Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

PANEL ON GRAND CHALLENGES FOR MODELING AND SIMULATION

Simon J. E. Taylor

Department of Information Systems
and Computing
Brunel University, Uxbridge
UB8 3PH, UK

Paul A. Fishwick

CISE Department
University of Florida
Gainesville
Florida, USA

Richard Fujimoto

School of Computational Science and Engineering
Georgia Institute of Technology
Atlanta, Georgia, 30332, USA

Adelinde M. Uhrmacher

University of Rostock
18052 Rostock
GERMANY

Ernest H. Page

The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102, USA

Gabriel Wainer

Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Dr. Ottawa, ON, CANADA

ABSTRACT

It has been a decade since the Workshop on Grand Challenge for Modeling & Simulation (M&S) was held at Dagstuhl in Germany (www.dagstuhl.de/02351). Grand challenges provide a critical focal point for research and development and can potentially create the critical mass needed to bring substantial transformation and benefit to a community. The Workshop addressed a wide variety of M&S theoretical, methodological and technological issues across many application areas. This Panel reflects on progress made since the Workshop, new Grand Challenges that have emerged over the past ten years and key M&S milestones for the next decade.

1 INTRODUCTION

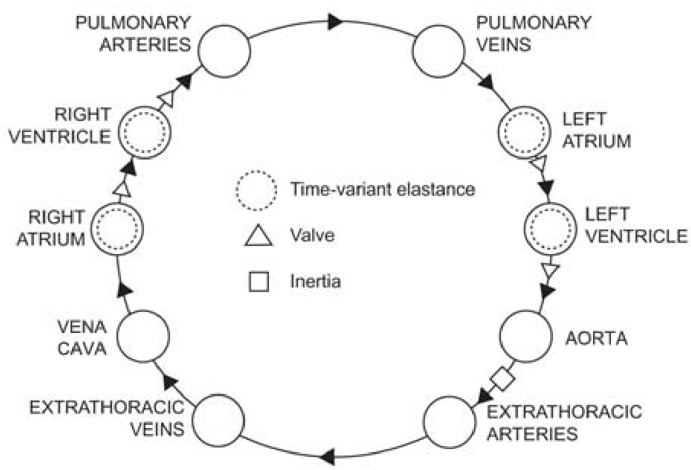
Our field of Modeling & Simulation has made many advances and has spread in use across familiar and some unfamiliar domains since the Workshop on Grand Challenge for Modeling & Simulation (M&S) was held at Dagstuhl in Germany (www.dagstuhl.de/02351) around a decade ago. The main output of the Workshop was a report that is still accessible through the Workshop weblink. To provide the opportunity to reflect on the past ten years, and to speculate for the next ten, this Panel discusses aspects of progress made since the Workshop, new Grand Challenges that are emerging and key M&S milestones for the next decade.

2 PAUL A. FISHWICK

2.1 Model Interactions

Models are language products that appear in text, in graphics, and sometimes in physical, scale form. When we consider "model interactions," we are concerned with how models interact with each other, how

978-1-4673-4782-2/12/$31.00 ©2012 IEEE



$$p(t) = E * (v(t) - UV)$$

$$f(t) = \frac{p_{in}(t) - p(t)}{R}$$

$$\frac{dv}{dt}(t) = f(t) - f_{out}(t)$$

E : elastance
f : flow
p : pressure
R : resistance
v : volume
UV : unstressed volume

Figure 1: Ten compartment human cardiovascular dynamic model with pressure and flow equations.

models connect to the real world, and how we interact with models. As simulation researchers, we have done remarkable things with models and their interactions, but our work has a long way to go—especially with regard to how models and their components interact with humans. Let's begin with three broad components that occupy our thoughts as we model: the human, the model, and physical reality. The concept of interaction in modeling, by itself, applies to ways that models and their components interconnect. For humans, the question is how to best achieve high density, and seamless, interaction with models, and often models as connected within a domain context.

Let's imagine interaction with a heart. We might want to point to it and see different schematic, illustrative, or high-fidelity graphical representations of the heart, or see blood flow, one or more dynamic models of the flow. We might begin with a real person and through a seamless type of human interaction investigate these models and representations. We want to have multiple sensory modalities from vision and touch to sound. We don't want to use ten different interfaces. We want only one interface that is capable of infinitely flexible degrees of exploration. For reference on this grand challenge vision, Figures 1 and 2 show two models. Figure 1 is a dynamic compartmental model of the human heart.

Ezzell et al. (2011) describe these model and human-computer interfaces designed to allow both a user and an instructional designer to perform their respective functions. The situation is one where we have several models, the models must interact with each other, and we must be able to interact with them. Figure 1 captures the dynamics of blood flow—typical of what we would call a "model" in simulation. Figure 2 is typical of what computer graphics, and computer aided design, researchers would call a "model." Figure 2 is a crude proxy for "physical reality" since it is what one expects to see in anatomy. In some cases, a video may be superior depending on the goal of the simulation analysis. The full interface also includes an ontology, which is a third type of model most often associated with the semantic web. Here are some questions that we must ask of all of these models:

- How do we formalize each of the models? What information structures do we employ?
- How are the model components interconnected using these formalisms?
- How can we navigate multiple abstraction levels of each of the models? The model in Figure 1 is based on ordinary differential equations, but a lower-level of detail model might employ finite element analysis (for structural questions) or fluid mechanics (for blood flow). Scales across space and time require different model structures.
- How do we connect each of these models with actual hearts?

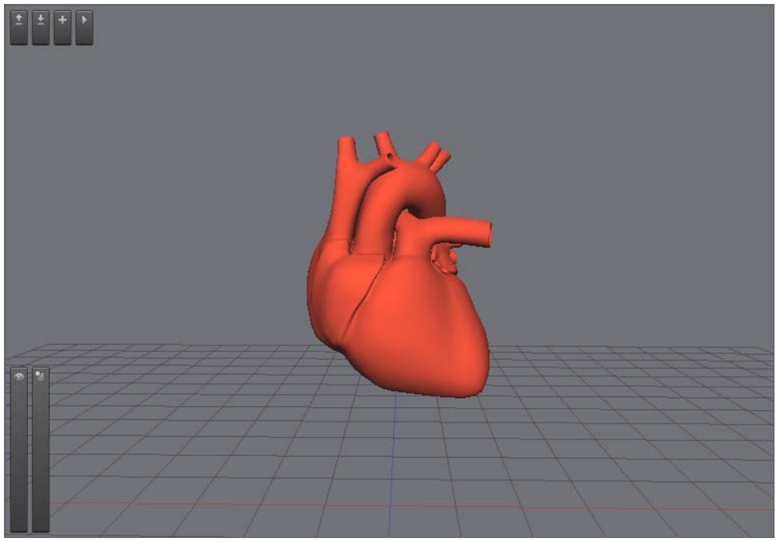


Figure 2: Simple mesh-based model of the human heart within a prototype skeletal interface.

- How many ways should we represent dynamic and geometry models? Should we use 2D schematics rather than a 3D mesh?

Even though Ezzell et al. (2011) attempt to answer some of these questions for the human heart, our community has only begun the scratch the surface for the model interconnection problem in general. Fishwick (2012) presents one framework in "hypermodelling" to capture some of the problems listed in this section. The dominant practice when multiple models are present for model authoring or analysis is to wrap each in a tool, and then integrate the tools. For example, consider connecting the output of TOOL X with the input of TOOL Y. The problem with this approach is that it breaks the human interface—we need to evolve toward a situation where the models and their computational methods are separate from the interface. Some of the semantic web standards may assist in this goal—specifically those that define ontologies (e.g, OWL, RDF).

3 RICHARD FUJIMOTO

3.1 Parallel and Distributed Simulation: Past, Present, and Future:

The parallel and distributed simulation field began in the late 1970's with seminal work that asked a key fundamental question: how can one ensure that the execution of a discrete event simulation program distributed across multiple processors will obtain exactly the same results as a sequential execution? At approximately the same time, the distributed simulation field grew out of work in the 1980's that integrated autonomous simulators over a network. Buoyed by this success, work in the 1990's focused on the development of interoperability standards and institutionalizing the distributed simulation paradigm.

In many ways, the 1980's and 1990's were the "golden age" of parallel and distributed simulation. The technology received much attention, new communities developed and flourished, and many new innovations were developed. The last decade has seen a maturation of the technology and an increased emphasis on its application. While synchronization is now a mature area of research, completely satisfactory solutions that yield both high performance and ease of use remain a challenge. While applications research has spanned many areas, simulation of large-scale telecommunication networks has been one area that has perhaps seen the most extensive activity. Advances in distributed simulation for virtual environments has focused on issues such as reducing the effort to realize interoperable simulation models and data distribution techniques to improve scalability, among others.

3.2 Parallel and Distributed Simulation Today

In 1993 an article asked a provocative question that was intended to draw community reaction: “Parallel Discrete Event Simulation: Will the Field Survive?” (Fujimoto 1993). This paper emphasized a number of challenges that stood in the way of widespread adoption of the technology, focusing primarily on issues concerning the development of parallel discrete event simulation software. Four avenues were proposed: (1) creation of application specific libraries to hide the details of parallel simulation software development from its users, (2) new parallel simulation languages to simplify the development of parallel simulation codes, (3) support for shared state, also to simplify software development and (4) development of methods to automate the parallelization of sequential simulation codes. To date, progress has been made in all four areas, but arguably, the greatest impact has been in areas (1) and (4). The application library approach is exemplified by Qualnet (Scalable Network Technologies 2012) that enables one to configure a network simulation from a graphical user interface, while hiding the details of parallel execution in an underlying simulation executive. There is little question that hiding the many complexities associated with developing and executing parallel simulations from users is essential for widespread adoption by a large user community. In the automated parallelization arena, the “self-federating” approach where multiple instantiations of a single sequential simulation are created and interconnected, with each modeling a portion of the overall system has emerged as a practical approach to realizing parallel simulations. For example, each sequential simulation might model a sub-network of a large telecommunications system. Examples of this approach include Nicol and Heidelberger 1996, Riley, et al. 2004, among others. Because this approach effectively involves federating a sequential simulation with itself in a parallel or distributed computing environment, it avoids many of the interoperability issues that arise when federating different simulators. While not all sequential simulators are amendable to this approach, experience has shown that with the use of appropriate software design practices such as the avoidance of global state variables, this can be a viable approach to parallelization.

Despite these successes, the penetration of parallel and distributed simulation technology into the broader modeling and simulation industry still has a long ways to go. Strassburger, et al. (2008) reports the findings of a peer study in the context of distributed virtual environments that indicate the technology is widely believed to have the potential for broad impact, but widespread adoption in industry has not yet materialized. To be sure, there are locations where the technology has had a sizable economic impact. As early as 2003 the modeling and simulation industry was cited to create over $1 billion in jobs in Florida (National Center for Simulation 2003) and over $500 million in the Hampton Roads area of Virginia in the (Government Technology 2005) in the U.S. Massive, multiplayer on-line game systems also represent a sizable economic impact. Notwithstanding these impressive successes, widespread adoption of the technology does not appear to have reached its fullest potential. In Taylor, et al. (2009) the authors articulate the need for new standards to help address this issue by increasing interoperability of simulation models and tools.

3.3 Future Directions

It has always been the case that advances and innovation in parallel and distributed simulation have been driven by advances in the underlying hardware and software technologies. The parallel discrete event simulation field emerged in the 1980’s largely as a result of the development of commercially available parallel computing platforms. Distributed simulation technology arose from advances in networking technology and computer graphics. Just as then, it is the case today that to see the future of parallel and distributed simulation, one need only look at today’s emerging underlying technologies. Arguably, the most relevant computing trends include:

- ***Ubiquitous computing.*** It has long been the case that by far, the largest number of manufactured computers are not those found in machine rooms or offices, but rather those embedded in devices. This trend continues as sales of personal computers give way to tablets, and both are dwarfed in volume by cell phones and other smart mobile devices. Embedded on-line simulations and cyber-physical systems offer the potential to realize self-organizing, self-optimizing distributed systems. Dynamic data driven application systems (Darema 2004) and symbiotic simulations (Aydt, et al. 2009) offer much potential, but present challenging problems to incorporate data with inherent inaccuracies and errors into the simulation in real time, execution under constraints such as limited communications bandwidth, power, and time, realizing effective on-line analyses, and effecting optimizations on operational systems, to mention a few.
- ***Supercomputing***. The exponential increases in computing performance due to increased clock speeds that had persisted over decades of computing largely came to a halt in 2005, giving way to the need to exploit massively parallel and heterogeneous supercomputers for the most computationally intense simulation problems. Effective exploitation of emerging massively parallel supercomputers for real world applications remains a very challenging problem today. Parallel simulation over SIMD computing platforms, explored to a limited extent in the 1980's and 1990's, is re-emerging as an important area with the advent of general-purpose graphical processing unit (GPGPU) computing.
- ***Cloud computing***. A longstanding problem faced by the parallel simulation community has concerned how to make the technology widely accessible to the general modeling and simulation community, given the complexities associated with developing parallel simulation code. While much progress has been made, effective exploitation requires solving a number of other issues, including nontechnical challenges such as gaining access to suitable hardware platforms and the realization of a viable economic ecosystem. Cloud computing offers an approach that can help solve many of these issues, and offers the potential to make parallel simulation widely accessible to the mainstream modeling and simulation community. However, creation of parallel simulation codes that can execute effectively on cloud computing platforms, which are typically not well suited for efficient execution of closely coupled parallel codes, is a significant obstacle. Much work remains to be done to effectively exploit this emerging capability.
- ***Big Data and Complex Adaptive Systems.*** It is widely acknowledged that in many domains, the challenge has shifted from one of obtaining sufficient data to perform credible analyses to extracting useful knowledge and insight from the data that are available. The tsunami of data that is becoming available offers the potential to revolutionize many fields. This will lead to new advances in complex adaptive systems where underlying physical laws do not exist and data are essential to discovery. Simulation of complex adaptive systems often entails modeling large systems such as interdependent infrastructures of entire cities. Parallel and distributed simulation techniques are needed. Many challenges remain. One grand challenge problem in complex adaptive systems concerns the question of what to model to capture salient properties of the system under investigation, and conversely, deciding what to leave out. This is perhaps the greatest question facing the modeling simulation field since its inception.

4 ERNEST H. PAGE

First let me thank the panel organizers, Simon Taylor and Gabriel Wainer, for their efforts to renew a community discussion of grand, quasi-grand, and interesting-even-if-not-actually-grand challenges in modeling and simulation. Secondly, has it really been ten years since the Dagstuhl Seminar? Yikes.

I'll leave it to other members of the panel, if they are so inclined, to take a rigorous forensic look at the observations and predictions we made in 2002 and suggest whether or not any significant progress is evident. Having remained engaged with U.S. governmental applications of M&S for the past decade (in both the defense and civil sectors), I concede there is a "sameness" to the M&S pursuits of today and those of ten years ago: standards, interoperability and reuse remain a predominant concern. This apparent lack of progress might be due to many factors: these issues may simply be fundamentally hard to solve, or shifting priorities for research and development funding during wartime may have delayed progress, or possibly it is even the "pet rock syndrome" -- a community of practitioners become comfortable working on a set of problems (admittedly difficult ones) even to the exclusion of other problems.

Accepting that the issues raised in 2002 are still open and interesting, and gazing once again into my crystal ball – I really should have this thing calibrated – I see a few interesting problems that could arguably be called "grand challenges".

4.1 Overcoming the Overreach

Having a vision is good, as long as it is the right vision. Pursuit of the impossible might yield tangential benefits, but it is just as likely to waste valuable resources.

4.1.1 The Model that Does Everything and Ties Everything Together

There is a well-known proverb, which is said to have been in existence in one form or another since the 14th century, that provides a compelling portrayal of the nature and consequences of causation. Its spirit and teachings are far-reaching, from parental admonitions to misbehaving children (Franklin 1758), to explanations of Chaos Theory (Hazelwood 2009; Wilson, 2011), to official opinions expressed by the U.S. Supreme Court (Roberts 2007).

For want of a nail the shoe was lost.
For want of a shoe the horse was lost.
For want of a horse the rider was lost.
For want of a rider the battle was lost.
For want of a battle the kingdom was lost.
And all for the want of a horseshoe nail.

What advice does this proverb offer to simulation modelers? It seems to say that the tiniest detail can have significant impact at the kingdom scale. So valid kingdom-scale models must be highly detailed, right? Certainly, the trend in today's large-scale military models is to adopt this approach. Detailed engineering models must feed detailed engagement models which, in turn, must feed detailed campaign models. Seemingly unknown to the advocates of this approach is that a model, by definition, omits detail. The result is very highly detailed simulations that take too long to build (are often never completed), cost too much, and cannot be maintained. Those in charge of procuring the next-generation of large-scale decision support and training simulations must appreciate and enforce the principles of abstraction and approximation in the requirements for these systems.

4.1.2 So Simple Anyone Can Use It

There was a sketch on the television show, Saturday Night Live, in the 1980s that parodied a commercial for the Canon AE1 camera. The slogan for the AE1 at the time was, "So simple, anyone can use it." Actual commercials for the camera showed the camera being used by small children, older technologically-shy people, and so forth, to take very good photographs. In the parody, the famous musician Stevie Wonder (played by himself) and the famous tennis play Rod Laver (played by Joe Piscapo) are on the tennis court. First Rod uses the camera to take pictures of Stevie as he swings and misses at tennis balls that go past him. Then Stevie takes the camera to take pictures of Rod as he plays. Stevie's pictures are mostly out of focus shots of the sky, half a face, an arm, a foot, and so on. The moral of the parody, of course, is that blind people tend take really bad photographs no matter how sophisticated the camera.

The analogy for simulation is obvious – poorly trained modelers (analysts) will build really bad models (and make really bad decisions) no matter how sophisticated the modeling tools.

4.2 Identifying the Essential and Accidental Complexities of M&S

As a community of researchers and practitioners, we should rigorously consider how Fred Brooks' elegant analysis of Software Engineering (Brooks 1987) applies to our work. Are we spending most of our research addressing the *accidental* complexities of M&S or the *essential* ones? Where do our efforts in topics like composability, web-enabled simulation, and agent-based simulation, for example, fall in Brooks' scheme?

4.3 Fostering an Environment for Ubiquitous Simulation

The notion of *ubiquitous simulation* has been around for a long time, and was a prominent theme at Dagstuhl. With the re-emergence of the utility computing, this time in the form of "the cloud", there is new opportunity to realize the visions associated with this topic. For example:

- *Embedded Simulation for Real-Time Decision Support.* The availability of inexpensive sensing, coupled with solutions to the storage and retrieval of very large volumes of data presents an opportunity to develop highly-instrumented systems, for example in factory management and air traffic control. This, in turn, gives us the opportunity to develop real-time simulations which can be embedded within the systems, continually calibrated using the instrumentation data, and used in a real-time decision support role.
- *Simulation On Demand.* Many of today's analytical simulations have very long shelf lives – they get applied to dozens, even hundreds of analytical studies. Many of these simulations also take a very long time to run. Arguably, the omission of unnecessary detail in these models, as called for above, would help remediate this. Metamodeling provides a means to remove model detail – even if the original modeler would not! The availability of cheap, elastic computing in the cloud gives us the opportunity to run our long-running simulations continuously as "background" processes. These runs could serve to both conduct detailed exploration of the model's response surface, and to generate metamodels, which could be used to shorten the analysis cycle when a new question is asked of the model.
- *Simulation Based Acquisition.* A concept from the 1990s that perhaps suffered due to high development costs and program failures associated with the "mega-simulations" of the same period. It remains, though, a perfectly reasonable notion. At every stage of system acquisition, simulation models should be constructed and employed to support the activities associated with that stage. The models themselves represent the requirements and become part of the delivered system. Even though it contains infinitely many assumptions, an executable model is far less ambiguous than a text-based requirements document.

5 SIMON J. E. TAYLOR

5.1 Simulation Interoperability Standards

The creation of simulation models consisting of many different components using simulation software environments and packages has been possible for some time. These components range from simple activities, entities and resources to complex conveyor systems and workers with complex shift patterns and availabilities. However, these packages are different and these components cannot be simply transferred without recoding from scratch. Arguably, we still cannot transfer components written in the same software between models without re-verification. Similarly, if one is to compose simulations from different interoperating models, despite advances in underlying distributed simulation technology, there are still major barriers to what is conceptually a simple task. Outside of military applications there have been some advances in standardizing this interoperability, particularly with the identification of general process model-level interoperability problems and manufacturing data standards (Strassburger and Taylor 2012). However, without the development of more simulation standardization, models will continue to be confined by the software that they have been developed in.

5.2 Grid or Cloud Computing for High Speed Experimentation and Optimization

In astronomy, physics and biology as well as some areas of engineering and finance, the computational demands of software are supported by Grid or Cloud Computing. The use of many real or virtually mapped processors speeds up the execution of software by the use of familiar task-level parallelism over manager-worker architectures. Most areas of modeling and simulation are exceptions to this. The provision of easy to implement and/or access Grid or Cloud Computing facilities will have a significant benefit to simulation practitioners and researchers by enabling results to be obtained faster with (if required) significantly more accuracy (Taylor et al. 2011). The challenge is how to create this in a realistic organizational setting with contemporary simulation software and also how to support simulation interoperability (a unique challenge to contemporary Grid or Cloud Computing).

5.3 Web Simulation Science

One of the most "challenging" Grand Challenge is *the scope, complexity, size and quality of M&S projects being limited by the methods and tools that we have today.* New areas of study are needed in M&S. Taking a simple view of M&S, in manufacturing, for example, models are created using a COTS Simulation Package. Models are populated with data from a variety of sources: sometimes models are linked to external data sources, sometimes information from CAD systems is used, sometimes input is taken from real-time systems, sometimes expert qualitative knowledge is used, etc. Some form of iterative development takes place involving verification and validation against the system under study. The model is experimented with and results are obtained. These are used to make decisions and the system being studied is changed to reflect the new insight gained from experimenting with the model. All this is well known. However, as the scope of these models grow and project constraints become tighter (time, money), it becomes more and more difficult to properly perform M&S as the *complexity* of the model grows. Software assists in the management of complexity to some extent but also hinders this management, especially when multiple software is used (simulation packages, spreadsheets, databases, etc.) Modelers often develop their own techniques to cope with this and, unavoidably, create their own artifacts that need to be managed in their own right. Trying to manage models, their artifacts and even the people involved in them with the current tools at our disposal arguably limit the scope and quality of M&S. This problem is exacerbated in less structured application areas such as healthcare.

The World Wide Web is a complex environment that has revolutionized many aspects of modern life. In turn, the *Semantic Web* is revolutionizing the World Wide Web. Key to this is the study and use of ontologies to organize and reason about complex information structures. Some new and some familiar fields have come with this: ontology engineering, ontology alignment, ontology mapping, discovery, composition, and interoperability. This is allowing us to consider the Web as a complex ecosystem of artifacts (data, programs, people, etc.) The field of *Web Science* studies this and its emergent complexity (Hendler et al. 2008). As argued in Taylor (2011), the Semantic Web may have a significant effect on the methods and tools of Modeling and Simulation (M&S), particularly in terms of model ontologies, discovery, composition, interoperability and reuse. Already new areas are emerging. For example, *Hypermodeling*, the general theory and practice of linking system models and their components, has been proposed by Fishwick (Fishwick 2012).

In the same manner that *Web Science* is starting to study the complexity of the Web, the Grand Challenge that I therefore put forward is the development of *Web Simulation Science* (WSS) or *Modeling & Simulation Ecosystems* as novel fields of study to help in creating the theories, methods and technologies needed for the successful realization of large scale simulations and M&S projects. Note that I use "Web" here in a similar sense of how "Web" is used in Web Science as it studies the vast socio-technological "ecosystems" of the Web as an entity in its own right. I therefore generalize the definition of WSS presented in Taylor (2011) as *the study of M&S ecosystems of artifacts*, e.g. the relationships between the elements of (large) M&S projects. This is strongly associated with developments in the Semantic Web and Hypermodelling. An M&S ecosystem is 'everything'; the emergent complexity of a web of people, organizations and society with regard to M&S artifacts (models, tools, technology, data, etc.). The Semantic Web gives potentially an abstract foundation through the use of ontologies and associated languages to form the structure of an M&S ecosystem on to which actual model elements, data and their relationships can be mapped and studied. Hypermodelling subsumes and extends some approaches to connect and integrate simulation models (Fishwick 2012). Hypermodelling includes issues of model linking, interoperability (to models, system models in other forms, media, heterogeneous model types and custom representations of the same model) and, to some extent, reusability. The development of WSS techniques that assist in the management and organization of large-scale M&S projects through the seamless integration of artifacts and their associated world views and technologies can provide a foundation for a new revolution in M&S.

Some key research questions might be:

- Can generalizable M&S ontologies be created that fully support discrete-event simulation, agent-based simulation, systems dynamics and hybrid simulation?
- Can these be made domain specific (manufacturing, defense, healthcare, etc.) and take into account the specific and subtle semantics of commercial simulation software and stakeholder domains?
- Is it possible to create tools/techniques that use these ontologies in a manner useful for M&S (verification, validation, management, etc.)? How can these be exploited for commercial use?
- Can the underlying relationships of all the artifacts in an M&S project be easily extracted and mapped to ontology?
- What aspects of ontological research are needed to support interoperability and reuse?
- Can this approach yield insight into M&S projects that would otherwise be inconvenient or impossible?
- How can the multidisciplinary teams needed for Web Simulation Science be assembled, inspired and sustained? How can computer scientists/technologists, domain experts and end users interact meaningfully to create strong theoretical underpinnings with clear links to practice?
- Can simulation ontologies be standardized and, if so, what standardization agency should be used?

Progress towards some of these questions is already being made in some domains (particularly the military). However, there is significant interesting work still to be done that can really revolutionize M&S and bring in new approaches to creating larger, high quality models within real world project constraints.

6 ADELINDE UHRMACHER

6.1 Progress in (spatial) Modeling and Simulation

Space adds to the complexity of problems which modeling and simulation methods have to deal with, and, correspondingly, aggravates some of the problems modeling and simulation research is facing. Thus, although we will focus on spatial simulation in cell biology, the arguments apply to other application areas of modeling and simulation as well, i.e. the need for new modeling and simulation methods but also the similarly large challenge of providing reliable means for evaluating new developments.

6.2 Modeling Languages – Syntax and Semantics Matter!

Spatial aspects in modeling dynamic systems typically refer to locating the interesting entities relative to each other, e.g. by nesting, or by assigning attributes to entities that refer to the coordinates in 1D, 2D, or 3D space. Formalisms like the ambient calculus and its variant Bioambients allow nesting, moving, and merging processes (Regev, et al. 2004).The index of cellular automata describes implicitly the position of the individual automata (Deutsch and Dormann 2004). Similarly, the position of individual models is determined in cellular Devs (Wainer 2001). Processes in process algebra can be attributed with spatial coordinates, e.g. (John et al. 2010), as can be tokens in colored Petri nets (Gao et al. 2011). Rule-based approaches support to combine spatial dynamics of entities relative to each other in form of dynamic nestings and dynamics referring to spatial coordinates, e.g. (Maus et al. 2011), and even consider continuous movement and excluded volume effects, e.g. (Bittig et al. 2011).

Obviously, the development of succinct, sufficiently expressive languages for spatial modeling and simulation is a challenge, the more so the more different spatial phenomena shall be considered. However, how can the impact of one of these languages be evaluated? A first step is to assess the expressiveness by case studies and by relating the language formally to existing ones. However, how to assess the usability of a newly developed modeling language? The effort of learning the language, creating a model, and interpreting a model, requires entirely different evaluation methods. Carefully designed and evaluated empirical studies are needed for analyzing the implied cognitive effort. So far only few have attempted to execute such user studies in modeling and simulation, e.g., (Tako and Robinson 2010).

6.3 Yet another Simulation Algorithm – or What Difference does it really make?

The last decades brought forth a multitude of spatial simulation algorithms for cell biological systems. To simulate reaction and diffusion systems some may trade accuracy for execution speed by introducing additional approximations, and combine different algorithms e.g., (Ferm et al. 2010). The computational effort becomes worse if individual molecules moving continuously in space shall be simulated. However, certain spatial phenomena like molecular crowding can only be analyzed at this higher level of detail (Takahashi et al. 2010). To balance effort and detail, multi-algorithm approaches combine reaction diffusion systems and individual entities moving continuous in space. Parallelization of computation tasks is another means to increase the efficiency, i.e., (Himmelspach et al. 2010). New architectures, like the GPU, are exploited for simulating reaction diffusion system (Dematte and Prandi 2010).

This portfolio of available methods will increase, the more dynamics become of interest that operate at widely disparate scales in time and space. However, how do we evaluate the impact of new simulation algorithms and select among those the most adequate one for the task at hand, and combine them in a suitable manner, so that crucial model parts are still executed in sufficient detail? How can the user be supported in accessing the implications of trading accuracy for speed? How can the process of evaluation be standardized and automated? Adopting machine learning methods might be a crucial step into this direction (Jeschke et al. 2011) but more need to follow.

6.4 In-silico Experiments - Lost in Data, intricate Processes, and unconfirmable Statements

Since the announcement of the crises of simulation (Pawlikowski 2001) it is clear, that a more thorough support for the modeling and simulation life cycle incl. guidance and documentation is required. The exploitation of workflows (Rybacki et al. 2012) and the integration of machine learning methods may provide part of the answer. In addition, new visualization and interaction techniques are required as the usual batch method – run multiple experiments and then mine the output visually for clues will not suffice for analyzing complex cell biological dynamics like intra- and intercellular dynamics in space and in time (Mazza et al. 2010). A combination and close linkage of multiple views, each revealing different aspects in the data, will help handling models, experiments, and multi-run simulation data (Unger 2009). However, adequate visual support to steer experiments interactively and exploit effectively the users' intuition is an open research question (Keim et al. 2010). So how do we support experimenters best and if we have developed new methods, how do we evaluate their impact on concrete simulation studies?

7 GABRIEL WAINER

7.1 Ubiquitous Simulation in the Cloud: Simulation Everywhere

Modeling and Simulation (M&S) has enabled the discovery of knowledge with unprecedented level of detail; it is now widely accepted that scientific analysis is founded on three pillars: theory, experimentation, and simulation. The computational demands imposed by this model of research are continually pushing the envelope of the available technologies, as many sectors have growing needs to process, visualize, make readable, understand, and deploy complex models that use immense amounts of data. These players need to transform data into hypothesis building and critical decision-making, and to change their models in response to new hypotheses, usually involving multiple highly specialized experts working together in geographically distant areas. The Grid and Cloud computing paradigms introduced new ways of sharing computing power and storage in heterogeneous environments (Vanmechelen et al. 2012; Page et al. 2012; Ribault and Wainer 2012). Resources are virtualized as services consumed on demand (with minimal limitation for resource location).

Web-based and distributed simulation techniques have become popular to conduct these online simulations (Taylor 2011). Although Distributed Simulation middleware is now widely use for Defense applications (Page et al. 2012), its practical use in non-military applications is still very limited. As indicated by recent surveys (Strassburger, Schulze and Fujimoto 2008), the lack of practical plug-and-play interoperability makes most simulation middleware are complex to interoperate, and their composition scalability is limited. In particular, it is very complex to interface simulations with other web services. In particular, many advanced implementations are now built on SOAP Web Services to communicate (Wainer and Al-Zoubi 2010). Nevertheless, building SOA-based simulations is still complex, as the services usually address the interoperability of simulation engines at a low level of abstraction (Wainer et al. 2010; Wainer and Al-Zoubi 2011). The Grid and Cloud models of computation have some limitations. In most cases, jobs are processed off-line (which is not adequate when the players require on-line interactivity) and simulation interoperability on the Grid requires complex ad-hoc tailoring, which is a costly and lengthy process usually requiring defining complex workflows (Yu and Thain 2012; Örstberg et al. 2012).

Most existing M&S software runs on workstations, clusters or thin clients; nevertheless, there has been little work done with distributed simulation on Smartphones and handheld devices. These mobile devices have become popular, assisting their users by performing a myriad of personal tasks. They are equipped with advanced hardware (camera, GPS, compass, gyros and accelerometers), which makes them attractive for distributed simulation experiments (which would allow users to have experimentation software in the tip of their fingers). This has the potential to provide the same simulation results to geographically isolated teams working together.

We thus suggest that, although there have been numerous advances in this field, the following questions need to be addressed:

1. How to interface simulation software with Smartphone APIs? How to deal with the inherent performance issues of these devices? (power consumption, CPU speed, communication latency)
2. How to deal with power and communication disruptions?
3. How to enable multi-user collaboration between numerous users participating in a joint experiment?
4. How to integrate different online services and real-time data available in the Cloud?
5. How to include advanced algorithms and methods for combining discrete-event simulation, cloud computing (with web services interfaces) and mobile devices for distributed simulation and collaboration?
6. How to build advanced mashup applications using simulation, sharing and reusing models and experiments?
7. How to use a more abstract approach to deal with these problems? (i.e., instead of dealing with the data and simulation levels, interoperability will be dealt with at the modeling and experimentation levels, improving reuse and providing better ways to mashup models, experiments and other services).
8. How to handle the large amounts of simulation data through instrumentation of scenarios, aggregation policies and dynamic adaptation of the simulation to varying computing conditions based on different policies?

We have started some research in this field, with application in environmental sciences, in particular for forest fire simulation (Wainer and Castro 2010; Wainer and Liu 2009). In (Harzallah et al 2008, Zapatero et al. 2011) we showed how to mash up such simulations with Geographical Information Systems as input/output (see http://youtu.be/PRfNDeBUPFs and http://youtu.be/smeKecxEJwY). These mashups combine a simulation with a Global Weather service, whose real-time data is fed into a theoretical forest fire model to obtain precise results on a Google Map. Originally, we used a two-tier architecture consisting of a SOA-based simulator (Wainer et al. 2008), a Java Servlet and a Javascript client, different XML-based technologies such as XHTML, SOAP, XPATH and XSLT to enhance performance, interoperability and usability. Although this experiment was successful, the development required a considerable implementation effort (approximately 2160 person/hours), mostly due to the utilization of a SOA-based solution. Instead, we have had success with a new RESTful Web Services interface, called RISE (Restful Interoperability Simulation Environment), which can solve these issues by imitating the Web interoperability style (Ribault and Wainer 2012; Wainer and Al Zoubi 2011 Al Zoubi and Wainer 2009). The Representational State Transfer style can help solving the interoperability limitations, and it makes easy the development of mashups, which can be developed in a shorter period of time (for instance, the last clients we developed, which will be adapted for this research project took only 375 person/hours for the same simulation environment discussed above). The following figure shows a mashup application (Zapatero et al. 2011) and a smartphone client (Mancini et al. 2012).

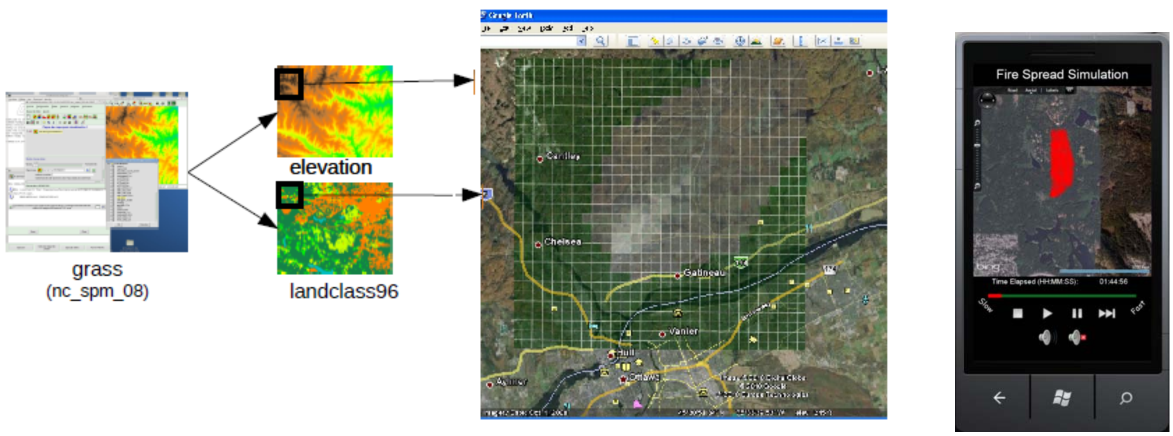


Figure 3: (a) Mashup application: forest fires simulation based on GIS; (b) Client Interface on Smartphones.

One issue is that most collaborative systems for M&S only focus on sharing *data* (in varied formats) and *processing power*. Instead, formal M&S techniques like DEVS (Discrete Events Systems specification), promise better success by addressing these issues at a higher level of abstraction (Zeigler et al. 2000). With DEVS, models, simulators and experiments are systematically built and interoperability can be enhanced. We need to explore new architectures, including hybrid approaches suitable for handheld devices. We also need to explore new mechanisms to allow users to collaborate on joint simulation exercises, sharing information and data between existing devices and the simulation server.

We argue that the combination of simulation with RESTful WS will improve the definition of very advanced applications in this field. It is important to note that the availability of RISE makes the implementation much simpler, errors are easier to find (due to the well defined methods in the interface), and the development of a mashup application like the client presented here can be done in a more effective way. Moreover, exposing a simulator as a service (Software as a Service layer in a Cloud infrastructure), is important from an integration point of view.

8 CONCLUSIONS

This paper has presented Grand Challenges in M&S from the perspective of six researchers. We hope that the views presented are stimulating and present interesting research themes for the next decade.

REFERENCES

Aydt, H., S. J. Turner, W. Cai, and M.Y.H. Low. 2009. "Research Issues in Symbiotic Simulation." In *Proceedings of the 2009 Winter simulation Conference*. 1213-1222. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Al-Zoubi, K., G. Wainer. 2009. "Using REST Web-Services Architecture for Distributed Simulation." In *Proceedings of the 23rd Workshop on Principles of Advanced and Distributed Simulation (PADS)*, Lake Placid, NY. 114-121. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bittig, A. T., F. Haack, C. Maus, and A. M. Uhrmacher. 2011. "Adapting Rule-based Model Descriptions for Simulating in Continuous and Hybrid Space". In *Proceedings of CMSB '11*. ACM Press, N.Y.

Brooks, Jr., F. P. 1986. "No Silver Bullet--Essence and Accidents of Software Engineering." *Computer* **20**(4):10-19, April 1987. Reprinted from *Proc. IFIP Congress,* Dublin, Ireland, 1986.

Darema, F. 2004. "Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements." In *Proceedings of the International Conference on Computational Science*. 662-669.

Dematte, L., and D. Prandi. 2010. "GPU computing for systems biology." *Brief. Bioinf.* 11(323): 323-333.

Deutsch, A., and S. Dormann. 2004. *Cellular Automaton Modeling of Biological Pattern Formation.* Birkhäuser Boston.

Ezzell, Z., P. A. Fishwick, B. Lok, S. Lampotang, and A. Pitkin. 2011. "An Ontology-Enabled User Interface for Simulation Model Construction and Visualization." *Journal of Simulation*, 5 (3), 147-156.

Ferm, L., A. Hellander, and P. Lötstedt. 2010, "An adaptive algorithm for simulation of stochastic reaction-diffusion processes". *J. Comput*. Phys. 229 (2): 343–360.

Fishwick, P.A. 2012. "Hypermodeling: An Integrated Approach to Dynamic System Modeling." *Journal of Simulation*, 6, 2-8.

Franklin, B. 1758. Poor richard's almanac. The Complete Poor Richard Almanacs, facsimile ed., vol. 2.

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

Gao, Q., F. Liu, D. Gilbert, M. Heiner, and D. Tree. 2011. "A Multiscale Approach to Modelling Planar Cell Polarity in Drosophila Wing using Hierarchically Coloured Petri Nets". In *Proceeings of CMSB'2011*, 209–218.

Government Technology (2005). Virginia Governor Warner Proposes $27 Million to Further Promote Modeling and Simulation.

Harzallah, Y., V. Michel, Q. Liu and G. Wainer. 2008. "Distributed Simulation and Web Map Mash-Up for Forest Fire Spread." In *Proceedings of IEEE ICWS*. Honolulu, HI.

Hazelwood, B. 2009. "For Want of a Nail". Accessed February 23 2011. http://www.freshbusinessthinking.com/articles print.php?CID=2&AID=2347.

Hendler, J., N. Shadbolt, W. Hall, T. Berners-Lee and D. Weitzner. 2008. "Web Science: An Interdisciplinary Approach to Understanding the Web." *Communications of the ACM.* 51(7) 60-69.

Himmelspach, J., R. Ewald, S. Leye, and A. M. Uhrmacher. 2010. "Enhancing the Scalability of Simulations by Embracing Multiple Levels of Parallelization". In *Proceedings of HiBi'2010*, 57–66, IEEE CPS.

Jeschke, M., R. Ewald, and A. M. Uhrmacher. 2011. "Exploring the performance of spatial stochastic simulation algorithms". *J. of Comput. Phys.* 230 (7): 2562–2574.

John, M., C. Lhoussaine, J. Niehren, and A. Uhrmacher. 2010. "The Attributed Pi-Calculus with Priorities". In *Transactions on Computational Systems Biology XII*, LNCS-5945*, 13–76*. Springer.

Keim, D., J. Kohlhammer, G. Ellis, and F. Mansmann. (Eds.) 2010. Mastering the Information Age: *Solving Problems with Visual Analytics*. EuroGraphics Association.

Mancini, E., G. Wainer, K. Al-Zoubi and O. Dalle. 2012. "Simulation in the Cloud Using Handheld Devices". In *Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. Ottawa, ON, Canada.

Maus, C., S. Rybacki, and A. M. Uhrmacher. 2011. "Rule-based multi-level modeling of cell biological systems". *BMC Systems Biology* 5: 166.

Mazza, T., G. Iaccarino, and C. Priami. 2010. "Snazer: the simulations and networks analyzer". *BMC Systems Biolog*. 4: 1.

National Center for Simulation. 2003. Impact of Florida's Modeling, Simulation and Training Industry.

Nicol, D. M. and P. Heidelberger. 1996. "Parallel Execution for Sequential Simulators." *ACM Transactions on Modeling and Computer Simulation*, 6 (3): 210-242.

Östberg, P-O., A. Hellander, B. Drawert, E. Elmroth, S. Holmgren and L. Petzold. 2012. "Reducing Complexity in Management of eScience Computations". In *Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. Ottawa, ON, Canada.

Page, E., L. Litwin, M. McMahon, B. Wickham, M. Shadid and E. Chang. 2012. "Goal-Directed Grid-Enabled Computing for Legacy Simulations". In *Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. Ottawa, ON, Canada.

Pawlikowski, K. H.-D., Jeong, and J.-S. Lee. 2002. "On credibility of simulation studies of telecommunication networks". *Communications Magazine*, 40: 132-139.

Regev, A., E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. 2004. "BioAmbients: an abstraction for biological compartments". *Theoretical Computer Science* 325 (1): 141–167.

Ribault J. and G. Wainer. 2012. "Simulation Processes In The Cloud for Emergency Planning". *In Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. Ottawa, ON, Canada.

Roberts, J. G. 2007. "Massachusetts v. Environmental Protection Agency". 549 U.S. 497 (dissenting opinion). Accessed February 23 2011. http://www.law.cornell.edu/supct/html/05-1120.ZD.html.

Rybacki, S., J. Himmelspach, F. Haack, and A. Uhrmacher. 2011. "WORMS- A Framework to support workflows in M&S". In *Proceedings of the 2011 Winter Simulation Conference*. 716-727. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Scalable Network Technologies. 2012. "QualNet." from http://www.scalable-networks.com/content/products/qualnet.

Strassburger, S. and S.J.E. Taylor. 2012. "A Comparison of CSPI and CMSD Standards." In *Proceedings of the Spring Simulation Interoperability Workshop*. 12S-SIW-010. SISO, Orlando, Fl.

Strassburger, S., T. Schulze, and R. Fujimoto. 2008. "Future Trends in Distributed Simulation and Distributed Virtual Environments – Results of a Peer Study." In *Proceedings of the 2008 Winter Simulation Conference.* 777-785. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Takahashi, K., S. Tanase-Nicola, and P. R. ten Wolde. 2010. "Spatio-temporal correlations can drastically change the response of a MAPK pathway". *PNAS* 107 (6): 2473–2478.

Tako, A. A., and S. Robinson. 2010. "Model development in discrete-event simulation and system dynamics: An empirical study of expert modellers". *Eur. J. of Operational Research* 207:784–794.

Taylor, S.J.E. 2011. Challenges for Web Simulation Science. In *Proceedings of the 2011 Winter Simulation Conference*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. pp. 2904-2908.

Taylor, S.J.E., M. Ghorbani, T. Kiss, D. Farkas, N. Mustafee, S. Kite, S.J. Turner, and S. Strassburger. 2011. "Distributed Computing and Modeling & Simulation: Speeding Up Simulations and Creating Large Models." In *Proceedings of the 2011 Winter Simulation Conference*. 161-175. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Taylor, S. J. E., N. Mustafee, S.J. Turner, K. Pan, and S. Strassburger. 2009. "Commercial-Off-The-Shelf Simulation Package Interoperability: Issues and Futures." In *Proceedings of the 2009 Winter Simulation Conference*. 203-215. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Taylor, S.J.E., T. Eldabi, G. Riley, R.J. Paul, and M. Pidd. 2009. Simulation modelling is 50! Do we need a reality check? *Journal of the Operational Research Society*, 60(1) S69-82.

Unger, A., S. H. 2009. "Visual Support for the Understanding of Simulation Processes." In *Pacific Visualization Symposium*, 57–64: IEEE.

Vanmechelen, K., S. De Munck, and J. Broeckhove. 2012. "Conservative Distributed Discrete Event Simulation on Amazon EC2". In *Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. Ottawa, ON, Canada.

Wainer, G. and K. Al-Zoubi. 2011. "Distributed Simulation Using RESTful Interoperability Simulation Environment (RISE) Middleware". In *Handbook of Intelligence-based Systems Engineering*. A. Tolk Ed. Springer-Verlag.

Wainer, G. and Q. Liu. 2009. "Tools for Graphical Specification and Visualization of DEVS Models". *Simulation, Transactions of the SCS*. 85 (3): 131-158.

Wainer, G. and R. Castro. 2010. "A survey on the application of the Cell-DEVS formalism in cellular models." *Journal of Cellular Automata*. 5:6. pp. 509-524.

Wainer, G., K. Al-Zoubi and R. Madhoun. 2008. "Distributed simulation of DEVS Cell-DEVS models in CD++ using Web-Services". *Simulation Modeling, Practice and Theory*. 16 (9): 1266–1292.

Wilson, D. G. 2011. "For Want of a Nail - A Proverb of Chaos". Accessed February 23 2011. http://fitforrandomness.wordpress.com/2011/01/31/for-want-of-a-nail-a-proverb-of-chaos-via-catagenesis/.

Yu, L. and D. Thain. 2012. "Resource Management for Elastic Cloud Workflows". In *Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. Ottawa, ON, Canada. 2012.

Zapatero, M., R. Castro, G. Wainer and M. Hussein. 2011. "Architecture for Integrated Modeling, Simulation Visualization of Environmental Systems GIS Cell-DEVS". In *Proceedings of the 2011 Winter Simulation Conference*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zeigler, B. P., H. Praehofer, T. G. Kim. 2000. *Theory of Modeling and Simulation*. Academic Press.

AUTHOR BIOGRAPHIES

PAUL A. FISHWICK (Ph.D., University of Pennsylvania) is Professor of Computer and Information Science and Engineering at the University of Florida.

RICHARD FUJIMOTO is a Regents' Professor and the Chair of the School of Computational Science and Engineering at the Georgia Institute of Technology.

ERNEST H. PAGE is a member of the technical staff of The MITRE Corporation. He was one of the organizers for the Dagstuhl Seminar on Grand Challenges for M&S in 2002. His email address is epage@mitre.org.

SIMON J. E. TAYLOR is a Reader in the Department of Information Systems and Computing at Brunel University, UK, and leads the ICT Innovation Group. His email address is simon.taylor@brunel.ac.uk.

ADELINDE M. UHRMACHER is a Professor at the Institute of Computer Science at the University of Rostock and head of the research group Modeling and Simulation.

GABRIEL A. WAINER (SMSCS, SMIEEE) is a Full Professor at the Department of Systems and Computer Engineering at Carleton University. His email address is gwainer@sce.carleton.ca.