16

Theory and Practice for Simulation Interconnection: Interoperability and Composability in Defense Simulation

Ernest H. Page
The MITRE Corporation

If you nail two things together that have never been nailed together before, some schmuck will buy it from you.

George Carlin

16.1 Introduction

For the better part of the past quarter century, the defense simulation industry has invested significant resources in technologies and methods for making independently developed simulations work together at run-time. Many reasons for this activity exist. The initial impetus was the need for a common "synthetic environment" that could interconnect simulators in support of small-team training—such environments are now commonplace to today's online gamers. Subsequently, the notion of "moving electrons to the

0-8493-XXXX-X/02$0.00+$1.50
©2002 by CRC Press LLC

people" rather than "moving people to the electrons" led to a proliferation of geographically distributed simulation-based training environments. In addition, there was the belief that each Service (e.g., Army, Navy, Air Force, and Marines) could best model that Service's capabilities, and if you needed to access such capabilities, you should do so by interconnecting with that Service's "authoritative" simulations. And of course, the notion of cost savings through simulation *reusability* also drove the development of technologies for simulation interconnection. Today, simulation interconnection is pervasive in the defense simulation arena.

Issues in simulation interconnection are variously referred to as *integration*, *interoperation*, *composition*, *configuration*, and more. Today, the dominant notions are *interoperability* and *composability*. And while there is some ambiguity in their usage, for purposes of this chapter, interoperability is the realm of the practical aspects of simulation interconnection, and composability encompasses the theoretical work in simulation interconnection.

To a large extent, the simulation interconnection problem can be viewed as a computer science problem, tackled by the "modern miracles" of standards, middleware, distributed algorithms, data type coercion, and so forth. The practice of simulation interconnection (interoperability) is reviewed in these terms. But there is also the notion of what it *means* when you interconnect two simulations. And while this topic has received much less attention in the defense simulation arena than the more immediately tractable issues of aligning bits and bytes, some notable work has been done, and is covered under the theory of simulation interconnection (composability). A careful reader will no doubt observe a close connection between the issues confronted in the theory of simulation interconnection and the fundamental concerns of multiscale, multiresolution, and multiformalism modeling.

A tremendous volume of literature exists in this area. Simulation interconnection is, for practical purposes, an industry unto itself. There are several longstanding conferences and workshops devoted to the topic, most notably the Simulation Interoperability Workshops (see SISO, 2005). Entire texts are dedicated to aspects of the problem, e.g., the high-level architecture (HLA) (Kuhl et al., 1999), and simulation interconnection represents the primary business for many companies that support the defense simulation industry. The goal of this chapter is to provide a broad and gentle introduction to the topic, with lots of pointers to the more comprehensive, detailed sources.

And, just for the record: George Carlin was offering general advice to aspiring entrepreneurs. As far as this author is aware, Mr. Carlin had no connections with the defense simulation industry!

16.2 The Practice of Simulation Interconnection—Simulation Interoperability

The lineage of the practice of simulation interconnection is typically traced from the Defense Advanced Research Project Agency (DARPA) simulator networking (SIMNET) project, through the development of the distributed interactive simulation (DIS) protocols and aggregate level simulation protocol (ALSP), to the current approach defined by HLA for modeling and simulation. Each of these is briefly surveyed here. Interested readers should consult Voss (1993) and Miller and Thorpe (1995) for histories of SIMNET and DIS, Miller and Zabek (1996) and Weatherly et al. (1996) for ALSP, and Kuhl et al. (1999) for HLA.

Interconnecting simulations over computer networks may be rightly viewed as an application of *distributed simulation*. Within the defense simulation arena, work in simulation interconnection has sometimes been referred to as advance distributed simulation (ADS). Here, the distributed simulations are mostly (but not exclusively) used for training, and the purpose of distributed execution is enhanced functionality. A newcomer to this area should be aware that there is a separate body of work in distributed simulation whose objective is to reduce the execution time of a simulation by utilizing multiple processors. In his comprehensive text on parallel and distributed simulation, Fujimoto (2000) refers to distributed simulation in the defense community as distributed virtual environments (DVEs).

16.2.1 Simulator Networking

Prior to the 1980s, simulators were very expensive, special purpose devices used to train an individual in the essential skills needed to operate the real platform that the simulator represented. Although a few cases of one-to-one locally interconnected air combat simulators had been created, networking technology was inadequate to support the interconnection of large numbers of simulators or the interconnection of simulators at great distances. The evolution of the ARPAnet and its concomitant technologies (e.g., packet switching) changed things, however, and in 1983 the DARPA launched the SIMNET project. The purpose of SIMNET was to investigate the capability of networked simulators to support group (or "collective") training in large scales and at great distances. The idea was that a large-scale, interactive, networked simulation created a "synthetic environment" that could be entered by any authorized combatant from anywhere on the network using his simulator as a porting device (Miller and Thorpe, 1995). The initial project scope was to develop a SIMNET testbed with at least four geographically distributed sites with 50–100 vehicle simulators each. Because it was believed that the networking technologies of the day would not support the demands (owing to speed and maneuverability) of aircraft simulators, the initial SIMNET testbed focused on slower moving ground-based platforms, e.g., tanks and armored personnel carriers.

SIMNET spurred technical advancements in both computer networking and image generation, and is the basis for a variety of successful service programs, including the U.S. Army's Close Combat Tactical Trainer (CCTT). Most of the fundamental design principles underlying SIMNET have demonstrated lasting value for simulation interconnection (Miller and Thrope, 1995):

- *Selective fidelity*. To minimize simulator costs, a simulator should only contain high fidelity representations of those elements essential to the training task. Everything else should be represented at lower fidelities, or not all.
- *Autonomous simulation nodes*. Each node is responsible for maintaining the state of at least one object in the synthetic environment, and for communicating to other nodes any events caused by its object(s). Each node receives event reports from other nodes and calculates the effects of those events on its objects. All events are broadcast on the simulation network, and are available to any node that is interested. There is no centralized controlling process. Nodes may join and leave the network without affecting other nodes. Each node advances simulation time according to a local clock (typically a hardware clock).
- *Transmission of "ground truth" data*. Each node transmits the absolute truth about the current state of the object(s) it represents. Alteration of data to suit simulation objectives is the responsibility of the receiving node. For example, the position of a vehicle is broadcast to the network with 100% accuracy. If an object in another simulator determines that it could perceive the vehicle through a particular sensor, but with an accuracy determined by the alignment of the sensor and current weather conditions, then the receiving simulator should degrade the reported position accordingly.
- *Transmission of state change information*. To minimize communications processing, nodes transmit state update information only. To accommodate late-joining nodes and networks with high packet loss, this rule is often relaxed. In these situations, nodes send periodic (but relatively infrequent) updates for each owned object regardless of whether or not their state changes. This update interval is known as the "heartbeat."
- *Dead reckoning*. Between state update messages, receiving nodes may extrapolate the last reported state of remote objects that are of interest. To keep the extrapolated values from becoming too far afield of the actual values, the sending node maintains the same approximation and transmits a state update whenever the true position (or orientation) of an object diverges from the calculated dead reckoned values by more than an agreed-upon threshold. Fujiomoto (2000, p. 206) discusses common dead reckoning algorithms.

AQ1

Other mainstays of modern defense simulation introduced by SIMNET include semi-automated forces (SAF) and the "flying carpet" (or "stealth") display.

16.2.2 Distributed Interactive Simulation

Following the successful demonstrations of SIMNET in the late 1980s, the defense simulation community undertook an industry-wide effort to define a set of standard networking protocols for interconnecting simulations. This work was accomplished largely within a series of semiannual workshops, and the DIS protocols became an IEEE standard in the spring of 1993 (Pullen and Wood, 1995; Voss, 1993). The primary mission of DIS is (University of Central Florida, 1993, p. 3)

> . . . to create synthetic, virtual representations of warfare environments by systematically connecting separate subcomponents of simulation which reside at distributed, multiple locations . . . The property of connecting separate subcomponents or elements affords the capability to configure a wide range of simulated warfare representations patterned after the task force organization of actual units . . . Equally important is the property of interoperability which allows different simulation environments to efficiently and consistently interchange data elements essential to representing warfighting outcomes.

The fundamental design principles for DIS follow directly from SIMNET. Most of the standardization effort focused on extending the basic SIMNET communication structure, the protocol data unit (PDU), a bit-encoded packet for communicating entity state, and other types of information identified as useful for distributed combat simulations, e.g., weapons fire and weapons detonation events.

Like SIMNET, DIS was primarily designed to support the interconnection of simulations that (1) run in real-time, and (2) have a significant visual component. A great deal of focus in the DIS arena dealt with minimizing network latencies for PDUs. The creation of DIS led to a burgeoning market in SAF. SAFs were used to populate synthetic environments with background objects that behaved in a "reasonable" way. They were "semiautomated" because human intervention was often required to make the modeled entities maintain their reasonable behavior. The power and utility of SAFs was recognized very quickly, and eventually DIS-supported simulation environments consisting entirely of SAFs emerged.

DIS has been used as the protocol underlying numerous warfighting experiments and advanced concepts technology demonstrations (ACTDs), most notably, the Synthetic Theater of War (STOW) family of experiments.

16.2.3 Aggregate Level Simulation Protocol

As noted by Page and Smith (1998), defense simulation has a vernacular that can be nonintuitive to simulationists from outside the defense arena. For example, a commonly applied taxonomy for defense simulation is the *virtual*, *live*, *constructive* taxonomy (U.S. Department of Defense, 1997):

- *Virtual simulation* refers to a simulation involving real people operating simulated systems. Virtual simulations inject human-in-the-loop in a central role by exercising motor control skills (e.g., flying an airplane), decision skills (e.g., committing fire control resources to action), or communication skills (e.g., as members of a C4I team).
- *Live simulation* refers to a simulation involving real people operating real systems.
- *Constructive simulation* refers to a simulation that involves simulated people operating in simulated systems. Real people stimulate (make inputs) to such simulations, but are not involved in determining the outcomes.

Essentially, virtual simulation refers to the use of simulators, live simulation to rehearsal, or practice with "go-to-war" systems, and constructive simulation refers to "classical" computerized simulation models. These classical simulation models are also categorized with respect to their inherent level of abstraction. If the simulation includes explicit representation of individual vehicles, it is referred to as an *entity-level simulation*. If, however, the basic unit of representation in the simulation corresponds to a military echelon, e.g., a platoon, company, brigade, or battalion, then the simulation is referred to as an *aggregate-level simulation*.

Like the DIS protocols, the ALSP is rooted in SIMNET, but ALSP was targeted toward support for the interoperation of aggregate-level simulations used within command post exercises (Page et al., 1997; Weatherly et al., 1993, 1996). In addition, ALSP supported the explicit representation and synchronization of simulation time using a variant of the Chandy–Misra–Bryant protocol (Chandy and Misra, 1978; Bryant, 1977). AQ2

Fielded in the spring of 1991, the ALSP Joint Training Confederation (JTC), currently known as the Joint Training Transformation Initiatives Plus (JTTI+), has been successfully employed to support numerous major, large-scale, joint training exercises, including the annual Ulchi Focus Lens, Prairie Warrior, and Unified Endeavor exercises (Miller and Zabek, 1996; ALSP, 2005).

16.2.4 High Level Architecture

By 1995, SIMNET, DIS, and ALSP had each contributed to the demonstration that interconnecting simulations could be of practical value. SIMNET provided an efficient and effective mechanism for linking simulators. DIS extended SIMNET and provided scalability to many thousands of entities in SAF-based exercises. ALSP provided support for synchronization required to interconnect "logical time," e.g., discrete event, simulations. By 1995, many defense simulations had interconnection interfaces—some SIMNET, some DIS, some ALSP, some "home grown." To mitigate against the proliferation of homegrown interconnection standards, the U.S. Department of Defense (DoD) established a *unifying* standard for simulation interconnection known as the High Level Architecture (HLA). The HLA represents both a generalization and extension of SIMNET, DIS, and ALSP, and is defined by three components:

- *an object model template*—a common model definition and specification formalism,
- *an interface specification*—a collection of services describing the HLA run-time environment, and
- the *HLA rules*—governing compliance with the architecture.

The HLA is intended to have applicability across the full range of defense simulation applications, including those used to support training, analysis, mission rehearsal, and acquisition.

At the heart of the HLA is the notion of a *federation*. A federation is a collection of federates—simulations and other systems—that interoperate using the protocols described by the architecture. A federation object model (FOM) provides the model specification and establishes a contract between the federates regarding the nature of the activity taking place during federation run-time. Federation execution is accomplished through an HLA run-time infrastructure (RTI), which is an implementation of the infrastructure services defined by the architecture. In addition to defining services for the RTI, the HLA interface specification also defines services that must be implemented by federates.

In a typical federation execution, a federate joins the federation, indicates its operating parameters (e.g., information the federate will provide to the federation and information it will accept from the federation), and then participates in the evolution of federation state until the federate departs the federation or the simulation terminates. FOM data are provided to the RTI at run-time, enabling the infrastructure to provide a level of enforcement with respect to the "information contract" that the FOM represents.

16.2.5 Summary Thoughts on the Practice of Simulation Interconnection

It would be hard to argue that the pursuit of making independently developed systems work together at run-time has not been worthwhile for the defense simulation community. The technology has been used successfully too many times to condemn it. However, there is still the sense in the defense simulation community that simulations generally, and federations of simulations particularly, are expensive, difficult to use, and fragile. The community still has not achieved the level of interoperability that it would like. Although HLA was (is) a unifying standard, a great many DIS and a few ALSP applications still persist. The technology seems to be almost in a perpetual proof-of-concept mode. Organization after organization still spend nontrivial sums on the creation of federations with minimal utility. One the one hand, this seems like a bad sign. On the other, fundamental change within government organizations takes time, and 25 years is probably too soon to fully evaluate the impact of this technology.

16.3 The Theory of Simulation Interconnection—Simulation Composability

In recent years, the defense simulation community has begun to complement a robust practice of simulation interconnection with some regard to the development of supportive theories. Most of this work has arguably been accomplished under the rubric of simulation *composability*. Like many terms from the DoD lexicon, the notion of composability is vaguely and disparately applied, as evidenced by the history below, adopted from Page et al. (2004).

16.3.1 A Brief History of Composability

The earliest uses of the term composability within the defense simulation context date to the Composable Behavioral Technologies (CBT) project during the mid-1990s (see Courtemanche et al., 1997). The purpose of CBT was to give ModSAF users a convenient way to develop new entity behaviors without appealing to the underlying SAF source code. Shortly after the initiation of CBT, composability appeared as a system objective within the Joint Simulation System (JSIMS) Mission Needs Statement. A taxonomy and use case for JSIMS composability appears in (JSIMS Composability Task Force, 1997), and the impact of composability as a system objective on the JSIMS design is described in Pratt et al. (1999). Composability also appeared as a key system objective for OneSAF in 1999 (U.S. Army STRICOM, 2000).

In 1998, the DARPA Advanced Simulation Technology Thrust (ASTT)—which was chartered to develop technology in support of JSIMS—funded two separate studies on simulation composability: (1) the model-based simulation composition (MBSC) project, which developed a prototype composition environment for JSIMS (Aronson and Wade, 1998, 2000; Davis and Aronson, 1999; Wade and Aronson, 1999); and (2) a study by Page and Opper (1999) that investigated the composability problem from a computability and complexity theoretic perspective.

A focus-paper session at the 2000 Winter Simulation Conference addressed methodologies for composable simulation (Kasputis and Ng, 2000; Davis et al., 2000). Recently, the work of Petty, Weisel, and Mielke (Petty and Weisel, 2003a, 2003b; Petty et al., 2003, 2005; Weisel et al., 2003, 2005) provides a broad survey of the uses of the term composability, extends the work of Page and Opper, and examines the composite validation problem within the context of automata theory and computable functions.

The Defense Modeling and Simulation Office (DMSO) initiated a collection of studies as part of the Composable Mission Space Environments (CMSE) initiative in FY03 (DMSO, 2005). The comprehensive report by Davis and Anderson (2003) provides a broad survey of the topic of composability and suggests a wide-ranging investment strategy for the DoD in this area.

Related work outside the defense simulation community included the emergence of the topic of "web-based simulation" in the mid-1990s, which included the concept of composing simulations via web protocols (see Fishwick, 1996; Fishwick et al., 1998; Page et al., 2000). Most recently, the Extensible Modeling and Simulation Framework (XMSF) was initiated by the Naval Postgraduate School, George Mason University, SAIC, and Old Dominion University to develop a ubiquitous web-based simulation environment (XMSF, 2005).

16.3.2 Composability and Complexity

Motivated by the high degrees of automated support for composability expressed within the JSIMS and OneSAF program requirements, Page and Opper (1999) consider composition from a computability and computational complexity theoretic perspective. The authors observe that prior work in analyzing simulation model specifications suggests that many of the problems attendant with simulation model development, verification, and validation are *fundamentally hard*, and that automation can only provide so much relief (Page and Opper, 1999, p. 554). For example, problems such as the following cannot be solved in the general case: (1) determining if a model is finite (i.e., will run to completion); (2) determining if a model specification is complete; and (3) determining if a model specification is minimal. Other analyses

have shown that problems such as the following have no efficient solution: (1) determining whether any given state will occur during an execution of a model; (2) determining the existence of, or possibility for, simultaneous events; and (3) determining whether a model implementation satisfies a model specification.

Motivating the work of Page and Opper (1999) was the fairly simple observation that the class of *model specifications* must include the class of *model compositions*, and therefore all prior results in the computational complexity of model specifications also applied to model compositions. In support of this observation, the authors develop a formal analysis of a simple, generic methodology for composable simulation. They observe that building simulation models by composition implies not only identifying (via search) relevant candidates from (possibly massive) component repositories, but also answering the following: (1) does a combination of components exist that satisfies the modeling objectives, and (2) if so, can the "best" (or a "good enough") solution be identified in a reasonable time. If not, how closely can the objectives be met?

Determining whether a collection of components satisfies a modeling objective might be accomplished in any number of ways, including

- Determination made strictly on the basis of the descriptions of component capabilities (i.e., metadata).
- Determination made by modeling or approximating component interactions.
- Determination made by constructing the composed model and observing the result(s).

Page and Opper observe that a determination made on the basis of metadata is the least computationally intensive solution—assuming such a determination was possible. They suggest a simple formal model of composition based on set theory and describe a generic decision problem for composability as follows:

COMPOSABILITY
INSTANCE: A set O of objectives and a collection C of components.
QUESTION: Is there a valid composition that meets the objectives stated in O?

The authors conjecture that the decision problem COMPOSABILITY is NP-complete. In the development of a proof of this conjecture, the authors observe, however, that certain objectives may be undecidable on their face, e.g., the simulation terminates for a given set of circumstances. To accommodate this, Page and Opper suggest two variants of the COMPOSABILITY decision problem: (1) BOUNDED COMPOSABILITY—each objective in O is decidable; and (2) UNBOUNDED COMPOSABILITY—some objective in O is undecidable. Further, the authors observe that it may be possible for two components, A and B, to satisfy some objective O and that their ability to satisfy O could not be predicted based on any metadata for A and B. Page and Opper suggest that this characteristic is like the property of emergence in complex adaptive systems and suggests two more variants of the COMPOSABILITY decision problem: (1) EMERGENT COMPOSABILITY—composition cannot be evaluated based on metadata; and (2) NONEMERGENT COMPOSABILITY—the composition can be evaluated based on metadata.

The cross product of the variants yields four decision problems:

- UNBOUNDED EMERGENT COMPOSABILITY
- BOUNDED EMERGENT COMPOSABILITY
- UNBOUNDED NONEMERGENT COMPOSABILITY
- BOUNDED NONEMERGENT COMPOSABILITY (BNC).

From a complexity perspective, BNC is the simplest. Page and Opper provide a proof that BNC is NP-complete. This proof suggests that, use cases for composability that imply automated support for determining valid combinations of components (e.g., Steps 2 and 5 from the JSIMS composability use case [JSIMS, 1997]) cannot have an efficient solution in the general case.

Petty et al. (2003) extend the work of Page and Opper and suggest another variant of the problem, ANTI-EMERGENT COMPOSABILITY (AC). In AC, two models A and B may each satisfy some objective O, but their combination does not. Petty, Weisel, and Mielke suggest a general form of the component selection problem that subsumes the variants and prove that it is NP-complete.

16.3.3 Formalisms for Composability

Petty and Wiesel (2003b) develop a formal model of simulation composition based on computable functions operating on vectors of integers. In this formalism, the composition of simulation models is isomorphic to the composition of computable functions. The authors suggest a mechanism to measure the validity of a composition by comparing its output vector to the output vector of a "perfect model," a model whose outputs perfectly correspond to the outputs of the system being modeled. Weisel et al. (2005) evaluate their formalism with respect to the DEVS formalism, noting their analytic equivalence despite the fact that DEVS is not explicitly restricted to computable functions.

16.3.4 Proposal to Restrict the Scope of Composability

Page et al. (2004) observe that despite the decades of successful practice in simulation interconnection and the solid theoretical work ongoing by Petty and others in the context of simulation composability, a refinement of the terminology in use by the defense simulation community could serve to better focus the community's research and development efforts.

As illustrated by the history of composability above, the term composability is used within the military simulation domain to imply a variety of notions, ranging from interoperability, to end-user tailorability, to any act of creation. Page et al. (2004) suggest that for the term to be most useful, it should be unambiguously differentiated from these other concepts.

The definition suggested by Petty and Weisel (2003a) loosely differentiates the notions of composability and interoperability as follows:

> Essentially, interoperability is the ability to exchange data or services at run-time, whereas composability is the ability to assemble components prior to run-time . . . It can be seen that interoperability is necessary but not sufficient to provide composability. Composability (engineering and modeling) does require interoperability (technical and substantive). Federates that are not interoperable can not be composed, so interoperability is necessary for composability. However, interoperability is not sufficient to provide composability, i.e. federates may be interoperable but not composable. Recall that an essential aspect of composability is the ability not just to combine federates but to combine and recombine federates into different simulation systems. Federates that are interoperable in one specific configuration or with one specific object model, and cannot be combined and recombined in other ways, are not composable . . . The matter of substantial effort is crucial to the distinction between interoperability and composability.

These distinctions seem somewhat problematic, because it is not immediately clear how a run-time characteristic (interoperability) could be a necessary condition to enable a prerun-time characteristic (composability). Further, the term "substantial effort" requires quantification.

Petty and Weisel (2003a) distinguish composability and integratability as follows:

> Integration is the process of configuring and modifying a set of components to make them interoperable and possibly composable. Essentially any federate can be integrated into any federation with enough effort, but composability implies that the changes can be made with little effort.

This distinction also seems somewhat problematic since the level of "effort" is also used to distinguish between composability and interoperability.

As an alternative, Page et al. (2004) suggest that composability be viewed as a property of a set of models. Specifically, it should be expressed as some function on the congruity of the objectives and assumptions underlying each model in the set. In this sense, they agree with the definition of Petty and Weisel. That is, composability is a property that may be assessed prior to run-time. However, in their view *composability is independent from interoperability*. Interoperability is a property of the software implementation of a set of models (or other systems). The objectives and assumptions underlying two models, A and B, may be wholly congruent and thus the models are composable, but their software implementations may utilize different

programming languages, data types, marshaling protocols, and so forth such that the two implementations are not interoperable.

Informally, two models are composable if they share compatible objectives and assumptions. Quantifying and reasoning about the "compatibility" of objectives and assumptions should be the domain of research in algebras and calculi for composition. Thus, Page et al. (2004) suggest the following framework:

- *Composability*—realm of the model (e.g., two models are composable if their objectives and assumptions are properly aligned).
- *Interoperability*—realm of the software implementation of the model (e.g., are the data types consistent, have the little endian/big endian issues been addressed, etc.)
- *Integratability*—realm of the site the simulation is running at (e.g., have the host tables been set up, are the NIC cards working properly).

16.3.5 Summary Thoughts on the Theory of Simulation Interconnection

The defense simulation lexicon is generally underwhelming—although this may simply reflect the author's academic bias—and the community treatment of the terms surrounding the simulation interconnection problem is no exception. While, the definitions promulgated by Petty et al. and currently embraced by the defense simulation community, are not as "crisp" as they could perhaps be, their theoretical work in the composition of models is a great service to the community. We do a reasonably good job, as a community, in getting the bits to flow between simulations. We do not have a good handle on how to reason about the semantics of the interoperating simulations once the bits start flowing. Algebras and calculi of composition are much in need here.

16.4 Conclusions

For the better part of the past quarter century, the defense simulation industry has been heavily invested in technologies and methods to make independently developed simulations work together. This pursuit has met with significant technological successes, notably: SIMNET, DIS, ALSP, and HLA. The pressures that drove the development of this technology were primarily fiscal. Defense simulations represent a significant investment, and mechanisms for their reuse must be defined to preserve the defense simulation community's capabilities in an era of ever-shrinking budgets. Like many areas where the realities of the day demand a solution, the practice of simulation interconnection has led the theory. But important theoretical work is beginning to take shape to quantify and reason about the semantics of interconnected simulations.

References

Aggregate Level Simulation Protocol (ALSP). 2005. Homepage. http://ms.ie.org/alsp.

Aronson, J. and Wade, D. 1998. Model based simulation composition. In *Proceedings of the Fall 1998 Simulation Interoperability Workshop*, 98F-SIW-055.

Aronson, J. and Wade, D.M. 2000. Benefits and pitfalls of composable simulation. In *Proceedings of the Spring 2000 Simulation Interoperability Workshop*, 00S-SIW-155.

Bryant, R.E. 1977. Simulation of packet communication architecture computer systems. Technical Report MIT-LCS-TR-188, Massachusetts Institute of Technology.

Chandy, K.M. and Misra, J. 1979. Distributed simulations: case study in design and verification of distributed programs, *IEEE Transactions on Software Engineering*, **SE-5**(5): 440–452.

Courtemanche, A.J., von der Lippe, S.R., and McCormack, J. 1997. Developing user-composable behaviors. In *Proceedings of the Fall 1997 Simulation Interoperability Workshop*, 97F-SIW-068.

Davis, D. and Aronson, J. 1999. Component selection techniques to support composable simulation. In *Proceedings of the Spring 1999 Simulation Interoperability Workshop*, 99S-SIW-043.

AQ3

Davis, P.C., Fishwick, P.A., Overstreet, C.M., and Pegden, C.D. 2000. Model composability as a research investment: Responses to the featured paper. In *Proceedings of the 2000 Winter Simulation Conference*, pp. 1585–1591.

Davis, P.K. and Anderson, R.H. 2003. *Improving the Composability of DoD Models and Simulations*, RAND, National Defense Research Institute, Santa Monica, CA.

Defense Modeling and Simulation Office (DMSO). 2005. Composable mission space environments. Homepage. https://www.dmso.mil/public/warfighter/cmse/

Extensible Modeling and Simulation Framework (XMSF). 2005. Homepage. http://www.movesinstitute.org/xmsf/xmsf.html

Fishwick, P., Hill, D., and Smith, R. (Eds.). 1998. *Proceedings of the 1998 SCS International Conference on Web-Based Modeling and Simulation*, 11–14 January, San Diego, CA.

Fishwick, P.A. 1996. Web-based simulation: Some personal observations. In *Proceedings of the 1996 Winter Simulation Conference*, pp. 772–779.

Fujimoto, R.M. 2000. *Parallel and Distributed Simulation Systems*. New York: Wiley.

AQ3

Fujimoto, R.M., Lunceford, W.H., Page, E.H., and Uhrmacher, A.M. (Eds.). 2002. *Proceedings of the Dagstuhl Seminar on Grand Challenges for Modeling and Simulation*, Seminar 02351, 26–30 August, Dagstuhl, Germany. Available at: http://www.informatik.uni-rostock.de/~lin/ GC/report/index.html

JSIMS Composability Task Force. 1997. Final Report.

Kasputis, S. and Ng, H.C. 2000. Composable simulations. In: *Proceedings of the 2000 Winter Simulation Conference*, pp. 1577–1584.

Kuhl, F., Weatherly, R., and Dahmann, J. 1999. *Creating Computer Simulation Systems—An Introduction to the High Level Architecture*. New Jersey: Prentice-Hall.

Miller, D.M. and Thorpe, J.A. 1995. SIMNET: The advent of simulator networking. *Proceedings of the IEEE*, **83**(8): 1114–1123.

Miller, G. and Zabek, A.A. 1996. The joint training confederation and the aggregate level simulation protocol. *MORS Phalanx*, **29**: 24–27.

Page, E.H., Briggs, R., and Tufarolo, J.A. 2004. Toward a family of maturity models for the simulation interconnection problem. In *Proceedings of the Spring 2004 Simulation Interoperability Workshop*, 04S-SIW-145.

Page, E.H., Buss, A., Fishwick, P.A., Healy, K.J., Nance, R.E., and Paul, R.J. 2000. Web-based simulation: Revolution or evolution? *ACM Transactions on Modeling and Computer Simulation*, **10**(1): 3–17.

Page, E.H., Canova, B.S., and Tufarolo, J.A. 1997. A case study of verification, validation and accreditation for advanced distributed simulation. *ACM Transactions on Modeling and Computer Simulation*, **7**(3): 393–424.

Page, E.H. and Opper, J.M. 1999. Observations on the complexity of composable simulation. In *Proceedings of the 1999 Winter Simulation Conference*, pp. 553–560.

Page, E.H. and Smith, R. 1998. Introduction to military training simulation: A guide for discrete event simulationists. In *Proceedings of the 1998 Winter Simulation Conference*, pp. 53–60.

Petty, M.D. and Weisel, E.W. 2003a. A composability lexicon. In *Proceedings of the Spring 2003 Simulation Interoperability Workshop*, 03S-SIW-023.

Petty, M.D. and Weisel, E.W. 2003b. A formal basis for a theory of semantic composability. In *Proceedings of the Spring 2003 Simulation Interoperability Workshop*, 03S-SIW-054.

Petty, M.D., Weisel, E.W., and Mielke, R.R. 2003. Computational complexity of selecting components for composition. In *Proceedings of the Fall 2003 Simulation Interoperability Workshop*, 03F-SIW-072.

Petty, M.D., Weisel, E.W., and Mielke, R.R. 2005. Composability theory overview and update. In *Proceedings of the Spring 2005 Simulation Interoperability Workshop*, 05S-SIW-063.

Pratt, D., Ragusa, C., and von der Lippe, S. 1999. Composability as an architecture driver. In *Proceedings of the 1999 I/ITSEC Conference*.

Pullen, J.M. and Wood, D.C. 1995. Networking technology and DIS. In *Proceedings of the IEEE*, **83**(8): 1157–1167.

Simulation Interoperability Standards Organization (SISO). 2005. Homepage. http://www.sisostds.org.

U.S. Army Simulation, Training and Instrumentation Command (STRICOM). 2000. OneSAF Operational Requirements Document, Version 1.0.

U.S. Department of Defense. 1997. DoD modeling and simulation glossary. DoD 5000-59-M, December.

University of Central Florida. 1993. Distributed interactive simulation standards development: Operational concept 2.3. TRI ST-TR-93-25, Institute for Simulation and Training, Orlando, FL.

Voss, L.D. 1993. *A Revolution in Simulation: Distributed Interaction in the '90s and Beyond.* Virginia: Pasha Publications.

Wade, D.M. and Aronson, J. 1999. Model based simulation composition: A unified scenario model to support composable simulation. In *Proceedings of the Spring 1999 Simulation Interoperability Workshop*, 99S-SIW-044.

Weatherly, R.M., Wilson, A.L., Canova, B.S., Page, E.H., Zabek, A.A., and Fischer, M.C. 1996. Advanced distributed simulation through the aggregate level simulation protocol. In *Proceedings of the 29th Hawaii International Conference on Systems Sciences*, Vol. 1, pp. 407–415.

Weatherly, R.M., Wilson, A.L., and Griffin, S.P. 1993. ALSP—Theory, experience, and future directions. In *Proceedings of the 1993 Winter Simulation Conference*, pp. 1068–1072.

Weisel, E.W., Petty, M.D., and Mielke, R.R. 2003. Validity of models and classes of models in semantic composability. In *Proceedings of the Fall 2003 Simulation Interoperability Workshop*, 03F-SIW-073.

Weisel, E.W., Petty, M.D., and Mielke, R.R. 2005. A comparison of DEVS and semantic composability theory. In *Proceedings of the Spring 2005 Simulation Interoperability Workshop*, 05S-SIW-153.

Zeigler, B.P., Praehofer, H., and Kim, T.G. 2000. *Theory of Modeling and Simulation*, Second Edition. New York: Academic Press.

AQ3

QUERY FORM

CRC Press

Handbook of Dynamic System Modeling

JOURNAL TITLE:	**C5653**
ARTICLE NO:	**Ch016**

Queries and/or remarks

Query No.	Details Required	Authors's Response
AQ1	The sentence "Everything else. . .not all" is not clear. Please check.	
AQ2	Chandy and Misra (1978) is not listed in the reference list.	
AQ3	Chandy and Misra (1979); Fujimoto et al. (2002) and Zeigler et al. (2000) are not cited in the text.	