Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

THE STATE OF INNOVATION IN MODELING AND SIMULATION: THE LAST 50 YEARS

Osman Balci

Department of Computer Science
638 McBryde Hall, MC 0106
Virginia Tech
Blacksburg, VA 24061, USA

Richard M. Fujimoto

School of Computational Science and Engineering
266 Ferst Drive
Georgia Institute of Technology
Atlanta, GA 30332, USA

David Goldsman

H. Milton Stewart School of ISyE
433 Groseclose Bldg
Georgia Institute of Technology
Atlanta, GA 30332, USA

Richard E. Nance

Department of Computer Science
225 Stanger Street
Virginia Tech
Blacksburg, VA 24061, USA

Bernard P. Zeigler

Electrical and Computer Engineering
1230 E. Speedway Blvd.
The University of Arizona
Tucson, AZ 85721, USA

ABSTRACT

Innovation in Modeling and Simulation (M&S) refers to exploiting new ideas, exploiting new technology, and employing out-of-the-box thinking, which lead to the creation of new methodologies, techniques, concepts, frameworks, and software. This paper addresses the following questions: (a) what was the state of the art in M&S 50 years ago, what is it today, how much progress has been made? (b) how much innovation in M&S has been accomplished over the last half a century? (c) what were the obstacles to innovation in M&S? (d) what are some recommendations to promote innovation in M&S? (e) what message should be sent to the funding agencies to encourage innovation in M&S?

1 INTRODUCTION

With the advent of computers, simulation became a viable approach for solving complex problems. It was first named depending on the type of computer used as *Analog Computer Simulation* or *Digital Computer Simulation*. Then the name became Computer Simulation. The *Society for Computer Simulation* (SCS) was founded in 1952. The discipline was generally referred to as Simulation. In early 1990s, the U.S. Department of Defense adopted the name Modeling and Simulation (M&S) to emphasize the importance of modeling in simulation. SCS changed its name in 2000 to the *Society for Modeling and Simulation International*. Today, the discipline is commonly referred to as M&S.

M&S is used to solve complex problems in almost all disciplines today, like mathematics is used. Using the term M&S or just Simulation alone does not reflect the kind of M&S used and causes communication problems. The type of M&S must be clearly identified in any technical writing or discussion. Table 1 lists 18 types of M&S in use today (Balci, Arthur, and Ormsby 2011; SIGSIM 2017).

Table 1: Types of M&S

A.	Based on Model Representation	
	1.	Discrete M&S
	2.	Continuous M&S
	3.	Monte Carlo M&S
	4.	System Dynamics M&S
	5.	Gaming-based M&S
	6.	Agent-based M&S
	7.	Artificial Intelligence-based M&S
	8.	Virtual Reality-based M&S
	9.	System Theoretic M&S
B.	Based on Model Execution	
	10.	Distributed / Parallel M&S
	11.	Web-based (Cloud-based) M&S
C.	Based on Model Composition	
	12.	Live Exercises
	13.	Live Experimentations
	14.	Live Demonstrations
	15.	Live Trials
D.	Based on What is in the Loop	
	16.	Hardware-in-the-loop M&S
	17.	Human-in-the-loop M&S
	18.	Software-in-the-loop M&S

The following simulation programming approaches (also called world views or simulation strategies) were invented in 1960s: (1) event scheduling, (2) activity scanning, (3) three-phase approach, and (4) process interaction (Balci 1988). The event scheduling approach was so favored due to its execution efficiency that this M&S type has been called Discrete Event Simulation. However, a simulation model can be programmed under a paradigm of activity, event, object, or process. Thus, we use Discrete Activity Simulation, Discrete Event Simulation, Discrete Object Simulation, or Discrete Process Simulation. Why continue to use the term Discrete Event Simulation to incorrectly refer to other kinds of discrete simulation that are not event-based? It should generally be called Discrete Simulation to accommodate all types of discrete simulation programming approaches.

Can we all agree to make that change? Change is the only constant required for innovation. Without change, we cannot correct our terminology problems, employ out-of-the-box thinking, exploit new ideas, or exploit new technologies.

This paper does not cover innovation in all types of M&S, but just for some. Leading experts in some areas of M&S express their opinions about the state of innovation in M&S during the last half a century in the remaining sections of this paper.

2 INNOVATION IN PARALLEL AND DISTRIBUTED SIMULATION (Richard M. Fujimoto)

2.1 The Past: The Process of Innovation

The parallel and distributed simulation field began with seminal work occurring in the late 1970s. It is an area that has seen a tremendous amount of innovation over the years beginning with the definition of specific problems and research challenges and going on to the creation of innovative solutions and development of techniques and tools to apply the technology to real-world problems. It continues to be a vibrant and active area of research and practice to this day. As such, it is instructive to consider how innovation in this area has come about over the last four decades in order to gain some insight concerning how innovation might be created in the future.

Where do new ideas come from? Johnson describes a process termed the adjacent possible derived from evolutionary biology (Johnson 2011). In essence, this theory characterizes the creation of new ideas as something akin to the creation of new species; just as new species are created from existing species intermingling with each other and adapting in the face of change through Darwinian natural selection, Johnson argues that new ideas come about by existing ideas and concepts bumping into each other resulting to form new ones. Further, it is true that *necessity is the mother of invention*. New ideas often come about in the context of solving a specific problem. In many cases, innovation occurs when existing ideas and concepts developed in one field are transplanted into a new field. For example, Johnson cites the invention of the printing press which involved repurposing grape presses used to make wine among other innovations, and the case of an obstetrician named Stephane Tarnier in Paris who upon visiting a zoo found an exhibit of chicken incubators that led to the development of an incubator for human babies that has saved countless numbers of lives. Johnson's characterization of ideas colliding with each other and ideas transplanted into new fields combined with adaptation in the context of solving a specific problem provides a credible explanation of the origins of innovation.

The parallel and distributed simulation field followed this pattern. It was born from the development of new underlying hardware platforms. Specifically, the creation of the parallel discrete event simulation field originated when commercial multiprocessor platforms were first becoming available in the 1970s and 1980s. Similarly, distributed simulation emerged in the 1980s with the SIMNET project when local area networks became available, making possible the interconnection of training simulators (Miller and Thorpe 1995).

Parallel discrete event simulation (PDES) began as a field in the late 1970's by first defining the synchronization problem along with associated concepts and terminology (e.g., logical processes) and the development of algorithmic solutions. Seminal work in the field resulted in two different approaches. The first, now called conservative synchronization algorithms, resulted from work by two groups working independently, without knowledge of each other. K. Mani Chandy and Jay Misra at the University of Texas in Austin (Chandy and Misra 1979), and Randy Bryant (Bryant 1977) then a master degree student at MIT developed what is now referred to as the Chandy/Misra/Bryant (CMB) algorithm. A few years later, David Jefferson working in collaboration with Henry Sowizral at the Rand Corporation came up with an entirely different approach known as Time Warp (Jefferson 1985). Time Warp resulted in a class of methods referred to today as optimistic synchronization. Conservative and optimistic synchronization approaches remain the major classes of algorithms used in PDES to this day. Interestingly, Jefferson was unaware of the work by Chandy, Misra, and Bryant when he invented Time Warp, and later remarked that had he known about their work, he likely would not have invented Time Warp, as his thinking would have been steered into an entirely different direction (Jefferson 2016). There have been, of course, many other innovations developed in the field, but these are perhaps distinguished not only by the solutions that were developed, but in defining the synchronization problem in the first place.

Looking back, we see that these innovations followed many of the principles articulated by Johnson. New ideas emerged from the confluence of new hardware technologies that were developing at the time, coupled with specific problems that needed to be addressed. Chandy and Misra were motivated by the problem of simulating queueing networks, Bryant by telecommunications systems, and Jefferson by wargame simulations (Fujimoto et al. 2017). The parallel discrete event simulation community was faced with the question of how to distribute a simulation across multiple processors to accelerate its completion while still obtaining the same results as a sequential execution. The distributed simulation community confronted a different problem – how to enable simulations to interoperate to create seamless virtual environments for its participants. In both cases concepts and principles from parallel and distributed computing, modeling and simulation, and specific applications were combined to lead to new computational problems and innovative solutions.

2.2 The Future

While it is impossible to predict what innovations will arise in the future, one might suspect it will arise through much the same process that has driven innovation in the past: new ideas emerging from technological innovations in different areas and applications developed to address specific problems. Key emerging technologies impacting the parallel and distributed simulation field today include:

- *Massively parallel, heterogeneous supercomputing platforms*. Limitations in integrated circuit technology have resulted in a stagnation of clock speed improvements since 2005. Performance improvements are now coming from increased parallelism and exploitation of specialized hardware such as graphical processing units (GPUs) and field programmable gate arrays (FPGA).
- *Cloud computing and the Internet of Things*. The emergence of mobile and cloud computing has led to the creation of a new computational platform consisting of computers embedded in the real world, operating on live data streams, and performing local computations as needed, coupled with reach-back into the cloud for additional computing and storage capability as needed. Sometimes called the "third platform," with mainframe computers being the first, and client/server distributed computing being the second, these platforms will continue to have large impacts in society into the foreseeable future.
- *Big data*. Much has been written concerning the rapid growth in the amount of data being generated, and the need for new computational approaches to address associated challenges. Data and modeling and simulation have always been closely linked, with data necessary to create the model, and data analysis techniques needed to glean information and insight from results produced by simulation runs. Advances in machine learning can be synergistic with parallel and distributed simulation to enable new approaches to analyzing and improving complex systems.
- *Application challenges*. Many areas are seeing new disruptive technologies, often coupled with the above computational technologies, having strong impacts. Examples include the emergence of connected and automated vehicles, commercial drones, smart systems such as power grids and homes, and crowd-sourcing impacting industries such as taxi service and short-term housing, to mention a few.

In (Fujimoto 2016) six research challenges in parallel and distributed simulation are discussed. New innovation is likely to arise in at least these areas:

- *Application-driven, scalable simulations of large, complex networks*. Techniques to efficiently simulate very large networks with irregular topologies are needed. Such networks often arise in practice in modeling telecommunications, social interaction networks, and a host of other applications.
- *Exploitation of heterogeneous machine architectures using Graphics Processing Units (GPUs)*. Exploitation of parallel and distributed simulation methods on non-traditional computing architectures is beginning to receive some attention. Major challenges remain with respect to performance and programmability.

- *Making parallel and distributed simulation broadly accessible through simpler model development and cloud computing platforms*. Accessibility to parallel and distributed simulation techniques continues to be limited to experts in the field; new innovations are needed to make its use more widespread in the general modeling and simulation community.
- *Online decision-making using real-time distributed simulation*. Synergies with the Internet of Things and machine learning technologies create numerous opportunities for innovation in the use of parallel and distributed simulations to monitor and manage operational systems. This requires increased automation of the various steps involved in the life cycle of a modeling and simulation study.
- *Energy and power efficient parallel and distributed simulation*. Energy and power consumption by computing and communications technologies is an area of increasing importance with the increased use of battery powered mobile platforms and constraints such as power caps in supercomputers and data centers. This represents a new area for the parallel and distributed simulation community where there are many opportunities for innovation.
- *Rapid composition of distributed simulations*. Rapid composition remains an important problem in the distributed simulation community. The question of how to develop simulations now for later reuse in situations that cannot be predicted requires the development of new innovative techniques, particularly early in the simulation life cycle, e.g., development of conceptual models. The need to address problems in emerging areas, such as interdependencies among urban infrastructures and social processes, provides fertile ground for innovation to address the composition problem.

More broadly, a recent workshop articulated research challenges in the general modeling and simulation field, beyond issues concerning parallel and distributed simulation. These challenges spanned four main areas: (1) conceptual models, (2) computational issues, (3) quantifying and managing uncertainty, and (4) reuse of models and simulations (NSF/NASA/AFOSR/NMSC/NTSA Workshop 2016). Likely new innovation in the broader modeling and simulation field will arise in these areas for many years to come.

Much of the innovation that has arisen in the past has been made possible by federal funding both for civilian and defense purposes. It goes without saying that such funding is essential to fuel continued development of innovations. One area that has not been emphasized in the past by traditional research projects has been the commercialization of new technologies. While there has been some significant translation of parallel and distributed simulation technologies to commercial products and services, this is an area where further development is needed.

2.3 A Few Recommendations

In closing, a few suggestions for fostering increased generation of new ideas and innovation are summarized below. In (Fujimoto 2011) several suggestions for graduate students and junior researchers on this subject are presented. The following list builds upon and refines some of these suggestions:

- *Define the problem, not just the solution*. Some of the most innovative work in parallel discrete event simulation involved defining problems such as synchronization which in turn, helped spawn a new field and a tremendous amount of innovation. The definition of new problems creates new opportunities for innovation much broader than one's own work.
- *It's the platform, stupid!* New computing platforms in the 1970's and 1980's led to the creation of parallel and distributed simulation field. Today, new platforms and technologies continue to appear, and offer many opportunities for developing new capabilities and applications for parallel and distributed simulation, and more broadly, modeling and simulation as a whole.

- *Look for innovation at the seams between fields*. Johnson points out that innovation comes about from bringing together ideas that had not been combined before to create new ideas. What better place to find new combinations than to explore different fields, and look for transplanting, integrating and adapting ideas and concepts from one field into another. The interdisciplinary nature of modeling and simulation creates an excellent environment for the development of innovation utilizing this process.
- *Don't read the literature!* At least not initially. First try to solve the problem without looking at how others have approached it to avoid retracing paths others have already explored. Jefferson's experiences in developing Time Warp is a good example. Of course, after you've come up with a solution, or given up, one must certainly review the literature to identify related work.

In summary, parallel and distributed simulation has formed the grounds for much innovation over the last four decades. With many new challenges arising from underlying platforms as well as new demands and requirements imposed by new applications of increasing complexity and scale, the field continues to be well positioned for the development of new innovations into the foreseeable future.

3 STATISTICAL ASPECTS OF MODELING AND SIMULATION (David Goldsman)

3.1 The State-of-the-Art

3.1.1 What was Happening in 1967?

Statistical analysis for simulation input and output problems was just starting to come into play during the mid-1960s. As outlined in (Goldsman, Nance, and Wilson 2009, 2010; Nance and Sargent 2002; Nelson 2004) (among other sources), it is widely acknowledged that the foundations for statistical analysis in simulation were set forth in the papers (Conway, Johnson, and Maxwell 1959; Conway 1963). Generally speaking, those early papers discussed issues related to the proper design of simulation experiments so as to address the following fundamental problems:

- The start-up problem, i.e., when is a simulation in "steady state" so that any transients caused by the simulation's initial condition have died out? For instance, how long will it take the simulation of a complicated queueing network to reach some sort of steady state if it initialized in an empty-and-idle condition?
- Estimating the precision (variance) of simulation-based estimators of steady-state performance. How confident are we about a particular point estimate of a simulation measure such as the expected steady-state waiting time?
- Comparing alternative system simulations. In other words, which is the best of a competing set of alternative systems or system configurations?

For the start-up problem, Conway (1963) proposed the first widely used rule for deleting simulation observations that are biased by the way in which the simulation is initialized. With regard to the variance-estimation problem, he proposed the method of batch means, which is still widely used in practice and is even today still the basis for a great deal of ongoing research. And for the comparison problem, Conway rejected traditional analysis of variance (ANOVA) methods and proposed the use of statistical ranking-and-selection procedures, which are now widely used in practice and are the basis for a current flurry of work. The short story is that Conway showed remarkable insight into the problems and solution strategies that would shape the next fifty years of simulation methodology.

But the Conway papers were not particularly rigorous in nature. That was about to change. The mid/late 1960s saw a nascent theory-oriented literature, whose contributors included Fishman and Iglehart (studying new output analysis methods based on spectral theory and regenerating stochastic processes), Billingsley (setting down formalisms for probabilistic methods to treat correlated data), and Hammersley and Handscomb (on Monte Carlo methods to carry out efficient simulation experimentation).

3.1.2 How About Today?

Conway would likely be tremendously impressed with the research environment of today. First of all, a clear dichotomy exists between steady-state and what are known as "terminating" (finite-horizon) systems; and those two types of systems are attacked via different methods. With respect to steady-state simulation, identification and mitigation of the start-up phase can be dealt with via a variety of graphical and statistical techniques, e.g., (i) hypothesis testing to determine if the initial portion of the simulation is somehow different than subsequent portions (perhaps in terms of the portions' means or variances), and (ii) process control and change-point detection methods crossing over from the statistics literature.

Second, we now have a rich battery of point estimators for the precision of a process as well as the resulting confidence intervals (CIs) for mean performance. Notably, these new methodologies have drawn from such theoretical foundations as Billingsley (1968)'s work on Brownian motion, Iglehart's regenerative method (see, e.g., Crane and Lemoine 1977), or Efron (1982)'s bootstrap technique. The point estimators and CIs that have come onto the scene are significantly superior to those of naïve batch means, often with convergence properties that are orders-of-magnitude better. Third, the problem of choosing the best system has produced an explosion of excellent new work, much of which is chronicled in the yearly WSC proceedings as well as in archival journals. Nowadays, we can select the best among thousands of competing systems under a variety of assumptions on the competitors that have far exceeded even those in statistical literature. Moreover, recent advances in selection algorithms are currently being adopted to attack true optimization problems involving large-scale systems with constraints, as well as so-called robust optimization problems.

In fact, the advances from our field's statistical side are not limited merely to Conway's "Big Three". There is also a great deal of terrific work in such diverse topic areas as input modeling under uncertainty, time series modeling (with ubiquitous applications to financial engineering), Monte Carlo simulation, random variate and process generation, and Bayesian analysis, just to name a few.

3.1.3 What Has Been the Progress?

As we have detailed above, the progress that has been made in the last 50 years is nothing short of amazing with respect to all of the general areas that Conway discussed back in 1963. The short story is that simulation researchers have used high-powered mathematical tools to obtain rigorous algorithms that can help simulation analysts undertake input analysis output analysis, and simulation optimization. What is especially nice is that these theoretical achievements are being married to practice by being incorporated into simulation software products that can be used on real-world problems, to the point that vendors are including certain recently developed algorithms into their commercial packages — for instance, various ranking and selection procedures, stochastic process generators, and input analysis tools.

A typical example is that of completely automatic sequential steady-state CI algorithms that can essentially be treated as a black box. Simply put, (i) the user starts the simulation; (ii) based on an initial set of samples, the algorithm truncates data that it deems as transient; (iii) now in steady-state, the algorithm continues sampling under a pre-specified CI length criterion is satisfied.

3.2 How Much Innovation Has Been Accomplished?

The level of innovation has exponentiated over the last 50 years. For example, whereas the old batch means method is simple enough to explain to an undergraduate class in just a few minutes, one could actually devote an entire semester in a graduate special topics course to the wealth of new, useful results in output analysis. Same for ranking and selection / optimization. Further, as discussed previously, Conway's original areas of study have morphed into a wide pallet of current interesting topic areas.

3.3 What Were the Obstacles to Innovation?

Until a few years ago, the obstacles were computing power, data storage capabilities, lack of proper analysis tools for "Big Data", and practitioner reluctance to use the new tools. But these problems are rapidly disappearing even as we speak, as illustrated by some of the research being reported on during the present WSC.

- Computing power is almost no longer an issue, due to cloud-based parallelization;
- Ditto for data storage, where some researchers have even argue that it makes sense to keep all data from system snapshots of every event during the simulation run;
- Big Data access and analysis algorithm are now coming into wide play; and
- Major practitioners have shown great interest in incorporating some of the new statistical methodologies into their products.

3.4 What are Your Recommendations to Promote Innovation?

The short story is that the field should keep up the good work, as exemplified by the WSC itself. We ought to continue to teach rigorous, sound theory; but mix that theory with motivating applications and accompanying software. It is this mix of all branches of the simulation family that promotes rapid advancement of the field. In addition, simulation researchers are increasingly reaching out to the Computer Science, Mathematics, and Statistics communities to increase the number of tricks of the trade, and to engage with colleagues who see things from different points of view.

3.5 What Message Would You Like to Send to the Funding Agencies to Encourage Innovation?

Theory may be the bedrock of our field, on which much is built; and so that side of things must continue to be funded in order to drive new innovations. But we should also be encouraged to partner with practitioners in order to see the fruits of our labor be put to meaningful practical use.

4 FIFTY YEARS OF MODELING METHODOLOGIES: WHEREFORE INNOVATION? (Richard E. Nance)

The panel members for this session have been challenged, as has perhaps the audience – else what motivates their presence, to reflect on the state of innovation in modeling and simulation (M&S). This reflection should be conducted within a temporal context of 50 years, and conveyed through answers to five, actually eight (when embedded interrogatories are included) questions. Each question with the response for simulation modeling methodologies forms the structural subdivision of this section.

4.1 What was the State-of-the-Art in Modeling Methodologies in 1967?

An answer to this question is provided by the several coincidences marking this particular year. The conference that became known as the Winter Simulation Conference was launched with applications in General Purpose Systems Simulation (GPSS) focus. With a more catholic perspective, the IFIP Working Conference on Simulation Programming Languages (SPLs) (Buxton 1967; Buxton and Laski 1969) and the RAND reports by Philip J. Kiviat served to define the understanding of M&S methodologies at this point (Kiviat 1963, 1967, 1969). In summary, a SPL belonged to one of four conceptual frameworks (or world views): event scheduling (ES), activity scan (AS), transaction processing (TP), and process interaction (PI) (Balci 1988). Variations, both minor and major, provided discriminations within and among the frameworks. Each framework was represented by at least one SPL: ES – SIMSCRIPT

(Markowitz, Hausner, and Karr 1962); AS – CSL; TP – GPSS (Gordon 1961); and PI – SIMULA (Nygaard and Dahl 1978). Adherence to a framework and SPL was so strong that Kiviat characterized discrete event simulation as exhibiting an inversion of theory and practice: the language defines the theory. The obvious negative consequence is that the theory is not invariant.

Simula 67, a major revision of the language in 1967, established SIMULA as the first object-oriented programming language, clearly deserving of its distinction as a major innovation in modeling methodology. Considering a SPL as the modeling methodology seems justified; however, symbols representing language operations permitted a "flow graph" characterization, particularly emphasized in GPSS.

4.2 What has Happened Over the Last 50 Years and Where are We Now?

Providing a synopsis of 50 years in a few sentences is a daunting task. Table 2 is a rough attempt to trends in simulation programming / modeling by a totally arbitrary division of the five decades. While many ideas and techniques are described in the literature published during this period, do any qualify as "innovative" to the same degree as SIMULA?

Table 2: Trends in Modeling Methodology by Selected Periods Since 1967

Period	**Perceived Modeling / Language Trends**	**Observations / Consequences**
Early 1970s	Developing simulation packages in a new general purpose language, e.g. PL/I	Continuation of earlier choice to not learn a new language.
Mid 1970s	Extension and expansion of early SPLs, GASP IV, GPSS, SIMSCRIPT II.5	Feature additions for market appeal.
Late 1970s	Consolidation of SPL features and capabilities; appearance of SLAM	Packages in general languages
Early to Mid 1980s	New SPLs emerge; PC versions of extant and new SPLs; start environments research; parallel, distributed execution;	Modeling task beyond SPL; graphical output; support tools recognized
Mid to late 1980s	New general languages spawn M&S use, e.g. C++, Ada; specialized languages adapted for M&S, e.g. Prolog; training	Model life-cycle and environments evolve for development, sustainment
Early to late 1990s	Conceptual modeling emphasis, V&V becomes VV& Accreditation: DOD stresses interoperability and HLA	Commercial environments (MDE and SSE) appear; OOP dominates as paradigm
2000 to 2009	Hybrid simulation rebirth, multi-modeling; renewed interest in system dynamics; soft systems, agent-based modeling; Internet execution	Modeling methodology tracks expand; vendor and exhibit program at WSC enlarges

4.3 What were the Obstacles to Innovation?

The way this question is framed presumes that innovation is lacking in modeling methodology over the past 50 years. That conclusion matches the implied answer to the question posed in the prior paragraph. Identifying obstacles to innovation in M&S modeling methodology calls for an almost impossible degree of introspection or a rather reckless degree of speculation. A more prudent approach is to suggest possible barriers, likely stemming from unintended consequences, in a series of questions.

- Has the emphasis on M&S as a practical problem-solving technique inhibited the recognition of the importance of an underlying theory?
- Did the early "theory and practice inversion" identified by Kiviat divert potential motivation to formulate a general modeling theory?
- Did the creation of SIMULA outside the U.S. delay the discovery and appreciation of the language and the object-oriented paradigm in general by the M&S community?
- Did the stress on interoperability and the High Level Architecture (HLA) by the U.S. Department of Defense stifle new ideas and techniques in the motivation for reusability and cost savings?
- Does the interdisciplinary nature of M&S restrict its ability to find a funding home?

Questions and discussion raised in this panel session are likely to add to this list.

4.4 What are Your Recommendations to Promote Innovation? What Message Would You Like to Send to the Funding Agencies to Encourage Innovation?

The response to both questions is that I have nothing to offer beyond the following close: I am not sure that "innovation" is the proper evaluative criterion to be invoked; the progress, or lack thereof, in modeling methodologies over the past 50 years might better be judged by the criterion of "ingenuity."

5 SYSTEM-THEORETIC APPROACH TO MODELING AND SIMULATION (Bernard P. Zeigler)

5.1 What was the State of the Art 50 Years Ago, What is It Today, How Much Progress has been Made?

To assess the state of the art of the System-Theoretic Approach to Modeling and Simulation fifty years ago we must start perhaps a decade earlier. Around that time, the first use of a form of digital simulation appeared which we can roughly identify as event-oriented simulation. At its advent, event-oriented simulation was mainly thought to be a form of programming associated with the recent introduction of the digital computer and applied to operational research problems. In contrast, classical simulation was taken to be a form of numerical solution applicable to physics and related sciences whose speed could be greatly increased with mechanical, as opposed to, hand calculation. The concept of "system" was defined by Wymore (1967) as a basis for unifying various forms of discrete and continuous model specification. About a decade after event-oriented simulation took hold, Bernard Zeigler elaborated on Wymore's (and others') system theory in the book "Theory of Modeling and Simulation" (1976.) He defined the Discrete Event System Specification (DEVS) formalism as a specification for a subclass of Wymore systems that captured all the relevant features of the models underlying event-oriented simulations. In contrast, Discrete Time Systems Specification (DTSS) and Differential Equation System Specification (DESS) were introduced to specify other common distinct subclasses of Wymore systems – the first, as a basis for discrete time models (including those specified by finite automata and cellular automata); the second to represent the continuous models underlying classical numerical solvers. K.D. Tocher appears to be the first to conceive discrete events as the right abstraction to characterize the models underlying the event-oriented simulation techniques that he and others were adopting in the mid-1950s. According to Hollocks (2008), Tocher's core idea conceived of a manufacturing system as consisting of individual components, or 'machines', progressing as time unfolds through 'states' that change only at discrete 'events'. Indeed, Zeigler's DEVS took this idea one step further in following Wymore's formalistic approach, both being based on the set theory of logicians and mathematicians (Whitehead and Russel 1910; Bourbaki 1930).

Some distinctive modeling strategies soon emerged for programming event-oriented simulation. They became encapsulated in the concept of world views: event scheduling, activity scanning, and process interaction (references will be in Nance's presentation). Zeigler (1984) formally characterized these world views showing that they could all be represented as subclasses of DEVS, thus also suggesting its universality for discrete event model formalisms extending to other representations such as Timed Automata and Petri Nets. Also at the same time the distinction between modular and non-modular DEVS was made showing that the world views all fit within the nonmodular category. Moreover, while the modular class was shown to be behaviorally equivalent to that of the non-modular one, it better supported the concepts of modularity, object orientation, and distributed processing that were impending on the software engineering horizon.

An overview of some of the milestones in the development of DEVS is given in Figure 1.

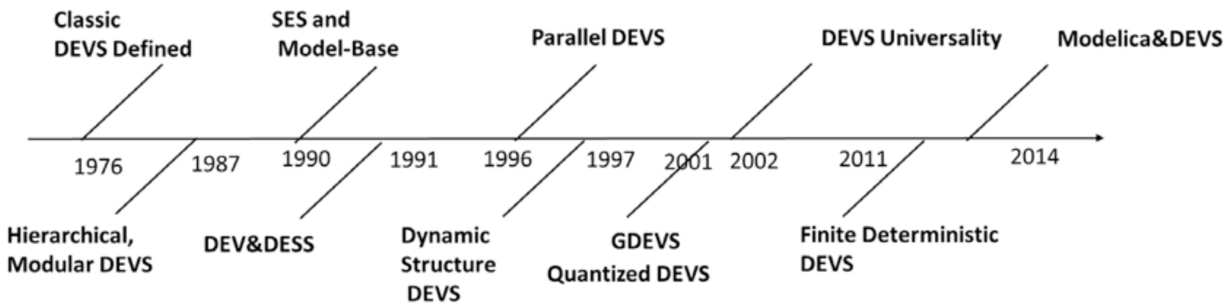

Figure 1: An overview of some of the milestones in the development of DEVS.

Classic DEVS (Zeigler 1976), as distinguished from Parallel DEVS introduced later, is a formalism for modeling and analysis of discrete event systems can be seen as an extension of the Moore machine formalism,]which is a finite state automaton where the outputs are determined by the current state alone (and do not depend directly on the input). The extension associates a lifespan with each state and provides a hierarchical concept with an operation, called coupling, based on Wymore's system theory.

Parallel DEVS (Chow 1996), revises the classic DEVS formalism to distinguish between transition collisions and ordinary external events in the external transition function of DEVS models, extends the modeling capability of the collisions. The revision also replaces tie-breaking of simultaneously scheduled events by a well-defined and consistent formal construct that allows all transitions to be simultaneously activated providing both conceptual and parallel execution benefits.

Hierarchical, Modular DEVS (Zeigler 1987) established the similarity and differences with, and implemented DEVS in the Object-oriented programming (OOP) and modular programming paradigm, among the first in numerous implementation (see list link in References).

System entity structure (SES) (Kim et al. 1990) is a structural knowledge representation scheme that contains knowledge of decomposition, taxonomy, and coupling of a system supporting model base management.

Dynamic Structure DEVS (Barros 1997), enables representing systems that are able to undergo structural change. Change in structure is defined in general terms, and includes the addition and deletion of systems and the modification of the relations among components.

DEVS considered as a universal computational formalism for systems (Mosterman and Vangheluwe 2004) found increasing implementation platforms that handled combined discrete and continuous models (also called co-simulation, hybrid simulation). Some of the milestones in this thread of development are:

- DEV&DESS (Discrete Event and Differential Equation System Specification) (Praehofer 1991) is a formalism for combined discrete-continuous modeling which based on system theoretical combines the three system specification formalisms-differential equation, discrete time, and the discrete event system specification formalism.

- Quantized State Systems (Kofman and Junco 2001), a dynamical systems are continuous time systems where the variable trajectories are piecewise constant and can be exactly represented and simulated by DEVS.
- GDEVS (Giambiasi, Escudé, and Ghosh 2000) (Generalized DEVS) organizes trajectories through piecewise polynomial segments utilizing arbitrary polynomial functions to achieve higher accuracies in modeling continuous processes as discrete event abstractions.
- Modelica & DEVS (Nutaro 2014) transforms Modelica continuous models into DEVS thus supporting models with state and time events that comprise differential-algebraic systems with high index.

Finite Deterministic DEVS (Hwang 2011) is a powerful subclass of DEVS developed to teach the basics of DEVS that has become the basis for implementations for symbolic and graphical platforms for full-capability DEVS.

This selection of milestones illustrates that much progress has been made.

5.2 How Much Innovation has been Accomplished?

One could argue that the number of milestones identified above suggests a measure of progress. However, the question implies that innovation is somehow different from progress. We interpret the question as asking for an assessment of the impact of progress on M&S more generally. Here we consider that one aspect of such impact is the increasing acceptance of non-code based descriptions of simulation models. We refer for example, to conceptual modeling use of descriptions as precursors of their coded implementations that are not necessarily computer-executable. In the cases of continuous and discrete time modeling formalisms, mathematical representation had proceeded their computerized incarnations (it has been three hundred years since Newton-Leibnitz!). However, the reverse was true for event-oriented simulation models. These models were largely prisoners of their simulation language implementations or algorithmic code expressions. Indeed, there was a prevalent belief that discrete event "world views" constituted new mutant forms of simulation, unrelated to the traditional mainstream paradigms. Fortunately, that situation has begun to change as the benefits of abstractions in control and design became clear. Witness the variety of discrete event dynamic system formalisms that have emerged (Ho 1994). Also note the influence of meta-modeling frameworks borrowed from software engineering (OMG 2015) and increasing applied to development of higher level domain specific languages (Jafer et al. 2016.) The confluence of such frameworks with the system-theory based unified DEVS development process (Mittal and Risco Martin 2013) may be yet another aspect of the innovation that has come in the emergence of non-code support of simulation model development.

5.3 What were the Obstacles to Innovation?

A primary obstacle to innovation stems from the way that disciplines form protective walls around themselves that present barriers to entry for non-accepted practitioners. Such entry barriers develop naturally as disciplines develop their specific knowledge bases, methodologies, and inevitably, perhaps, biases distinguishing the relevant and important from the irrelevant or unimportant. The walls serve useful functions in subjecting putative advances to tests of review that increase the likelihood that putative advances constitute sound knowledge. However, these barriers can also present obstacles to innovation when they allow irrational opinions to block innovation. Such irrationality may arise through habits of mind unconsciously controlling thought. This can occur for example, where unconventional form is confounded with poor content, attempts to convey complex ideas in simple manners are confused with lack or rigor, or basic (but previously implicit) concepts that lead to better understanding are dismissed out of hand as obvious or well known to the community.

Several trends might explain the phenomenon in which disciplines present barriers to those from other disciplines. In the recent past there seems to have been an explosion of fragmentation into subdisciplines perhaps fueled by the need to form sub-communities that allow new approaches to blossom. A look at the journals of one publisher reveals that it has over 300 titles including for example over 20 in modeling and simulation. The specificity of scopes for such journals make it difficult to cross barriers between them and point to the same underlying tendencies of knowledge development in earlier days. The system-theoretic approach encountered such obstacles when it was trying to break through the existing computer science and physical science disciplinary biases that it encountered.

5.4 What are Your Recommendations to Promote Innovation?

Innovations have occurred in the commercial world that perhaps could not have emerged in academia because of its strict disciplinary boundary predilections. NSF has been funding crosscutting projects but perhaps is overly reliant on panels of disciplinary experts to score proposals. However, participants from different disciplines may tend to bring the same disciplinary biases referred to above that prevent proposals being seen in the light of crosscutting innovations rather than contributions to the advancement of their own disciplines.

5.5 What Message Would You Like to Send to the Funding Agencies to Encourage Innovation?

One thing to do might be to not attempt to formalize too strictly the discipline the program is funding. It seems that Systems Engineering program of NSF has been trying to require that "new systems knowledge" must be generated by the grantees. Perhaps, to keep from being a meaningless requirement the program has been trying to define what constitutes new knowledge thus formalizing the type of boundaries that provide the barriers to innovation discussed above.

Funding agencies might learn from Johnson's (2011) treatise on where good ideas come from. Johnson contends that "great innovations emerge from environments that are partly contaminated by error. Error is present in both the evolution of life and the innovation of great ideas, and it is not always a bad thing." For example, he asserts that unexplained errors force us to adopt new strategies and to abandon our old assumptions.

REFERENCES

Balci, O. 1988. "The Implementation of Four Conceptual Frameworks for Simulation Modeling in High-Level Languages". In *Proceedings of the 1988 Winter Simulation Conference*, edited by M. A. Abrams, P. L. Haigh, and J. C. Comfort, 287-295. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Balci, O., J. D. Arthur, and W. F. Ormsby. 2011. "Achieving Reusability and Composability with a Simulation Conceptual Model". *Journal of Simulation* 5(3):157-165.

Barros, F. J. 1997. "Modeling Formalisms for Dynamic Structure Systems". *Transactions on Modeling and Computer Simulation* (TOMACS) 7(4):501-515.

Billingsley, P. 1968. *Convergence of Probability Measures*. New York: John Wiley and Sons.

Bourbaki, N. 1930. Wikipedia, https://en.wikipedia.org/wiki/Nicolas_Bourbaki

Bryant, R. E. 1977. "Simulation of Packet Communication Architecture Computer Systems". M.S. Thesis, MIT-LCS-TR-188, Massachusetts Institute of Technology, Boston, MA.

Buxton, J. N. ed. 1967. *Proceedings of the IFIP Working Conference on Simulation Programming Languages*. Oslo, Norway.

Buxton, J. N., and J. G. Laski. 1969. "Control and Simulation Language". *Computer Journal* 5:194–199.

Chandy, K. M., and J. Misra. 1979. "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs". *IEEE Transactions on Software Engineering* SE-5(5):440-452.

Chow, A. 1996. "Parallel DEVS: A Parallel, Hierarchical Modular Modeling Formalism and its Distributed Simulator". *Transactions of the Society for Computer Simulation International* 13(2):55-68.

Conway, R. W. 1963. “Some Tactical Problems in Digital Simulation”. *Management Science* 10(1): 47–61.

Conway, R. W., B. M. Johnson, and W. L. Maxwell. 1959. “Some Problems of Digital Systems Simulation”. *Management Science* 6 (1): 92–110.

Crane, M. A., and A. J. Lemoine. 1977. *An Introduction to the Regenerative Method for Simulation Analysis.* New York: Springer-Verlag.

Efron, B. 1982. *The Jackknife, the Bootstrap, and Other Resampling Plans*. 38. Society of Industrial and Applied Mathematics CBMS-NSF Monographs.

Fujimoto, R. M. 2011. “On Innovation, and Building and Sustaining a Successful Career in Research”. In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. C. Fu. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fujimoto, R. M. 2016. “Research Challenges in Parallel and Distributed Simulation”. *ACM Transactions on Modeling and Computer Simulation* 24(4).

Fujimoto, R. M., R. Bagrodia, R. Bryant, K. M. Chandy, D. Jefferson, J. Misra, D. Nicol, and B. Unger. 2017. “Parallel Discrete Event Simulation: The Making of a Field”. In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Giambiasi, N., B. Escudé, and S. Ghosh. 2000. “GDEVS: A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems”. *Transactions of the Society for Computer Simulation International* 17(3):120-134.

Goldsman, D., R. E. Nance, and J. R. Wilson. 2009. “A Brief History of Simulation”. In *Proceedings of the 2009 Winter Simulation Conference,* edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 310–313. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Goldsman, D., R. E. Nance, and J. R. Wilson. 2010. “A Brief History of Simulation Revisited”. In *Proceedings of the 2010 Winter Simulation Conference,* edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 567–574. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gordon, G. 1961. “A General Purpose Systems Simulator”. In *Proceedings of the EJCC*, Washington D.C., pp. 87-104. New York: MacMillan.

Ho, Y. C. 1994. *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World*. IEEE Press.

Hollocks, B. W. 2008. “Intelligence, Innovation and Integrity – KD Tocher and the Dawn of Simulation”. *Journal of Simulation* 2(3):128–137.

Hwang, M. H. 2011. “Taxonomy of DEVS Subclasses for Standardization”. In *Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, 152-159. San Diego, CA: Society for Computer Simulation International.

Jafer, S., B. Chhaya, U. Durak, and T. Gerlach. 2016. “Formal Scenario Definition Language for Aviation: Aircraft Landing Case Study”. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, doi: 10.2514/6.2016-3521.

Jefferson, D. 1985. “Virtual Time”. *ACM Transactions on Programming Languages and Systems* 7(3): 404-425.

Jefferson, D. 2016. “Past, Present, and Future of Parallel Discrete Event Simulation – Keynote Address”. In *Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation* (SIGSIM PADS 2016), New York: Association for Computing Machinery, Inc.

Johnson, S. 2011. “Where Good Ideas Come From, About the Natural History of Innovation”. Riverhead Books, 344 pages.

Kim, T. G., C. Lee, E. R. Christensen, and B. P. Zeigler. 1990. "System Entity Structuring and Model Base Management". *IEEE Transactions on Systems, Man, and Cybernetics* 20(5):1013-1024.

Kiviat, P. J. 1963. "GASP—A General Activity Simulation Program". Applied Research Laboratory, U.S. Steel Corporation, Monroeville, PA.

Kiviat, P. J. 1967. "Digital Computer Simulation Modeling Concepts". RAND Memo RM-5378-PR, Santa Monica, CA: RAND Corporation.

Kiviat, P. J. 1969. "Digital Computer Simulation: Computer Programming Languages". Santa Monica, CA: RAND Corporation. https://www.rand.org/pubs/research_memoranda/RM5883.html

Kofman, E., and S. Junco. 2001. "Quantized-State Systems: a DEVS Approach for Continuous System Simulation". *Transactions of the Society for Computer Simulation International* 18(1):2-8.

Markowitz, H. M., B. Hausner, and H. W. Karr. 1962. "Simscript: The Simulation Programming Language". Rand Corporation Report, Rm-3310, Cambridge, MA.

Miller, D. C., and J. A. Thorpe. 1995. "SIMNET: The Advent of Simulator Networking". In *Proceedings of the IEEE* 83(8): 1114-1123.

Mital, S., and J. L. Risco Martin. 2013. *Netcentric System of Systems Engineering with DEVS Unified Process*. CRC Press.

Mosterman, P. J., and H. Vangheluwe. 2004. "Computer Automated Multi-Paradigm Modeling: An Introduction". *Simulation* 80(9):433-450.

Nance, R. E., and R. G. Sargent. 2002. "Perspectives on the Evolution of Simulation". *Operations Research* 50(1): 161–172.

Nelson, B. L. 2004. "50th Anniversary Article: Stochastic Simulation Research in Management Science". *Management Science* 50: 855–868.

NSF/NASA/AFOSR/NMSC/NTSA Workshop. 2016. *Research Challenges in Modeling and Simulation for Engineering Complex Systems*. http://trainingsystems.org/publications/Research-Challenges-in-Modeling-and-Simulation-for-Engineering-Complex-Systems.pdf.

Nutaro, J. 2014. "An Extension of the OpenModelica Compiler for Using Modelica Models in a Discrete Event Simulation". *Simulation* 90(12):1328-1345.

Nygaard, K., and O. Dahl. 1978. "The Development of the SIMULA Languages". *ACM SIGPLAN Notices* 13: 245–272.

OMG. 2015. "Documents Associated with Meta Object Facility™ (MOF™) Version 2.5". http://www.omg.org/spec/MOF/2.5/

Praehofer, H. 1991. "System Theoretic Formalisms for Combined Discrete-Continuous System Simulation". *International Journal of General Systems* 19(3):219-240.

SIGSIM. 2017. "Modeling and Simulation Areas". ACM Special Interest Group (SIG) on Simulation and Modeling (SIM) M&S Knowledge Repository. New York: Association for Computing Machinery, Inc. (ACM). http://www.acm-sigsim-mskr.org/MSAreas/msAreas.htm

Whitehead, A. N., and B. Russell. 1910. *Principia Mathematica 1* (1 ed.), Cambridge: Cambridge University Press, JFM 41.0083.02

Wymore, A. W. 1967. *Mathematical Theory of Systems Engineering: The Elements*. New York: Wiley.

Zeigler, B. P. 1976. *Theory of Modelling and Simulation*. New York: Wiley.

Zeigler, B. P. 1984. *Multifaceted Modelling and Discrete Event Simulation*. Cambridge, MA: Academic Press.

Zeigler, B. P. 1987. "Hierarchical, Modular Discrete Event Models in an Object Oriented Environment". *Simulation* 49(5):219-230.

AUTHOR BIOGRAPHIES

OSMAN BALCI is a Professor of Computer Science and Director of the Mobile / Cloud Software Engineering Lab at Virginia Polytechnic Institute and State University (Virginia Tech). He received his Ph.D. degree from Syracuse University in 1981 under the supervision of Professor Robert G. Sargent. Dr. Balci served as the founding Editor-in-Chief of two international journals: *Annals of Software Engineering*, 1993-2002 and *World Wide Web*, 1996-2000. He served as Chair of ACM SIGSIM, 2008-2010; and Director at Large of the Society for Modeling and Simulation International, 2002-2006. Most of Dr. Balci's work has been funded by the U.S. Navy. His current areas of expertise center on mobile / cloud software engineering, modeling and simulation, and digital game-based learning. His e-mail and web addresses are balci@vt.edu and http://manta.cs.vt.edu/balci.

RICHARD M. FUJIMOTO is a Regents' Professor in the School of Computational Science and Engineering at the Georgia Institute of Technology. He received a Ph.D. in Computer Science & Electrical Engineering from the University of California-Berkeley in 1983. He has been an active researcher in the parallel and distributed simulation field since 1985, and has authored or co-authored 3 books and over 250 technical papers on this subject including 7 award winning publications. He led the definition of the time management services for the High Level Architecture for modeling and simulation (IEEE standard 1516). He received the ACM Distinguished Contributions in Modeling and Simulation Award in 2013. His e-mail address is fujimoto@cc.gatech.edu.

DAVID GOLDSMAN is a Professor in the H. Milton Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research interests include simulation output analysis, ranking and selection, and healthcare simulation. He is an active participant in the Winter Simulation Conference, having been Program Chair in 1995, and having served on the WSC Board of Directors from 2001–2009. His e-mail address is sman@gatech.edu, and his Web page is www.isye.gatech.edu/~sman.

RICHARD E. NANCE is an Emeritus Professor of Computer Science at Virginia Tech. He is the author of over 150 papers on discrete event simulation, performance modeling and evaluation, computer networks, and software engineering. Dr. Nance has held several editorial positions and was the founding Editor-in-Chief of the *ACM Transactions on Modeling and Computer Simulation*, 1990–1995. He served as Program Chair for the 1990 Winter Simulation Conference. Dr. Nance has received several awards for his editorial and professional contributions, most recently the Lifetime Professional Achievement Award from the INFORMS Simulation Society. He was named an ACM Fellow (1996) and an INFORMS Fellow (2008). His e-mail address is nance@vt.edu.

BERNARD P. ZEIGLER is an Emeritus Professor of Electrical and Computer Engineering at the University of Arizona and Adjunct Research Professor in the C4I Center at George Mason University. He is internationally known for his seminal contributions in modeling and simulation theory and has published several books including "Theory of Modeling and Simulation" and "Modeling & Simulation-based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange". He was named Fellow of the IEEE for the Discrete Event System Specification (DEVS) formalism that he invented in 1976. Among numerous positions held with the Society for Modeling and Simulation International (SCS) he served as President and was inducted into its Hall of Fame. For more information see the Wikipedia entry on Bernard P. Zeigler.