TECHNICAL ARTICLE

SIMULATION 73:5, 288-295

ISSN 0037-5497/99
Printed in the United States of America

Exploiting HLA and DEVS To Promote Interoperability and Reuse in Lockheed's Corporate Environment[1]

Bernard P. Zeigler
AI & Simulation Research Group
The University of Arizona
Tucson, AZ 86721
E-mail: zeigler@ece.arizona.edu
URL: www.acims.arizona.edu

Steve B. Hall
Lockheed Martin Missiles and Space
Sunnyvale, CA
E-mail: steve_hall@lmco.com

Hessam S. Sarjoughian
AI & Simulation Research Group
The University of Arizona
Tucson, AZ 86721
E-mail: hessam@ece.arizona.edu
URL: www.acims.arizona.edu

Joint MEASURE™ (Mission Effectiveness Analysis Simulator for Utility, Research and Evaluation) demonstrates the effective application of HLA-compliant simulation outside the traditional realm of training to analytic studies of mission effectiveness of systems within larger systems of systems configurations. This paper discusses how Joint MEASURE effectively exploits both the DEVS modeling and HLA simulation frameworks to support high performance distributed simulation and thereby, to overcome impediments to interoperability and reuse of Lockheed's models arising from the reluctance of groups to share their code at the source level with others within the corporation.

Keywords: Mission effectiveness simulation, DEVS formalism, High Level Architecture, interoperablility, site-proprietary impediments, data distribution management

1. Introduction

Lockheed Martin, one of the few large surviving aerospace conglomerates, is an aggregate of perhaps a hundred smaller entities, many of which were originally independent companies. This vestigial variety provides a treasury of modeling and simulation assets that offers a rich potential for reuse in new projects, especially systems-of-systems studies that require models of earlier developed components. However, interfacing simulations originally developed for disparate purposes, in variegated languages and platforms, represents a major challenge—one, of course, that the HLA standard is intended to address. Nevertheless, the developers of HLA, intent on solving the interoperability problem within the government, probably did not foresee a second impediment to interoperability that arises in an evolved corporate environment such as Lockheed's. This is the reluctance of subsidiaries to share their code at the source level with others within

[1] This work was supported by Advance Simulation Technology Thrust (ASTT) DARPA Contract N6133997K-0007.

the corporation—which we will refer to as the "site-proprietary" problem and elucidate further on. In this paper, we show how HLA and distributed simulation enable interoperability and reuse while eliminating the need to share source code. While critical to the success of Lockheed in marshalling its resources to meet the challenges of the new era of DoD budget allocations—and as one of the defense contractor giants, to the national interest, lessons learned in overcoming site-proprietary constraints are applicable to the larger commercial world where independent companies may wish to safeguard their intellectual property while interoperating their models without going the full distance of elaborate licensing arrangements.

2. Joint MEASURE™ Applications and Distributed Simulation Issues

In 1995 a DEVS-compliant simulation system, **Pleiades**, was developed by Lockheed Martin, Sunnyvale to enable analytic simulation studies of mission effectiveness of systems within larger systems of systems configurations. Recently (early 1999), Pleiades was renamed **Joint MEASURE** (**M**ission **E**ffectiveness **A**nalytic **S**imulator **U**tility **R**esearch **E**nvironment) after having been ported to execute on DEVS/HLA, an HLA-compliant distributed simulation environment supporting high-level development of DEVS models in C++ and Java. Joint MEASURE has been used to perform analysis on advanced surface ships, underwater vehicles and various sensor systems—underwater, terrestrial, airborne and space-based. The objectives in making Joint MEASURE HLA-compliant were to improve run-time performance by exploiting the parallel/distributed simulation supported by HLA. Enhanced performance is needed so that executing Monte Carlo runs of complex scenarios involving large numbers of entities could be accommodated. Further, the future could be seen to require support of the phases of the system development life cycle, as envisioned by the Simulation-Based Acquisition initiative. In particular, the migration of initially developed design models to real-time operation for system testing could be supported by exploiting the time management services of HLA in support of linking together federates operating on physical and logical time. Moreover, as mentioned and will be explained, site-proprietary issues limiting access to models owned by remote Lockheed sites could be overcome by the interoperability and information hiding afforded by HLA-distributed simulation. In sum, the goal of reducing simulation development costs could be furthered by HLA through its support of reuse of corporate models through interoperable distributed simulation.

We'll place these objectives in context by describing some current and planned applications.

- *SBL (Space-Based Laser)*—This is a study of alternative high-energy laser constellation architectures that is distinguished by the enormous volume of analysis it requires. Since the study is in the initial stages, there are a large number of alternatives to examine. (For example, how many space-based lasers, what orbits to use, what orbiting frequencies, whether to use relay mirrors, etc. In addition, there are combinations of space-based and ground-based sensors which significantly multiply the alternatives that need to be considered. Finally, there may be several alternative missions, besides the main one of detecting ballistic missiles, such as space control, ground-based surveillance, etc.) This large number of architecture-level alternatives, when multiplied by the number of experiments required to investigate each to an acceptable level of confidence, yields an enormous number of simulation runs. Such a workload demands that Joint MEASURE supplies the speed and high-performance possible with parallel/distributed simulation.
- *JCTN (Joint Composite Tracking Network)*—The objective is to analyze the utility of measurement-based fusion of sensor data obtained in real time from a variety of sources, including satellite-borne IR, reconnaissance aircraft, and Thadd. The questions address fundamental feasibility issues, such as: can composing such multiple sources be done fast enough and with sufficient accuracy and quality to be useful for a missile interception? Developing credible answers requires access to remote Lockheed Martin models and authenticated data, such as from the Aegis, SBIRS, and Thadd. The HLA-compliant distributed simulation of Joint MEASURE enables federating such models together into a usable composite.

We will now present a brief review of DEVS/HLA to provide some background for the discussion of the role of HLA and Joint MEASURE in overcoming barriers to model reuse and interoperability in Lockheed's corporate environment.

3. Review of DEVS/HLA

DEVS/HLA [1] is an HLA-compliant modeling and simulation environment formed by mapping the DEVS-C++ system [2] to the C++ version of the DMSO RTI. DEVS is a mathematical formalism for expressing discrete event models which supports discrete event approximation of continuous systems and has an object-oriented substrate supporting model implementation and repository reuse. Advantages of the DEVS methodology for model development include well-defined separation of concerns supporting distinct modeling and simulation layers that can be independently verified and reused in later combinations with minimal re-verification. The resulting divide-and-conquer approach can greatly simplify and accelerate model development, leading to greater credibility at reduced effort. DEVS has a well-defined concept of

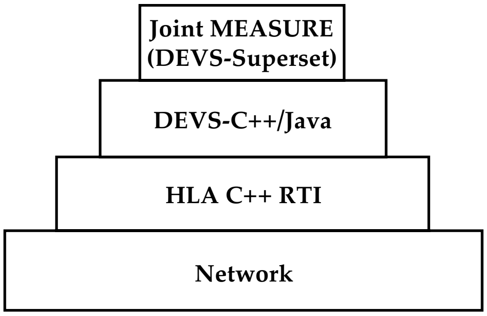


Figure 1. DEVS/HLA layered architecture and its support of supersets such as Joint MEASURE

system modularity and component coupling to form composite models. It enjoys the property of closure under coupling which justifies treating coupled models as components and enables hierarchical model composition constructs.

Such formal properties of the DEVS methodology enable DEVS/HLA to support high-level federation development and execution. Modularity, in which component models are coupled together through input/output ports, allows messages to be sent from one federate to another using the underlying HLA interaction messages. In addition, DEVS/HLA supports attribute updating and quantization-based message filtering [3]. For this, the modeler declares HLA objects and attributes desired for reflecting states of DEVS federates. By attaching quantizer objects supplied by DEVS/HLA to such attributes, the appropriate publish/subscribe mechanisms are automatically set up. The resulting environment for high-level federation development and simulation is shown in Figure 1. Models developed in a DEVS/C++ or DEVSJAVA (C++ and Java-based DEVS implementations, respectively) can be directly simulated in the DEVS/HLA environment over any TCP/IP, ATM, or other network of hosts executing an HLA C++ RTI. Based on model-supplied information, DEVS/HLA takes care of the declarations and initializations needed to create federations, joining and resigning of federates, communication among federates and time management.

3.1 Joint MEASURE

Joint MEASURE was developed atop DEVS/HLA by Lockheed Martin, Sunnyvale. Simulation-based acquisition requires that early in the development of a new defense system there be an assessment of its expected mission effectiveness. The benefit aspect of this cost/benefit analysis examines how well the proposed system would contribute to the nation's overall ability to counter an enemy threat. This contribution is a result of the system's ability to gather and share information, survive, and/or service hostile targets. Unfortunately, making this assessment can be difficult. Subtle design decisions can result in significant impacts on effectiveness. Similarly, modest alterations in the way the system is used can have a marked impact on effectiveness, as can relatively minor modifications of the scenario that the system is immersed within and is reacting to.

Joint MEASURE (JM) is intended to assist in evaluating a proposed defense system's mission effectiveness in the context of a specified set of possible design

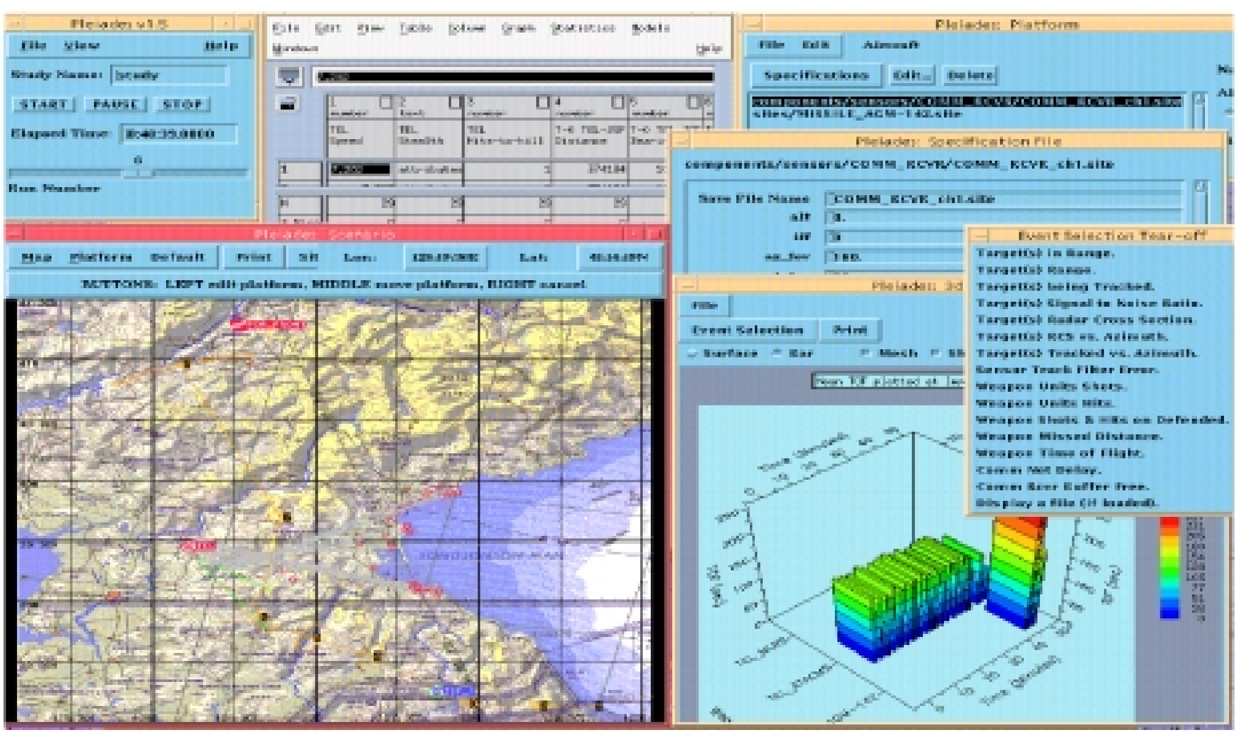


Figure 2. Joint MEASURE screen shot with GIS-based map displayed along with dynamic data output

variations, operational utilization patterns and engagement scenarios. Examples of such defense system proposals will be given later in the paper. Joint MEASURE contains a library of sensor, weapon, and command/control communication platform models that can be composed to model military systems of systems operating in physically realistic environments. It is intended to support Monte Carlo and/or optimization simulation runs involving large numbers of runs with different random number seeds and parameter settings. JM contains Event Distribution and Predictive Encounter controllers to efficiently manage interactions among mobile, communicating platforms. As shown in Figure 2, JM contains a visualization interface to display moving platforms that are spatially referenced within a Geographical Information System (GIS). It also contains a Data Modeling and Analysis Toolkit to support statistical analysis of simulation output. In JM, experiments can be specified interactively, to include attributes and behaviors of system components. Such scenarios can be stored, retrieved and concatenated to form new composites. JM has been in development for five years and includes approximately 200K lines of internally developed C++ code in addition to its substantial utilization of available libraries.

3.2 *DEVS Compliance of Joint MEASURE*

Lockheed Martin made an early decision to comply with the DEVS formalism due to its formal precision and event-based simulation efficiency. This suggested its capability to support speed, repeatability, code maintainability/reusability, and high-level portability. We discuss these requirements individually:

- **Speed**: It is easy to generate a requirement to simulate a multi-hour analysis scenario thousands of times. For example, if we assess three alternatives to each of three platforms' design parameters (e.g., speed, signature and vulnerability) and three alternatives to three utilization patterns (e.g., distance from enemy, friendly deployment configuration and friendly reactivity) and three alternatives to three scenario characteristics (geographical location, enemy deployment configuration and enemy reactivity) and repeat each of these configurations 20 times (due to the probabilistic nature of the simulation), then we will need to run nearly 400,000 simulations of this multi-hour scenario. If we can run our simulator 1,000 times faster than real time, it will still take a month to finish the analysis. The developed architecture makes it possible to step through time when necessary and selectively jump over intervals when useful on a selective (platform-by-platform) basis.
- **Repeatability**: Since it is based on DEVS, a logical model formalism with a well-defined state concept, Joint MEASURE has the ability to exactly replicate simulation runs given the same initial state (which could include pseudo-random number seeds). Moreover, the architecture, when mapped into DEVS/HLA, can improve performance without risk of losing repeatability or inducing discrepancies since the temporal ordering of events is guaranteed by the Parallel DEVS protocol used in DEVS/HLA when running over a distributed network.
- **Maintainability/Reusability**: The modular aspect of DEVS models enforces a software engineering practice that simplifies code reuse and simplifies the construction of new federations from existing components.
- **Portability at a High Level**: The first DEVS simulation environment used was DevSim++ [4], as a single-processor implementation of the DEVS formalism. The high level of abstraction provided by the DEVS formalism subsequently facilitated porting the system to DEVS/C++ and later to DEVS/HLA.

3.3 *Component Models*

Based on DEVS concepts, Joint MEASURE defines four basic kinds of coupled[2] models: Generators and Transducers, (forming the Experimental Frame Module) and Platforms and Space Model. This architecture is designed to exploit the fact that often nothing interesting is happening during some interval of a simulated scenario. By exploiting this fact we can produce significant performance gains beyond those achieved by the standard use of a discrete event simulator. Each of these classes of models is described briefly below. The Space Model is then discussed in somewhat more detail.

Generator

The Generators (one of the components of the Experimental Frame) are relatively simple models in Joint MEASURE. Only one Generator can be instantiated in a given simulation. The Generator is responsible for instantiating platform models and interfacing with models outside the scope of Joint MEASURE.

Transducer

The Transducer model (the other component of the Experimental Frame) is responsible for gathering data, performing simple analyses, generating dynamically updating data displays, and creating and updating analysis databases for later detailed statistical and mathematical examination.

The runtime data display sets are graphically updated in 3D at the end of every N simulation runs. This capability has proven to be especially useful for early detection of scenarios that have been specified incorrectly or have not been specified so as to produce "interesting" results, e.g., when none of the Platforms

[2] A coupled model in DEVS is a model that encapsulates other models.

ever detects a Platform with an opposing affiliation.

The databases are generated in a generic format that is readable by multiple tools such as PROPHET, a statistical visualization module developed for life science research which has useful tools for examining characteristics such as "survivability" on top a robust set of standard statistical and modeling tools.

Platforms

Platforms represent the systems that are to be evaluated. Platforms have a location in the world; they have extent; they can be collided with; and they are potentially detectable. Platforms may also actively or passively sense the world around them; move their location; generate emissions of various sorts; launch sub-platforms (e.g., missiles); participate in physical transfer operations; and fail for a variety of reasons. Multiple Platforms can be instantiated in a given simulation. The standard Platform model is a coupled model containing three singly instantiated (coupled) models: the Hull model, the Global Status model and the Subsystems model.

The Hull coupled model models the structure of the platform. It knows about the relative locations of the subsystems onboard the platform, and routes the transfer of material and data to them and between them with the appropriate temporal delays. It also models the impact of collisions and detonations on the viability of the platform itself and on the conduits that connect the various subsystems.

The Global Status coupled model generates and maintains a model of the aggregated appearance (response to other platforms' emissions, e.g., IR signature) and capability of the platform for the consumption of the world beyond the platform. That is, what does this platform look like and sound like at the current time, and what are its current capabilities (e.g., could it detect a radar burst from another platform)? The Space Model, discussed below, is the primary consumer of this data.

The Subsystem coupled model models the various subsystems on board the Platform. The standard Subsystem model is composed of the following six (coupled) models: a Sensor-Transceivers model; a Weapon-Launchers model; a C3I model; a Stores model; a Mechanical model; and the Subsystem Router model.

The Sensor-Transceivers component includes all the sensors and active emitters on board the platform. The Weapon-Launchers component includes all the weapon systems on board the platform. The C3I model models the decision processes on board the Platform. The Stores model models the renewable and consumable resources of the platform. The Mechanical model models the propulsion, maneuvering and effector capabilities of the Platform. Finally the Subsystem Router model is a software device to facilitate the interchangeability of various Subsystems models.

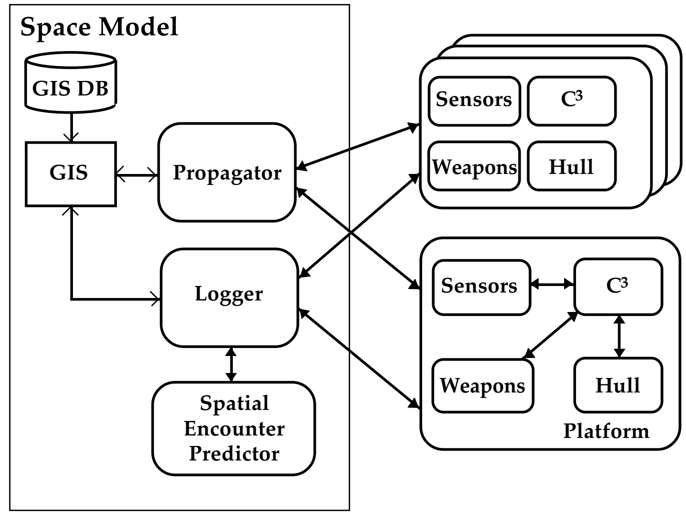


Figure 3. Joint MEASURE Architecture

Space Modeling Infrastructure

JM includes an intelligent model of the "space" enclosing the platforms. This Space Model performs two important functions. First, it infers when interesting encounters will occur among any number of moving platforms in a pair-wise manner and provides such spatial encounter prediction services to the Platforms, resulting in significant simulation performance improvement. Second, it determines the propagation losses and delays of various emissions and how much noise the environment will add to a signal. It then propagates the emissions of a Platform to other "within-range" Platforms capable of receiving them. These two functions are localized in components of the Space Model as illustrated in Figure 3. Additionally, to maintain knowledge of Platform whereabouts, the Space Model includes a Logger component.

Spatial Encounter Prediction

The Space Model predicts interesting spatial relationship events and informs the various Platforms so that they are not unnecessarily attempting to interact. In order to make these predictions, the "trajectory" of a platform must be known by the Space Model. The Spatial Encounter Prediction (SEP) component currently knows about the following kinds of trajectories: Rhumb line (constant heading) with constant elevation changes; Great Circle (shortest distance) with constant or ballistic elevation changes and orbital trajectories (without atmospheric drag). The Platform is responsible for updating the Logger anytime the Platform's trajectory parameters change.

The SEP can produce a number of spatial encounter predictions with what it knows about any two Platform trajectories. These predictions currently include: the time at which they will be at their closest point of approach (potentially a collision) and the first and last time they will be: "over the horizon;" within a given distance; within a given azimuth angle; or within a given elevation angle. It can also make optimistic estimates of the probability of detection and inform a sensor when that probability is exceeded. The Space Model also predicts at what point in time, if at all, a Platform will "run aground." Platforms subscribe to encounters of interest and are informed of them at the appropriate point in time.

When Platforms exploit this information effectively, significant simulation efficiencies are achieved as a consequence of jumping over dead time intervals (when no Platforms are interacting) and by limiting which Platforms are time stepping to those that are potentially interacting.

Emission Propagation

The Emissions Propagator component of the Space Model propagates emissions with their appropriate propagation losses, delays and associated background noises. Platforms are limited to two forms of interaction: physical transfers (e.g., supplies or fuel) and emissions. Emissions cover pressure waves from explosions, acoustic noise (self-generated and reflected), and electromagnetic emissions (self-generated and reflected). The model allows emissions to carry semantic content as appropriate, for example, in radio communications.

Propagation loss (i.e., how much of the source signal strength remains when the signal arrives at the destination's location) is a function of both the distance traveled and the nature of the environment. Both factors (distance and environment) similarly influence propagation delay (i.e., how long it takes for the signal to transit from the source to the destination). And finally noise (in which the signal is embedded) is a function not only of the environment, but also of the other Platforms in the region. These delays (especially in the acoustic realm) and losses can be significant. Furthermore, the environmental influences themselves can be dynamic. For example, wind, rain and fire can significantly change the nature of the medium that is being transited through. For these reasons the Space Model includes as a component, a model of the environment (integrated with a GIS system). Emissions received by the Space Model are passed on (when transmission is not blocked by terrain[3]), at the appropriate point in time (with the signal strength diminished and embedded in noise[4]), to all the "eligible" Platforms. Eligible Platforms are those that have functioning receivers (of the right sort) that conservatively might detect the signal.

4. Overcoming Site-Proprietary Impediments to Interoperability with Joint MEASURE

As already mentioned, Lockheed Martin must be able to marshal the models and simulations of its once-independent groups to synthesize composites that can answer emerging system-of-system development issues such as outlined earlier. Although now part of a larger corporation, such groups are often reluctant to share their source code with peers.[5] In the absence of a distributed simulation solution, this "site-proprietary" disposition can severely impede the integration of codes from different areas of the corporation. In the following we examine a case study to illustrate this point.

4.1 Lockheed's DD21/Deepwater Project

The major Navy and Coast Guard ship design programs, DD21 and Deepwater, respectively, require extensive analysis of system-by-system options. Originally, the Medusa simulator was developed by the GES company to study the Aegis warship's ability to detect and intercept missile threats. However, the model considered only a single missile attack on a motionless ship. Subsequently, the need arose to embed the ship's sensor in a more realistic environment of multiple missiles, ship's motion, and air and sea emission propagation effects. The shortest route to achieving this embedding was to couple Medusa with Pleiades, the non-HLA compliant predecessor to Joint MEASURE. However, reluctance to share source code (even though GES was now incorporated within Lockheed Martin) made this coupling difficult to achieve in a completely satisfactory manner. One approach that was taken was to replace Medusa with a table lookup to yield the probability of missile kill under

---

[3] We do line-of-sight obstruction checking, which includes both the curvature of the Earth and the influence of the terrain. The terrain check uses either 30-second data (from the GLOBE (Global Land One-km Base Elevation) database from the National Geophysical Data Center or the ETOPO5 five-minute data. If the 30-second data is missing for a given region, we automatically use the five-minute data. We assume a spherical Earth, but could approximate a spheroid without much additional work.

[4] On water, radar noise is calculated as a function of Sea State. On land, it is calculated as a function of vegetation type, urbanization and land hilliness. Wind effects are a planned enhancement. Laser energy propagation is diminished not only by the distance traveled, but by the atmosphere absorption (via the FASCODE model). Thermal blooming will be added to the equation in the near future. Currently, we are in the process of integrating acoustic propagation models as well as the SSGM (Synthetic Scene Generation Model) database to support infrared signature and noise models.

[5] Reluctance to share model source code may derive from several reasons. The customer, whose system is modeled, may rightly be concerned about the potential for indiscriminate dissemination of the code within the corporation, unauthorized use, and claims made based upon model application outside its domain of validity. The model developer may be rightly concerned about the loss of ownership control of the source in the event that the group finds itself sold into another corporation.

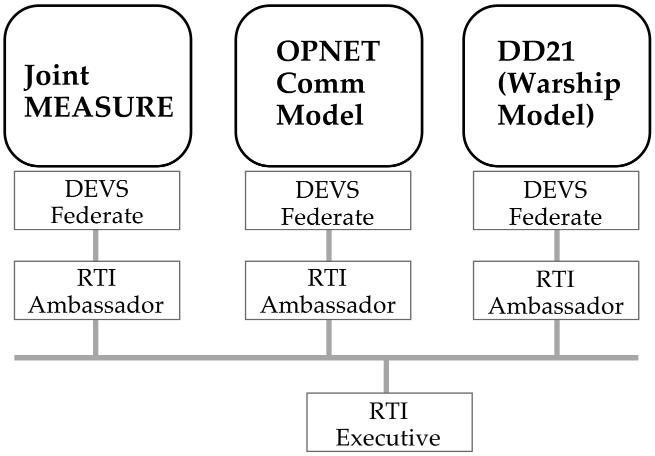


Figure 4. DEVS /HLA supporting integration of heterogeneous distributed components

the given situation arising in the combined model. To generate a table attempting to cover all situations that might arise in the coupled model, a thousand cases were employed (missiles starting from some 10 distances from the ship, approaching at some 10 azimuth angles, and 10 loading conditions (numbers of other active missiles)). Each case was executed 100 times in Medusa to obtain an acceptable level of confidence in the probability estimate. This effort foundered due to the time-consuming nature of the table generation and the fact that at this resolution level, it could not accurately represent the actual result of interaction between Medusa and Pleiades.

HLA and distributed simulation provides a means to bypass such site-proprietary problems. To develop the appropriate FOM and SOM interfaces for the model components requires the collaboration of their developers but can be done without access to source code. Models can be compiled locally with appropriately programmed HLA interfaces so that model source and object code can remain at the owners' sites. Federations can't be established and executed without the consent of each federate developer and the authorization of the customer. So models remain fully under the control of their developers on the one hand and customers on the other.

Developing and executing federations, however, is not necessarily a routine process. Indeed, a methodology, expressed in the FEDEP (Federation Development and Execution Process) is being developed to facilitate it. To interoperate properly, developers must understand both the data exchange requirements and the dynamics of the components to be federated. The DEVS formalism provides a means to express dynamics in a standardized manner. In particular, the DEVS Bus concept provides a basis for wrapping non-DEVS models within DEVS components, enabling DEVS/HLA to support integrating legacy or other simulation software. In application to DD21 and Deepwater development, Joint MEASURE has been coupled to an OPNET model of an inter-satellite communication network. To complete the federation, these federates will be coupled to a warship model that extends the Medusa legacy simulation code.[6] Figure 5 illustrates our approach to such integration of heterogeneous model components—where each is represented by a DEVS federate in DEVS/HLA, that is, each is either a DEVS model or wrapped within a basic DEVS interface.

In such federations, owners of components can retain complete control of their models since they remain at their own sites. Further, provided that the appropriately specified SOM, FOM and DEVS interfaces enable proper coupling of components, the latter can be inter-operated at the object code level without requiring source code integration. This obviates the thorny issue of site-proprietary concerns of participating Lockheed groups.

An issue to be addressed in developing distributed simulations with federate components located in various parts of the country or the world is how to reduce significantly enormous numbers of updates and messages to a manageable quantity given limited network bandwidth. Predictive contract and other message filtering methods are intended to overcome the bandwidth and propagation latencies introduced by the connecting wide-area network, e.g., the Internet [3].

5. Conclusions

Joint MEASURE exploits the advantages of DEVS and HLA to support the high performance and accuracy demands of analytic mission effectiveness studies for complex systems. Its distributed simulation capability provides an end run around site-proprietary issues of concern to participating groups in an evolved corporate environment such as Lockheed's. JM is being extended to address migration of simulation models to real-time operation and other life cycle support functions needed for Simulation-Based Design. One key to meeting speed performance requirements is the computation efficiency afforded by spatial encounter prediction and its ability to reduce communication events among mobile, interacting platforms without impacting simulation fidelity. Also required for overcoming the network bottleneck are the predictive contract and other message reduction mechanisms afforded by the HLA and DEVS/HLA layers within JM. Unfortunately, because of the possibility of unpredictably jumping over intervals of time, hardware and man-in-the-loop simulation are not currently possible with Joint MEASURE. However, the emergence of commercial real-time RTIs together with the ability to execute models

[6] The current status of these projects is as follows: the Deepwater work is real and operational. It deals with scenarios that have thousands of objects. The simulation supports the Coast Guard in evaluating alternative strategies for monitoring, coordinating and interdicting suspicious sea traffic. The DD21 federate is currently under development.

in real time [5, 6] could provide a development path for real-time execution of DEVS models. Eventually, this would enable Joint MEASURE to support all life cycle phases, especially design and test, of Simulation-Based Acquisition. For additional information, please see the Website at *www.acims.arizona.edu*.

6. References

[1] Zeigler, B.P., et al. "Implementation of the DEVS Formalism over the HLA/RTI: Problems and Solutions." *Fall SIW*, Orlando, FL, 1999.

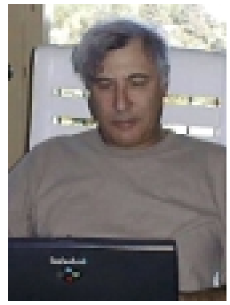
[2] Zeigler, B.P., et al. "The DEVS Environment for High-Performance Modeling and Simulation." *IEEE C S & E*, Vol. 4, No. 3, pp 61-71, 1997.

[3] Zeigler, B.P., et al. "Bandwidth Utilization/Fidelity Tradeoffs in Predictive Filtering." *Fall SIW*, Orlando, FL, 1999.

[4] Hong, G.P. and Kim, T.G. "A Framework for Verifying Discrete Event Models Using a Dual Specification Approach." *TRANSACTIONS of the Society for Computer Simulation International*, Vol. 13, No. 4, pp 19-34, 1996.

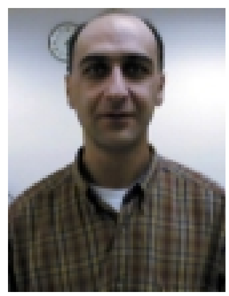
[5] Zeigler, B.P., et al. "Distributed/Real Time Simulation: Integrating Legacy and Modern Codes." *Int'l Conf. on Env. Sys., Soc. Aut. Engr*, Danvers, MA, 1998.

[6] Hong, J.S. and Kim, T.G. "Real-time Discrete Event System Specification Formalism for Seamless Real-time Software Development." *Discrete Event Dynamic Systems: Theory and Applications*, Vol. 7, No. 4, pp 355-375, 1997.



Bernard P. Zeigler is Professor of Electrical and Computer Engineering at the University of Arizona, Tucson. He has written several foundational books on modeling and simulation theory and methodology. He is currently leading a DARPA-sponsored project on DEVS framework for HLA and predictive contracts. He is a Fellow of the IEEE.

Steven B. Hall is a Senior Technical Manager at Lockheed Martin Missiles and Space Corporation, Sunnyvale, California.



Hessam S. Sarjoughian is Assistant Research Professor of Electrical and Computer Engineering at the University of Arizona. His current research interests are in theory, methodology, and practice of distributed/collaborative modeling and simulation.