Beyond Speedup: PADS, the HLA and Web-Based Simulation

Ernest H. Page
The MITRE Corporation
1820 Dolley Madison Boulevard
McLean, VA 22102-3481
epage@mitre.org

Abstract

This year's installment of the PADS "community assessment" looks at previous assessments and considers some current directions in distributed simulation and their possible relationship to PADS.

1. Introduction

Welcome to this year's edition of PADS Gestalt therapy—the opportunity to look around and pick out all the things that are wrong with the PADS community. I know it is a rite that many in the community anticipate in much the same way they do those other notable inevitabilities, death and taxes. So I am honored that Stephen Turner invited me to give this Keynote Address—at the very least it gives me license to step outside of accepted technical writing dogma and write in first person for a change!

As I understand it, the current model for the workshop is to identify Keynote addressors as *insiders* or *infiltrators*. I have been invited as an infiltrator, although I'm not sure that's the category I would place myself in. Having twice attended the workshop, once published within it, acted as a referee for six workshops, and as a member of the executive committee for the ACM Special Interest Group on Simulation for the past four years having had a keen interest in the profitability of the workshop, I don't feel so much like an outsider. On the other hand, I've never published a paper whose focus was speedup.[1] So I suppose I fit somewhere between infiltrator and insider. Certainly, I am not part of the core PADS community. As a researcher and practitioner in distributed simulation, though, I have a longstanding interest in the PADS community and an interest in making contributions within that community. I suppose that with respect to this Keynote, though, *my* perception of my relationship to the PADS community is not the important thing. How the PADS community views me—and those like me—is more the matter at hand.

2. A navel well-contemplated

The PADS community has done yeoman's work in the area of critical self-assessment. A good example of this introspection is a series of articles within the Summer 1993 issue of the *ORSA Journal on Computing* [1, 2, 9, 10, 14, 24, 28].

2.1. Will the field survive?

In the lead article of the *ORSA JOC* series, Richard Fujimoto asked the question, "Parallel discrete event simulation: will the field survive?" The genesis of the question was an observed (or perceived) lack of impact of parallel discrete event simulation (PDES) techniques within the broader simulation community. Fujimoto suggested that PDES techniques were too difficult to apply and too inscrutable to the non-parallel programmer. If a sequential program could be developed in a few hours and then run in a few hours more, why spend a day (or more) developing a parallel simulation that ran in a few minutes? He believed, however, that eventually performance demands would necessitate the adoption of PDES techniques by the simulation community at large. He suggested that the future of simulation as a technique might, in fact, hinge on the success of PDES, since PDES offered the best hope to modeling large, complex systems. Realizing the value of meeting the broader community half way, though, Fujimoto proposed several areas in which PDES could make inroads with respect to usability, including: (1) application specific libraries, (2) parallel simulation languages, (3) support for shared state, and (4) automatic parallelization.

The responses to Fujimoto's observations were somewhat mixed. Reynolds suggested that any focus on usability, while perhaps well-intentioned, might serve only as a distraction to the PDES research community [24]. Reynolds

---

[1] Arguably, of course, were I a better parallel programmer I would have!

asserted that the key issue to acceptance of parallel simulation was one of performance. If you make it fast enough, they will come.

In their response, Brian Unger and John Cleary echoed Fujimoto's sentiments calling for an investment in tools, specifically, tools that support the incremental performance improvement of parallel simulations [28, p. 243]. Rajive Bagrodia also concurred with Fujimoto's observations and stressed his belief in the central role that parallel simulation languages must play in the widespread adoption of PDES techniques [2]. Jason Lin discussed the importance of performance predictors and the need for parallel simulation engines of the future to be modularized and tailorable [14]. Having recently joined the faculty at Virginia Tech, Marc Abrams was motivated to observe what he referred to as a "culture clash" between parallel simulationists and modeling methodologists [1].

Dick Nance and I elaborated on Abrams' observations in a paper given at PADS '94 [22]. We argued that the PDES literature of the day seemed to exhibit a focus on speedup to the exclusion of other factors important to the modeling process. Rarely were modeling objectives stipulated. Rarely were performance results described in terms of a complete simulation study. Sometimes models were changed to enable efficient execution. Since simulations are simply abstractions of a system, changing the abstraction is not necessarily unreasonable. But without a consideration of modeling objectives, there is no way to determine whether changes in the model are acceptable or not. These factors, we argued, caused the general simulation community to look at PDES with a degree of skepticism. In addition, we believed, despite the attraction of parallel computers to computer scientists (ourselves included) the general modeling populace was unlikely ever to develop an interest in them.

Certainly, it is interesting to look back and see how much has—and has not—changed in the past 5 or 6 years. I'll leave it as an exercise to the reader (listener) to determine the degree to which any of us involved in the discussion got it right.

2.2. Who cares?

In a 1996 PADS Keynote address [11], Fujimoto revisited many of the themes from the *ORSA JOC* articles. In 1997, it was David Nicol's turn to address the issue [17]. Nicol asked, "Parallel Discrete Event Simulation: Who Cares?" He noted that engineers find the PADS community unconvincing since many of the problems studied in PDES can be adequately solved using sequential processors, and the applicability of PDES solutions to large "industrial sized" applications is not completely evident. Nicol also asserted that with the exception of communications networks, the applications studied in the PDES domain are not highly relevant—in Nicol-speak, "where we tend to scratch doesn't itch."

Nicol contended that scientists find PDES research unconvincing due to haphazard and flawed experimental design. Modelers find PDES too complicated. He observed that performance is a *constraint,* not an objective function and that factors such as model reusability and maintenance, model development, analysis of results, and interoperability are often more important than performance. I'm surprised he was ever invited back to PADS.

Adopting a Total Quality Management (TQM) approach, Nicol suggested that PDES should be assessed in terms of its customer base and their needs. He identified five classes of customer:

- Industry
- Government
- Management
- Colleagues
- Graduate students

He contended that the PADS community served its management, colleagues and graduate students well, but that industry and government were less well-served by the community. He suggested that the PADS community should expand its notion of relevant applications to include very large models on moderately parallel platforms and distributed platforms. Further, that the PADS community should emulate relevant application environments when evaluating its approaches, better analyze scalability, and make the language of PADS more user-friendly. Lastly, Nicol stressed that PADS researchers should strive to become better experimentalists.

3. Excursions in distributed simulation

As evident from Section 2, the PADS community has done as well as any community in the area of critical self-assessment—so much so that I felt a literature survey was warranted for this Keynote! I hesitate to imagine that I could add anything substantive to the observations that have been made on the subject to date, but such is the expectation for this presentation I suppose.

In this section, I reflect briefly on some of my personal experiences from four years as a "pseudo-industrial" distributed simulationist (and the year or two preceding that), with the hope that there are perhaps some nuggets to be gleaned that echo or reinforce the thoughts and opinions others have already well-stated.

3.1. SIMNET and Distributed Interactive Simulation

Toward the end of my Ph.D. work at Virginia Tech—around the early part of 1993 or so—I became aware of a significant focus on distributed simulation that had been going on for quite some time within the U.S. Department of Defense (DoD). These efforts primarily dealt with the technologies to interconnect human-in-the-loop simulators. A technology called SIMNET was developed for that purpose and that technology had evolved into something called the Distributed Interactive Simulation (DIS) protocol. Still, from what I could tell, neither SIMNET nor DIS supported simulation of the kind I was familiar with—simulation within which *time* was a model attribute. SIMNET and DIS appeared to be little more than video game technology. In a SIMNET or DIS environment you had a bunch of different things running as fast as they could (or pacing their execution locally according to their perception of wallclock time) and exchanging data. It wasn't immediately clear to me that any of this was very interesting. After all, I remember thinking, you didn't see any of this stuff in PADS.

During the last year of my tenure at Virginia Tech I began working with a project being conducted by the Naval Surface Warfare Center (NSWC) called the Multiwarfare Assessment and Research System (MARS). MARS was to be a DIS-compliant environment that supported both training and analysis. It was being written in MODSIM II, which gave it an air of credibility with me; at least it had a discrete event simulation engine in it! During the middle-to-late part of 1994, MARS was part of a DIS exercise called "Warbreaker." Busily focusing on completing my thesis, I didn't pay much attention to the goings on. I did scan the email reports as they came in and remember being surprised that they made a very big deal out of the fact that they were—after great effort—able to get an object in one wargame to interact with an object in another wargame.

3.2. The Aggregate Level Simulation Protocol

In early 1995 I interviewed with MITRE for a position that involved developing parallel versions of a model called ModSAF (see Section 3.4). The irony wasn't lost on me that—after all that noise in PADS '94—I might be getting into the speedup business. Still, even though I didn't completely understand the problem, it seemed an exciting challenge.

As fate would have it, though, I was not hired to work on parallel ModSAF, but was rather brought on to work with the Aggregate Level Simulation Protocol (ALSP) and its primary application, the Joint Training Confederation (JTC). Probably because I could spell "validation" and because I worked with Osman Balci while at Virginia Tech, I was given the task of directing the testing process for the JTC.

The first thing to note is that the JTC consists of some *serious* applications. The primary models, like the Army's Corps Battle Simulation (CBS) and the Navy's Research, Evaluation, and Systems Analysis model (RESA), are several-hundred-thousand-line applications. They have great big architectures and connect to lots of ancillary systems. They are massively complicated and require a great deal of expertise to use individually—and more so to use in the context of ALSP.

The second thing to note is that the JTC is probably one of the most dynamic and complex applications of distributed simulation technology in existence. The JTC consists of from between 5 to 13 primary simulation models interoperating through ALSP and a collection of infrastructure software that supports the protocol. Additionally, dozens of support systems (applications that connect to the simulation network) are often used with the JTC to support training. And the population of these systems varies from event to event.

The JTC supports activities called Command Post Exercises (CPXs). These things are *major* endeavors. Typically, the training audience numbers between 500 and 1500, and the support and ancillary personnel involved number in the several thousands. Figure 1 (adapted from [29]) depicts a typical configuration for a large-scale computer-supported CPX. The term *players* refers to the training audience—in this example Corps, Division, and Brigade commanders, subordinate commanders and the affiliated battle staffs. The players are ensconsed in wartime Command Posts and communicate with higher and lower echelons using real-world mechanisms and techniques—so-called Command, Control, Communications, Computers and Intelligence (C4I) systems. In current practice, direct interfaces between C4I systems and the fielded training simulations are rare. Orders generated by the training audience are intercepted and translated into formats recognized by the training simulations by *controllers* and other personnel staffing *response cells.* Since the impacts of human interaction and widely varying experience and skill levels may result in battlefield situations that run counter to the training objectives, referees located in a *white cell* may intervene and influence the direction of the exercise and state of the simulation(s). The controllers, response cells and white cell are not visible to the training audience.

While much cheaper than live exercises, a JTC-supported exercise is a very complex and expensive endeavor. But such exercises are vital to troop readiness, and are high visibility events. So, what challenges confront distributed simulation at this level?

- The systems are too hard to configure. Database construction can literally take months of effort. For major exercises like Ulchi Focus Lens and Prairie Warrior,

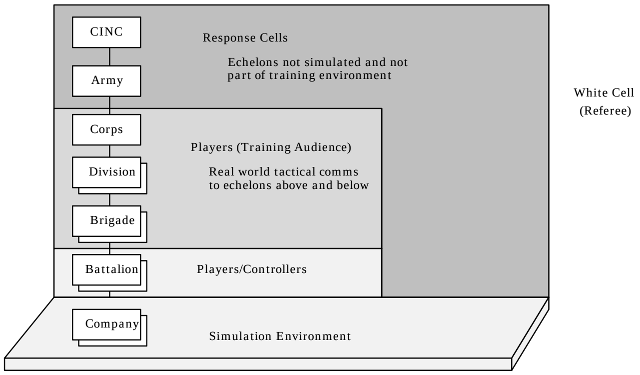

Figure 1. Architecture for a Command Post Exercise (CPX).

planning is a year-round process.

- They are too difficult to operate. A significant degree of expertise is required to run the models and support the training exercise. Since support staffs vary from exercise to exercise and people rotate in and out of assignments, support staff training must accompany every JTC event. System complexity and inadequate training result in high risks for mistakes. (Sometimes, getting that object in one model to interact with the object in another model still brings a cheer.)
- It is too difficult to analyze the results. Given the sheer number of systems involved in the exercise, data fusion for after action analysis is a big problem.

But what about performance? Is better performance desired? Absolutely. Give an exercise sponsor more computational and network capabilities, and the sponsor will consume them with a bigger scenario.

The Joint Simulation System (JSIMS) is being developed to replace the JTC and to attempt to address (at least partially) the issues identified above. JSIMS designers are currently planning on the use of SPEEDES as part of the JSIMS core infrastructure. Is synchronization software all the PADS community could offer here?

3.3. The High Level Architecture

DIS and ALSP begat the High Level Architecture (HLA). The HLA provides a unifying framework for realtime and logical time (e.g. discrete event) simulations. The HLA also addresses many of the problems associated with making simulations interoperate effectively by mandating a common model definition and specification formalism which can be used to guide the identification and integration of candidate models.

The HLA was introduced to PADS in 1996 and has had some visibility each year since. In her presentation to this workshop, Dr. Judith Dahmann will discuss the HLA and PADS in some detail, so I will forego any further treatment of it here. I'm leaving HLA in the title of this paper, though, because it's an important topic!

3.4. Semi-automated forces

Semi-automated forces (SAF) or computer-generated forces (CGF) evolved from the class of systems called SAFOR that were used in SIMNET exercises to populate the "synthetic battlefield." Typically, SAF systems include models of human behavior and decision making that enable the forces modeled to exhibit a certain degree of autonomy. Generally, though, the level of autonomy is limited and varying degrees of human-directed control during runtime are required (hence use of the modifier "semi-automated").

Notable examples of SAF include such systems as ModSAF and CCTT-SAF. A great deal of effort has gone into the design and evolution of SAFs and a large community of researchers and developers has formed around SAFs. Interestingly, the intersection of the SAF community and the discrete event simulation community is almost null. It was largely through the emergence of the HLA that many in the SAF community began discovering discrete event simulation principles and the joys of repeatability and causality.

The SAF community is currently interested in such issues as [23]:

- *Synthetic natural environments.* An emerging approach in military simulation involves the separation of models of the environment from the physical and behavioral models constituting a simulation. Independent representations of the environment make it necessary to collect and manage a large volume of data. This data may include characteristics of the terrain surface, natural and cultural features, atmosphere, sea surface, subsurface, and ocean floor. The representation of radio and acoustic energy, chemical and biological agents, natural and man-made obscurants and nuclear effects are also considered part of the environment since these create a medium within which the objects must operate.

  The collection and management of environmental data presents significant challenges, as does the validation of environmental models. Consistent environmental effects in a multiresolution simulation (see below) are particularly challenging.

  Another problem is the proliferation of representation schemes and the absence of suitable standards. To overcome this problem an effort known as the Synthetic Environment Data Representation and Interchange Specification (SEDRIS) project is being pursued. The intent is to create a common database format that completely supports the characteristics of existing databases. The principles underlying SEDRIS are that the data representation: (1) be complete, (2) support loss-less translation, and (3) be unambiguous.

- *Behavior representation.* Because of its complexity, behavioral modeling has traditionally been very basic. The goal has been to provide military vehicles and units with the ability to *react* to basic events in the absence of human intervention. These models allow aircraft on patrol to "decide" to return to base when getting low on fuel, rather than continuing until the aircraft falls to the ground. Ground units respond to enemy attacks by focusing firepower on the aggressor rather than blindly continuing their preprogrammed mission. Algorithms like these have been the extent of behavioral modeling for many years. However, more recent models have attempted to increase the reasoning capabilities of simulated objects. One approach taken by many CGF systems is to replicate the *product* of human decision making, rather than the *process.* Since we do not completely understand the inner workings of the human mind, it is easier to gather information about observed human reactions to certain situations than it is to represent the process of cogitation. However, recent developments in Artificial Intelligence (AI) technologies, e.g. intelligent agents, potentially offer new solutions to this problem.

- *Multiresolution modeling.* The notion of *resolution* in a simulation model is commonly used to describe the level of abstraction employed in the model—some measurement of the differences between the actual system under study and the model of that system. *Multiresolution* issues exist in single simulation models, but are made more acute by the system composition afforded by today's interoperability technologies (see Section 3.7). Determining meaningful "levels of discourse" between simulations federated through the HLA, for example, can be a very difficult task. A classical example of this involves the interoperation of so-called entity-level and aggregate-level simulations. Solutions to this problem involve aggregation and disaggregation, but these approaches are often *ad hoc* and generally require that objects have a *singular* level of aggregation at any point in simulation time. The problem of multiple, consistent, simultaneous, representations—at multiple levels of resolution—is exceedingly challenging [5, 6].

3.5. OneSAF

Based on the results of several recent studies and assessments of existing SAFs, the U.S. Army has begun working toward the development of its next-generation SAF system, called *OneSAF.* For the past year I have been involved in a MITRE effort to lead the specification and analysis of an architecture for OneSAF. This architectural specification will form part of the Request For Proposal (RFP) for OneSAF when it is released in 2000.

OneSAF objectives are *very* ambitious. The system is envisioned to support all aspects of Army simulation needs, including support for both training and analysis. The system is required to be highly scalable and run on commodity hardware. It must support realtime, faster-than-realtime, interactive and non-interactive execution and accommodate dynamic switching between these modes. It must be highly composable; users will have the ability to construct models from palettes of object and behavior primitives.

The SAF assessments conclude that the capabilities for OneSAF cannot be met by extending or modifying current SAF systems, but they make no recommendations beyond that. In specifying the architecture, we are confronting such questions as: (1) which style of simulation executive (realtime or discrete event) is most appropriate? (2) given interactive computation where the probability of rollback is very high, do optimistic approaches lose any inherent edge they have over conservative synchronization? and (3) do PDES techniques support high reliability, e.g. do they support continuous (interactive) execution for weeks? We are looking to the PADS community for many of our answers.

3.6. Web-based simulation

Now how's that for a buzzword? As I've said many times before, there's so much hype surrounding the web that web-based toasters are probably a hot research topic! Isn't web-based simulation more or less just the use of Java applets for simulation? That sentiment pretty much captured my thinking when I was asked to lead a small research effort in web-based simulation back in 1996. Having been involved in the area for the past few years I can tell you that, yes, a lot of work in web-based simulation involves how to play nice with Java. But, at the same time, I believe that web-based simulation has aspects of great significance. Broadly, web-based simulation includes [18]:

- *Simulation as hypermedia.* Text, images, audio, video ... simulation—the nature of the WWW design enables the production, storage and retrieval of "documents" containing any or all of these (and other kinds of) elements. The availability of simulation as a desktop, browser-based commodity has the potential to significantly alter current teaching and training methodologies, both for simulation as a technique, and for disciplines that apply simulation, like engineering, physics, and biology. Paradigms that focus on distance learning and interactive, simulation-based education and training are emerging.

- *Simulation research methodology.* The ability to rapidly disseminate models, results and publications on the web permits new approaches to the conduct of simulation research, and scientific research in general. The practical, economic and legal issues associated with the electronic publication of documents, for example, are numerous (e.g. see [25]). The electronic publication of simulation models raises additional considerations.

- *Web-based access to simulation programs.* Most commonly associated with the term web-based simulation, this area includes both the remote execution of existing (so-called "legacy") simulations from a web browser through HTML forms and CGI scripts, and the development of mobile-code simulations (e.g. those lovely applets) that run on the client side.

- *Distributed modeling and simulation.* This area includes activities that deal with the use of the WWW and web-oriented technologies (e.g. CORBA, Java RMI) as infrastructure to support distributed simulation execution [8, 12, 21, 26, 27]. Internet gaming issues are included here [3], as is research in tools, environments and frameworks that support the distributed (collaborative) design and development of simulation models [4, 7].

- *Simulation of the WWW.* Modeling and analysis of the WWW for performance characterization and optimization.

Certainly the emergence (and outright dominance) of the web brings increasing decentralization. The network is the computer. Simulation is no less affected by this technology that any other technique. Web-based simulation may simply be the natural term for ubiquitous distributed simulation: a web populated with digital objects—models of physical counterparts—is envisioned where modeling objectives are provided to search engines that, in turn, identify the appropriate digital object(s) from which to construct an experimental model [7]. Common interfaces permit the objects to interoperate at runtime [18]. The physical locations of the objects involved in the computation are not relevant to the modeler. In this envisioned future, simulation becomes ubiquitous. Model conceptualization, construction, execution and analysis is distributed, collaborative, and interactive. Levels of automated support for the modeling process significantly increase, and the pace of modeling is rapid [19, 20].

3.7. Dealing with composability

A common theme running through SIMNET, DIS, ALSP, HLA, JSIMS, OneSAF and web-based simulation is a notion of constructing systems by *composition* of well-defined components. The broader software community in recent years seems to have re-invented the object-oriented paradigm (which in the 1980s seemed to me like a reinvention of structured programming) under the auspices of patterns, frameworks and components. Perhaps I am being too cynical though (as is my tendency I think) and simply lack sufficient exposure to these concepts to merit my commenting on them. So I'll leave the matter of whether JavaBeans represents the elusive silver bullet to those more qualified to judge. However, composition in the simulation domain seems to be more than a simple extension of composition in the general software domain, and having spent a little time ruminating on the problem, here's what I think at the moment.

It seems that there are (at least) two "dimensions" of the simulation composition problem (somewhat analogous to the *is-a* and *has-a* notions from object-oriented design). In the *horizontal* dimension, simulations are composed from peer-level components. For example, a ground combat model, a sea combat model and an air combat model could be integrated to form a Theater-level combat model. Composition in this dimension introduces (invites?) multi-resolution problems (see Section 3.4). In the *vertical* dimension, simulation components at one level of abstraction are formed from the aggregation of components at higher levels of resolution. For example, constructing a tank battalion model from a collection of tank platoon models. Abstraction through aggregation seems to place at risk the identification of the most efficient solution to a problem. Or worse, such abstraction may yield poor or incorrect models. For example, a fully realized simulation of an atom based on Quantum physics could not be aggregated to provide good estimates for planetary orbits.

In addition to the risks attendant with the horizontal and vertical dimensions of composability, there seems to be an obvious scalability limit to the general reusability problem (in many ways composability seems to be a special case of the general resuability problem). As the number of candidates for reuse (composition) becomes large, the benefits of reuse (composition) become negated by the costs (money and time) of storage, organization and retrieval. The web would seem to be good example of this phenomenon. As more and more information is made available on the web, the harder it becomes to find what you're looking for, and the less useful the web becomes. If the web-based simulation vision is realized, and it becomes profitable to publish models on the web, the proliferation of models will mirror the proliferation of information on the web that we see today.

But modeling through composition requires not only identifying relevant candidates from (possibly massive) component repositories but also determining whether a combination of candidate components exists that satisfies the modeling objectives. If so, it would be desirable to identify the "best" solution, or alternately, identifying a "good" solution quickly. Such a proposition naturally introduces combinatorial explosion and intractability.

I see a real need for robust theories of composition within the distributed simulation realm. Theories in hierarchical modeling and multimodeling (e.g. [13, 15, 16, 30]) are potentially a very useful starting point, but much more could be done in this area.

3.8. Hasn't all the research has been done?

In a recent conversation I had with a corporate decision maker of significant influence, the individual told me, "Well, all the M&S research we need has been done. Now it's time to apply it." While I understand this sentiment and its motivation—certainly a significant investment has been made in simulation technology like the Synthetic Theater of War (STOW), for example, and there is well-justified motivation to see such technology infused into the mainstream and employed to support operations—any such suggestion is dangerous and simply not accurate.

Simulation, generally, has a poorly defined strategic vision. *As researchers we need to be proactive in defining the strategic vision in our area.* The PADS community can help here. Now is the time.

4. Conclusions

The question of whether or not the PADS community will survive has been answered. The community could easily continue in perpetuity on the strength of self-assessment alone! PADS is a success. As Nicol observes, students graduate and get jobs, research is funded and technology is adopted.

It is wrong to conclude that the performance problem has been solved. It will never be solved. Therefore, there will always be a need for people who think about—and devise methods for—making things go fast. On the other hand, it appears that many of the big problems facing the distributed simulation world right now are not those being addressed by PADS. Distributed simulation has become, and all indications are that it will increasingly become, an important and widespread technology. Distributed simulation faces many challenges beyond performance. It is certainly within the PADS charter to embrace those non-performance challenges. Alternatively, the PADS community might choose to leave such challenges to other organizations, like the Simulation Interoperability Standards Organization (SISO).

As a community, PADS has a proven track record of being able to solve hard problems. Lots of hard problems remain in distributed simulation. Some of the problems relate to model execution, many don't. Should the PADS community champion the solutions to these problems? Should folks like me be regarded as insiders or infiltrators?

Acknowledgements

Thanks to Stephen Turner for inviting this address and to David Nicol for providing the slides from his 1997 presentation. Thanks also to Jeff Opper and Jayne Lyons for their comments on early drafts of the paper. Lastly, thanks to David Bruce for his pearls of wisdom regarding what to do with this "poisoned chalice" that Stephen handed me.

References

[1] M. Abrams. Parallel discrete event simulation: Fact or fiction? *ORSA Journal on Computing*, 5(3):231–233, Summer 1993.

[2] R. Bagrodia. A survival guide for parallel simulation. *ORSA Journal on Computing*, 5(3):234–235, Summer 1993.

[3] T. M. Crews and K. F. Louis. Investigating interoperability in a commercial gaming engine. In A. G. Bruzzone, A. Uhrmacher, and E. H. Page, editors, *Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation*, pages 49–155, January 17-20 1999. San Francisco, CA.

[4] R. M. Cubert and P. A. Fishwick. A framework for distributed object-oriented multimodeling and simulation. In S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, editors, *Proceedings of the 1997 Winter Simulation Conference*, pages 1315–1322, December 7-10 1997. Atlanta, GA.

[5] P. K. Davis and J. Bigelow. Introduction to multiresolution modeling (MRM) with an example involving precision fires. In A. F. Sisti, editor, *Proceedings of SPIE: Enabling Technologies for Simulation Science II*, pages 14–27, April 14-16 1998. Orlando, FL.

[6] P. K. Davis and J. H. Bigelow. Experiments in multiresolution modeling. available from the DARPA ASTT web site: http://www.astt.com/, 1998.

[7] P. A. Fishwick. Issues with web-publishable digital objects. In A. F. Sisti, editor, *Proceedings of SPIE: Enabling Technologies for Simulation Science II*, pages 136–142, April 14-16 1998. Orlando, FL.

[8] G. C. Fox, W. Furmanski, S. Nair, H. T. Ozdemir, Z. O. Ozdemir, and T. A. Pulikal. WebHLA – an interactive multiplayer environment for high performance distributed modeling and simulation. In A. G. Bruzzone, A. Uhrmacher, and E. H. Page, editors, *Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation*, pages 163–168, January 17-20 1999. San Francisco, CA.

[9] R. M. Fujimoto. Future directions in parallel simulation research. *ORSA Journal on Computing*, 5(3):245–248, Summer 1993.

[10] R. M. Fujimoto. Parallel discrete event simulation: Will the field survive? *ORSA Journal on Computing*, 5(3):213–230, Summer 1993.

[11] R. M. Fujimoto. Ten years of PADS: Where we've been, where we're going. In M. L. Bailey, W. M. Loucks, and B. R. Preiss, editors, *Proceedings of the 10th Workshop on Parallel and Distributed Simulation*, May 22-24 1996. Philadelphia, PA.

[12] U. Klein, S. Straßburger, and J. Beikrich. Distributed simulation with JavaGPSS based on the High Level Architecture. In P. A. Fishwick, D. R. C. Hill, and R. Smith, editors, *Proceedings of the 1998 International Conference on Web-Based Modeling and Simulation*, pages 85–90, January 11-14 1998. San Diego, CA.

[13] K. Lee and P. A. Fishwick. OOPM/RT: a multimodeling methodology for real-time simulation. *ACM Transactions on Modeling and Computer Simulation*, 1998. submitted.

[14] Y.-B. Lin. Will parallel simulation research survive? *ORSA Journal on Computing*, 5(3):236–238, Summer 1993.

[15] J. J. Luna. Hierarchical relations in simulation models. In G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, editors, *Proceedings of the 1993 Winter Simulation Conference*, pages 132–137, December 12-15 1993. Los Angeles, CA.

[16] V. T. Miller and P. A. Fishwick. Graphic modeling using heterogeneous hierarchical models. In G. W. Evans, M. Mollaghasemi, E. C. Russell, and W. E. Biles, editors, *Proceedings of the 1993 Winter Simulation Conference*, pages 612–617, December 12-15 1993. Los Angeles, CA.

[17] D. M. Nicol. Parallel discrete event simulation: So who cares? In A. Ferscha, R. Ayani, and C. Tropper, editors, *Proceedings of the 11th Workshop on Parallel and Distributed Simulation*, June 10-13 1997. Lockenhaus, Austria.

[18] E. H. Page. The rise of web-based simulation: Implications for the High Level Architecture. In D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, editors, *Proceedings of the 1998 Winter Simulation Conference*, pages 1663–1668, December 13-16 1998. Washington, D.C.

[19] E. H. Page, A. Buss, P. A. Fishwick, K. J. Healy, R. E. Nance, and R. J. Paul. The modeling methodological impacts of web-based simulation. In P. A. Fishwick, D. R. C. Hill, and R. Smith, editors, *Proceedings of the 1998 International Conference on Web-Based Modeling and Simulation*, pages 123–128, January 11-14 1998. San Diego, CA.

[20] E. H. Page, A. Buss, P. A. Fishwick, K. J. Healy, R. E. Nance, and R. J. Paul. Web-based simulation: Revolution or evolution? *ACM Transactions on Modeling and Computer Simulation*, 1999. submitted March.

[21] E. H. Page, R. L. Moose, and S. P. Griffin. Web-based simulation in Simjava using Remote Method Invocation. In S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, editors, *Proceedings of the 1997 Winter Simulation Conference*, pages 468–474, December 7-10 1997. Atlanta, GA.

[22] E. H. Page and R. E. Nance. Parallel discrete event simulation: A modeling methodological perspective. In D. K. Arvind, R. Bagrodia, and J. Y.-B. Lin, editors, *Proceedings of the 8th Workshop on Parallel and Distributed Simulation*, pages 88–93, July 6-8 1994. Edinburgh, Scotland, UK.

[23] E. H. Page and R. Smith. Introduction to military training simulation: A guide for discrete event simulationists. In D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, editors, *Proceedings of the 1998 Winter Simulation Conference*, pages 53–60, December 13-16 1998. Washington, D.C.

[24] P. F. Reynolds. The silver bullet. *ORSA Journal on Computing*, 5(3):239–241, Summer 1993.

[25] P. Samuelson. Regulation of technologies to protect copyrighted works. *Communications of the ACM*, 39(7):17–22, July 1996.

[26] H. S. Sarjoughian and B. P. Zeigler. DEVSJAVA: Basis for a DEVS-based collaborative M&S environment. In P. A. Fishwick, D. R. C. Hill, and R. Smith, editors, *Proceedings of the 1998 International Conference on Web-Based Modeling and Simulation*, pages 29–36, January 11-14 1998. San Diego, CA.

[27] C.-C. Shen. Discrete-event simulation on the internet and the web. In P. A. Fishwick, D. R. C. Hill, and R. Smith, editors, *Proceedings of the 1998 International Conference on Web-Based Modeling and Simulation*, pages 57–62, January 11-14 1998. San Diego, CA.

[28] B. W. Unger and J. G. Cleary. Practical parallel discrete event simulation. *ORSA Journal on Computing*, 5(3):242–244, Summer 1993.

[29] A. A. Zabek. ALSP update. Project Briefing, The MITRE Corporation, Not in Public Domain, May 1994.

[30] B. P. Zeigler. *Object-Oriented Simulation and Hierarchical, Modular Models*. Academic Press, Orlando, FL, 1990.