNET: An insider's perspective, IDA Document D-1661, Institute for Defense Analyses, Defense Technical Information Center, Washington, D.C.
- Cota, B. A., R. G. Sargent. 1990. Control flow graphs: A method of model representation for parallel discrete event simulation. CASE Center Technical Report No. 9026, Syracuse University, Syracuse, NY.
- , D. G. Fritz, R. G. Sargent. 1994. Control flow graphs as a representation language. J. D. Tew, M. S. Manivannan, D. A. Sadowski, A. F. Seila, eds. *Proc. 1994 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 555–559.

- Crain, R. C., J. O. Henriksen, S. D. Roberts, T. J. Schriber, J. M. Susman. 1992. The winter simulation conference: Celebrating twenty-five years in progress. J. Swain, D. Goldsman, R. Crain, J. Wilson, eds. *Proc. 1992 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 37–62.
- Devroye, L. 1986. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.
- Efron, B. 1982. *The Jackknife, the Bootstrap, and Other Resampling Plans*. Society for Industrial and Applied Mathematics, Philadelphia, PA.
- , R. J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York.
- Evans, G. W., II, G. F. Wallace, G. L. Sutherland. 1967. *Simulation Using Digital Computers*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Feigenbaum, E. A., J. Feldman, eds. 1963. *Computers and Thought*. McGraw-Hill Book Company, New York.
- Forrester, J. W. 1961. *Industrial Dynamics*. The M.I.T. Press, Cambridge, MA.
- Fritz, D. G., R. G. Sargent. 1995. An overview of hierarchical control flow graph models. C. Alexopoulos, K. Kang, W. R. Lilegon, D. Goldsman, eds. *Proc. 1995 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 1347–1355.
- Fujimoto, R. M. 1993. Guest editor. *ORSA J. Comput.* **5**(3)
- Glynn, P. W. 1989. A GSMP formalism for discrete event systems. *Proc. IEEE* **77**(1) 14–23.
- Goldsman, D., B. L. Nelson. 1998. Comparing systems via simulation. J. Banks, ed. *The Handbook of Simulation*. John Wiley, New York, 273–306.
- Guetzkow, H., P. Kotler, R. L. Schultz. 1972. *Simulation in Social and Administrative Science—Overviews and Case-Examples*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Haas, P. J. 1999. On simulation output analysis for generalized semi-Markov processes. *Comm. Statist.—Stochastic Models* **15**(1) 53–80.
- Hammersley, J. M., D. C. Handscomb. 1964. *Monte Carlo Methods*. Methuen & Co. LTD, London, U.K.
- Hartson, H. R. 1998. Human-computer interaction: Interdisciplinary roots and trends. *J. Systems and Software* **43** 103–118.
- Jansson, Birger. 1966. *Random Number Generators*. Victor Pettersons, Stockholm, Sweden, 15.
- Jefferson, D., H. Sowizral. 1982. Fast and concurrent simulation using the time warp mechanism, part I: Local control. RAND Report N-1906-AF, RAND Corporation, Santa Monica, CA.
- Kiviat, P. J. 1969. Digital computer simulation: Computer programming languages. RAND Memo RM-5883-PR, RAND Corp., Santa Monica, CA.
- , R. Villanueva, H. M. Markowitz. 1968. *SIMSCRIPT II Programming Language*. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Kleijnen, J. P. C. 1975. *Statistical Techniques in Simulation: Part II*. Marcel Dekker, Inc., New York.
- , R. G. Sargent. 2000. A methodology for fitting and validating metamodels in simulation. *Eur. J. Oper. Res.* **120** 14–29.
- Knuth, D. E. 1997. *The Art of Computer Programming/Volume 2: Semi-Numerical Algorithms*, 3rd ed. Addison-Wesley, Reading, MA.
- Krasnow, H. S. 1967. Dynamic representation in discrete interaction simulation languages. S. H. Hollingsdale, ed. *Digital Simulation in Operational Research*. American Elsevier Publishing Company, Inc., New York, 77–92.
- Lackner, M. R. 1962. Toward a general simulation capability. *Proc. 1962 Spring Joint Comput. Conf.* American Federation of Information Processing Societies, Montvale, NJ, 1–14
- . 1964. Digital simulation and system theory. SDC Document SP-1612, System Development Corp., Santa Monica, CA.
- Law, A. M., W. D. Kelton. 2000. *Simulation Modeling and Analysis, Third Ed.* McGraw-Hill, New York.
- L'Ecuyer, P. 1998. Random number generation. J. Banks, ed. *Handbook on Simulation*. John Wiley, New York, 93–137. Available at <http://www.iro.umontreal.ca/~lecuyer/> (myftp/papers/handsim.ps) (accessed October 30, 2001).
- . 2001. Software for uniform random number generation: Distinguishing the good and the bad. B. A. Peters, J. S. Smith, M. W. Rohrer, D. J. Medeiros, eds. *Proc. 2001 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 95–105. Available at <http://www.iro.umontreal.ca/~lecuyer/myftp/> (papers/wsc01rng.pdf) (accessed October 30, 2001).
- Lehmer, D. H. 1951. Mathematical Methods in Large-Scale Computing Units. *Ann. Comput. Lab., Harvard Univ.*, **26**, 141–146.
- Licklider, J. C. R. 1967. Interactive dynamic modeling. G. Shapiro, M. Rogers, eds. *Prospects for Simulation and Simulators of Dynamic Systems*. Spartan Books, New York.
- McCormack, W. M., R. G. Sargent. 1981. Analysis of future event set algorithms for discrete event simulation. *Comm. ACM* **24**(12) 801–812.
- Mealy, G. H. 1967. Another look at data. *Proc. 1967 Fall Joint Comput. Conf.* American Federation of Information Processing Societies, Montvale, NJ, **31** 525–534.
- Nance, R. E. 1971. On time flow mechanisms for discrete system simulation. *Management Sci.* **18**(1) 59–73.
- . 1979. Model representation in discrete event simulation: Prospects for developing documentation standards. N. Adam, A. Dogramaci, eds. *Current Issues in Computer Simulation*. Academic Press, New York, 83–97.
- . 1981. The time and state relationships in simulation modeling. *Comm. ACM* **24**(4) 173–179.
- . 1983. A tutorial view of simulation model development. S. D. Roberts, J. Banks, B. Schmeiser, eds. *Proc. 1983 Winter Simulation Conf.* Institute of Electrical and Electronic Engineers, Piscataway, NJ, 325–331.
- . 1996. A history of discrete event simulation programming languages. T. J. Bergin, R. G. Gibson, eds. *History of Programming Languages—II*. ACM Press, New York, 369–427.
- Naylor, T. H. 1971. *Computer Simulation Experiments with Models of Economic Systems*. John Wiley & Sons, Inc., New York.
- , J. L. Balintfy, D. S. Burdick, K. Chu. 1966. *Computer Simulation Techniques*. John Wiley & Sons, Inc., New York.
- , J. M. Finger. 1967. Verification of computer simulation models. *Management Sci.* **14**(10) 92–101.
- Nygaard, K. 1978. Personal communication during the ACM SIGPLAN History of Programming Languages Conference, June 1–3, Los Angeles, CA.
- O'Keefe, R. M. 1986. Personal comment during an open discussion following a technical session at 1986 Winter Simulation Conference, December 8–10, Washington, D.C.
- Peacock, J. K., J. W. Wong, E. G. Manning. 1979. Distributed simulation using a network of processors. *Comput. Networks* **3**(1) 44–56.

- Pritsker, A. B. 1974. *The GASP IV Simulation Language*. John Wiley & Sons, Inc., New York.
- Reitman, J., D. Ingerman, J. Katzke, J. Shapiro, K. Siman, B. Smith. 1970. A complete interactive simulation environment: GPSS/360-NORDEN. *Proc. Fourth Conf. Appl. Simulation*. Institute of Electrical and Electronics Engineers, Piscataway, NJ, 260–270.
- Sargent, R. G. 1979. Validation of simulation models. H. J. Highland, M. G. Spiegel, R. E. Shannon, eds. *Proc. 1979 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 497–503.
- . 1981. Verification and validation of simulation models. F. E. Cellier, ed. *Progress in Modelling and Simulation*. Academic Press, London, U.K.
- . 1988. Event graph modelling for simulation with an application to flexible manufacturing systems. *Management Sci.* **34**(10) 1231–1251.
- , K. Kang, D. Goldsman. 1992. An investigation of finite-sample behavior of confidence interval estimators. *Oper. Res.* **40**(5) 898–913.
- Schruben, L. 1983. Simulation modeling with event graphs. *Comm. ACM* **26**(11) 957–963.
- Shanthikumar, J. G., R. G. Sargent. 1983. A unifying view of hybrid simulation/analytic models and modeling. *Oper. Res.* **31**(6) 1030–1052.
- Shapiro, G., M. Rogers, eds. 1967. *Prospects for Simulation and Simulators of Dynamic Systems*. Spartan Books, New York.
- Siegel, A. I., J. J. Wolf. 1969. *Man-Machine Simulation Models—Psychosocial and Performance Interaction*. John Wiley and Sons, Inc., New York.
- Som, T. K., R. G. Sargent. 1989. A formal development of event graphs as an aid to structured and efficient simulation programs. *ORSA J. Comput.* **1**(2) 107–125.
- Tocher, K. D. 1963. *The Art of Simulation*. The English Universities Press LTD, London, U.K.
- , D. G. Owen. 1960. The automatic programming of simulations. *Proc. Second Internat. Conf. Oper. Res.* 50–68.
- U.S. General Accounting Office. 1973. *Advantages and Limitations of Computer Simulation in Decision Making*. B-163074. Washington, D.C.
- . 1976. *Ways to Improve Management of Federally Funded Computerized Models*. LCD-75-111. Washington, D.C.
- Wilson, J. R., D. Goldsman. 2001. Alan Pritsker's multifaceted career: Theory, practice, education, entrepreneurship, and service. *IIE Trans.* **33**(3) 139–147.
- Zeigler, B. P. 1972. Towards a formal theory of modelling and simulation: Structure preserving morphisms. *J. Assoc. Comput. Mach.* **19**(4) 742–764.
- . 1976. *Theory of Modelling and Simulation*. John Wiley & Sons, Inc., New York.