High Level Architecture for Simulation: An Update

Dr. Judith S. Dahmann
Chief Scientist
Defense Modeling and Simulation Office

Ms. Katherine L. Morse
Senior Computer Scientist
Science Applications International Corporation

Abstract

The High Level Architecture (HLA) provides the specification of a common technical architecture for use across all classes of simulations in the US Department of Defense. It provides the structural basis for simulation interoperability. The baseline definition of the HLA includes the HLA Rules, The HLA Interface Specification, and the HLA Object Model Template. The HLA Rules are a set of 10 basic rules that define the responsibilities and relationships among the components of an HLA federation. The HLA Interface Specification provides a specification of the functional interfaces between HLA federates and the HLA Runtime Infrastructure. The HLA OMT provides a common presentation format for HLA Simulation and Federation Object Models.

This paper provides a description of the development of the HLA, a technical description of the key elements of architecture, and a discussion of HLA implementation, including HLA supporting processes and software.

1. Introduction

The High Level Architecture (HLA) is an architecture for reuse and interoperation of simulations. The HLA is based on the premise that no simulation can satisfy all uses and users. An individual simulation or set of simulations developed for one purpose can be applied to another application under the HLA concept of the federation: a composable set of interacting simulations. The intent of the HLA is to provide a structure which will support reuse of capabilities available in different simulations, ultimately reducing the cost and time required to create a synthetic environment for a new purpose and providing developers the option of distributed collaborative development of complex simulation applications.

The HLA has wide applicability, across a full range of simulation application areas, including education and training, analysis, engineering and even entertainment, at a variety of levels of resolution. These widely differing application areas indicate the variety of requirements that have been considered in development and evolution of the HLA.

The definition of architecture used in this effort -- "major functional elements, interfaces, and design rules, pertaining as feasible to all simulation applications, and providing a common framework within which specific system architectures can be defined" -- is one that is commonly accepted and is consistent with the IEEE definition of architecture for computer simulations. For the purpose of this effort, the emphasis is on the development of a high level architecture which pertains as widely as possible to all simulation areas and which will provide a framework for the development of specific system architectures.

The HLA does not prescribe a specific implementation, nor does it mandate the use of any particular set of software or programming language. Over time, as technology advances become available, new and different implementations will be possible within the framework of the HLA. This paper will describe implementation experience with HLA.

2. HLA development process

The HLA was developed based on a process involving Government, academia, and industry. In 1995, three industry teams developed concepts for the definition of a high level architecture. The results of these efforts were combined along with additional insights from other modeling and simulation projects to arrive at the initial definition of the HLA. On 31 March

The AMG developed the architecture based on cooperative efforts to apply the HLA in a series of prototypes designed to ensure the HLA addresses the broad set of application requirements. The result was a baseline HLA definition, completed in August 1996. The AMG has continued to evolve the HLA based on experience with its use, reviewing and updating the HLA specifications on a six-month cycle. In February 1998, version 1.3 of the HLA specification was adopted ([1], [2], [3]). These specifications form the basis for a draft IEEE standard simulation interoperability architecture ([4], [5], [6]). The HLA is already a standard for use in the US Department of Defense ([7]), has been nominated for standardization in NATO ([8]), and is in discussion by the Object Management Group (OMG).

3. Technical architecture

3.1. Functional overview

Figure 1 shows how an HLA federation partitions into its major functional components. The first key component are the simulations themselves, or more generally, the federates. A federate can be a computer simulation, a manned simulator, a supporting utility (such as a viewer or data collector), or even an interface to a live player or instrumented facility. All object representation is in the federates. The HLA imposes no constraints on what is represented in the federates or how it is represented, but it does require that all federates incorporate specified capabilities to allow the objects in the simulation to interact with objects in other simulations through the exchange of data supported by services in the Runtime Infrastructure (RTI).

1995, this definition was presented to an Architecture Management Group (AMG) formed to develop the HLA from this initial definition.

The second functional component is the RTI. The RTI is, in effect, a distributed operating system for the federation. The RTI provides a very general purpose set of services that support the simulations in carrying out these federate-to-federate interactions and federation management support functions. These services will be discussed in a subsequent section. All interactions among the federates go through the RTI.

The third functional component is the runtime interface. The HLA runtime interface specification provides a standard way for federates to interact with the RTI, to invoke the RTI services to support runtime interactions among federates and to respond to requests from the RTI. This interface is implementation independent and is independent of the specific object models and data exchange requirements of any federation.

Two other general capabilities of simulation systems are supported by the architecture. First, the HLA supports the passive collection of simulation data and monitoring of simulation activities. In the HLA, these tools act in the same way as simulations and interact with the RTI using the HLA interface.

Second, the HLA supports interfaces to live participants, such as instrumented platforms or live systems. Live participants interact with the simulated world through something that acts like a simulation from the point of view of the HLA, that feeds a representation of the live world into the simulated world and that projects data from the simulated world back to the live system.

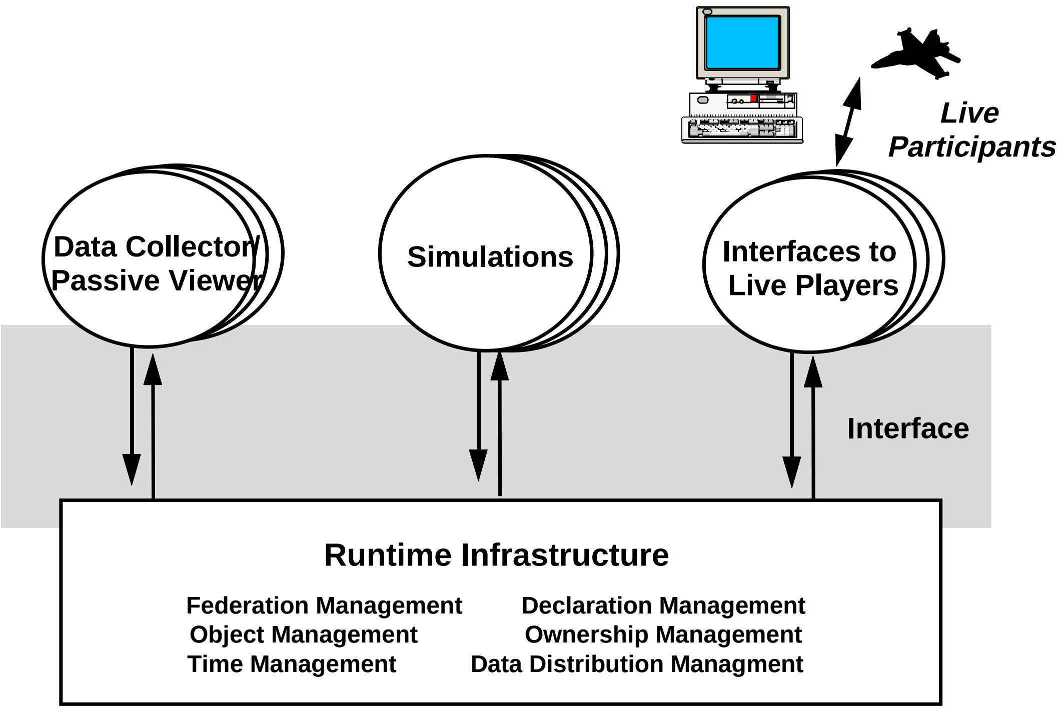

Figure 1. Functional view of an HLA federation

While figure 1. shows the logical relationship among the components of the architecture, it should not be taken as reflecting implementation. Figure 2. displays the distributed RTI implementation approach taken in the DMSO development of RTI version 1.3. The HLA does not prescribe implementation characteristics of an RTI; different implementors will follow different design approaches.

3.2. HLA interface specification

The HLA interface specification ([1]) describes the runtime services provided to the federates by the RTI, and by the federates to the RTI. There are six classes of services. *Federation management* services offer basic functions required to create and operate a federation. *Declaration management* services support efficient management of data exchange through the information provided by federates defining the data they will provide and will require during a federation execution. *Object management* services provide creation, deletion, identification and other services at the object level. *Ownership management* services support the dynamic transfer of ownership of object/attributes during an execution. *Time management* services support

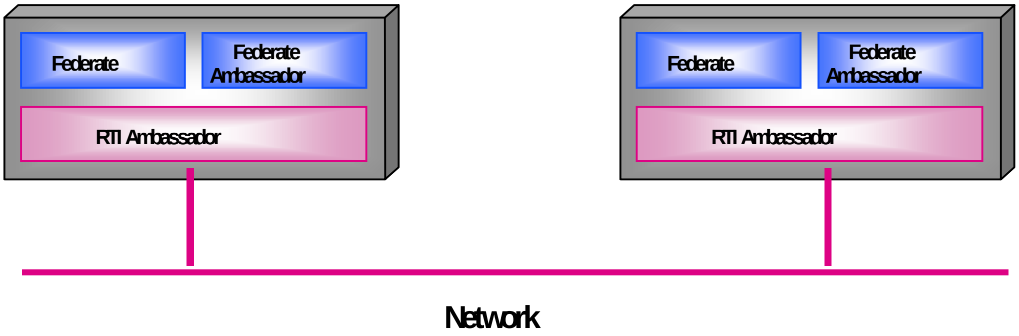

Figure 2. Distributed RTI Implementation Approach

synchronization of runtime simulation data exchange. Finally, *data distribution management* services support the efficient routing of data among federates during the course of a federation execution. The HLA interface specification defines the way these services are accessed, both functionally and in an application programmer's interface (API). APIs in CORBA IDL, C++, Ada, and Java are incorporated in the interface specification.

3.3. HLA object models

HLA object models are descriptions of the essential sharable elements of the simulation or federation in 'object' terms. The HLA is directed towards interoperability; hence in the HLA, object models are intended to focus on description of the critical aspects of simulations and federations, which are shared across a federation. The HLA puts no constraints on the content of the object models. The HLA does require that each federate and federation document its object model using a standard object model template ([2]). These templates are intended to be the means for open information sharing across the community to facilitate reuse of simulations. These completed templates will be openly available, and tools are being developed to allow for automated search and reasoning about object model template data, to further facilitate cost-effective information exchange and reuse.

The HLA specifies two types of object models: the HLA Federation Object Model (FOM) and the HLA Simulation Object Model (SOM). The HLA FOM describes the set of objects, attributes and interactions, which are shared across a federation. The HLA SOM describes the simulation (federate) in terms of the types of objects, attributes and interactions it can offer to future federations. The SOM is distinct from internal design information; rather it provides information on the capabilities of a simulation to exchange information as part of a federation. The SOM is essentially a contract by the simulation defining the types of information it can make available in future federations. The availability of the SOM facilitates the assessment of the appropriateness of the federate for participation in a federation.

While the HLA does not define the contents of a SOM or FOM, it does require that a common documentation approach be used. Both the HLA FOM and SOM are documented using a standard form called the HLA Object Model Template (OMT).

3.4. HLA rules

Finally, the HLA rules ([3]) summarize the key principles behind the HLA. The rules are divided into two groups: federation and federate rules. Federations, or sets of interacting simulations or federates, are required to have a FOM in the OMT format. During runtime, all object representation takes place in the federates (not the RTI) with only one federate owning any given attribute of an instance of an object at any given time. Information exchange among the federates takes place via the RTI using the HLA interface specification.

Additional rules apply to individual federates. Under the HLA, each federate must document their public information in their SOM using the OMT. Based on the information included in their SOM, federates must import and export information, transfer object attribute ownership, updates attributes and utilize the time management services of the RTI when managing local time.

4. HLA supporting processes

While HLA is a runtime architecture for simulation interoperability, increasingly attention is being devoted to developing views on the processes, which surround the use of HLA. In particular, the HLA federation development and execution process (FEDEP) has been developed and is being evolved based on user experience with the application of HLA. In this view the process covers five basic steps: concept development, federation design, federation execution implementation, testing, and operations. The basic elements of these steps are then articulated in more detail to provide a concrete, albeit abstract, perspective on the activities and processes which constitute the end-to-end process. The FEDEP is updated every six months, based on growing user experience with HLA. The FEDEP is depicted in figure 3.

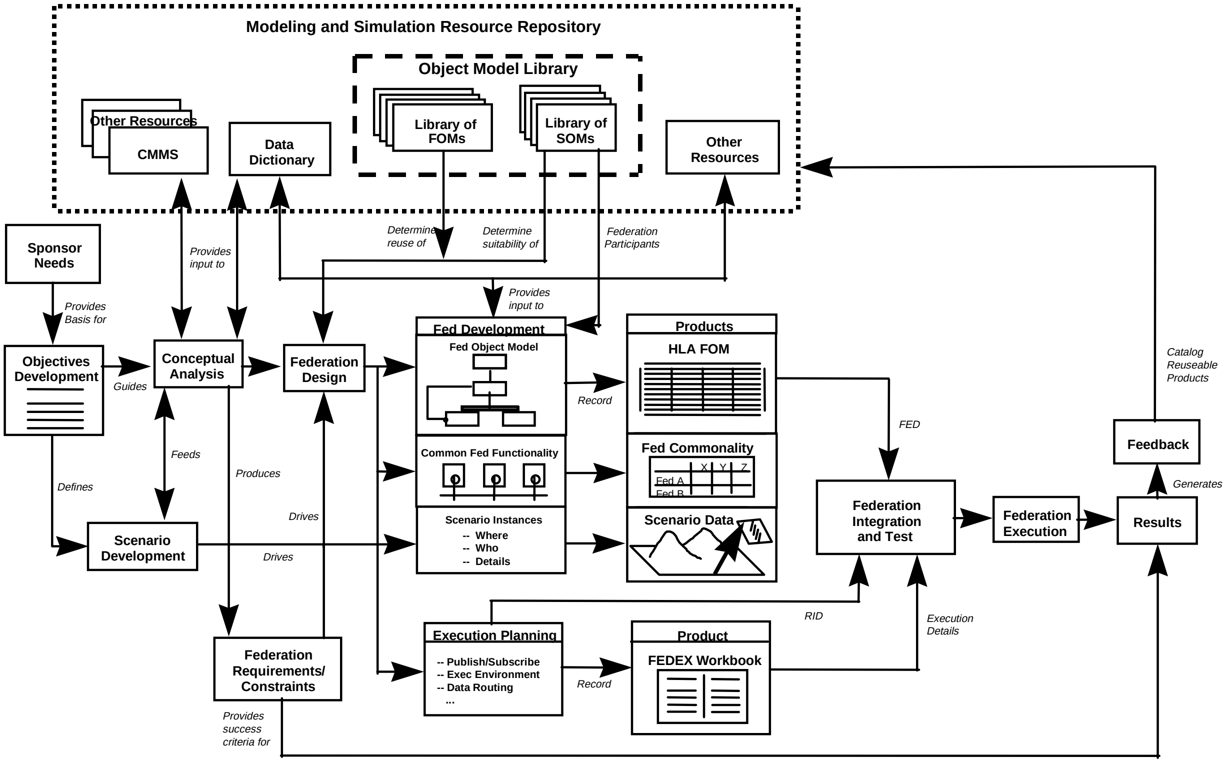

Figure 3. Federation Execution and Development Process

Several groups who are addressing specific aspects of the use of HLA have used the FEDEP. These include those who are responsible for verifying and validating the representations in a federation of simulations ([9]) and users of the HLA whose applications operate in a secure environment ([10]). In both cases, the FEDEP provides a general-purpose framework for identification and description of special actions required in the particular domain of interest.

5. HLA supporting software

The FEDEP has also served as the basis for the development of a tool architecture ([11], [12], [13], [14], [15], [16]). From the outset it has been recognized that the utility of the HLA will be based on the extent to which cost-effective supporting software for the creation, development and operation of HLA federations becomes available to HLA users. In the eighteen months since the adoption of the baseline HLA, examples of HLA supporting software have become available and, in some cases, are in broad use across the user community.

5.1. RTI software

RTI software was initially developed as part of the prototyping to support HLA baseline definition. This initial RTI software build, known as the 'X series' was developed with an IDL API using CORBA. With the acceptance of the HLA baseline, the 'X series' RTI software was retired and the 1.0 series development began. The 1.0 series was developed in C++, and supported the other APIs through bindings or software caps. The first release, known as the familiarization release (F.0), came out in December 1996. It supported most of the HLA services except certain federation management control functions and data distribution management (DDM) services. The next RTI 1.0 release, in May 1997, supported all but the DDM service group. DDM services were the subject of a large scale HLA experiment, called the Synthetic Theater of War (STOW), in which RTI software, with the objective of scalability, was prototyped. The STOW exercise supported over 300 federates and 5000 objects operating in perceptible real-time. The STOW prototype software and experimentation supported the evolution of the HLA specification in the DDM service area and has been incorporated into RTI 1.3, the first full service RTI implementation, supporting HLA specification 1.3, released in March 1998. This RTI software is in the public domain and can be accessed via the web http://www.dmso.mil/RTISUP/hla_soft/hla_soft.htm. Currently over 700 users worldwide have downloaded this software, with about 30 percent of these users being from outside of the United States.

As part of HLA experimentation, RTI 1.0 was reimplemented in Java in an effort to demonstrate the robustness of the architecture to new implementation approaches and to HLA use for web-based applications. The Java RTI 1.0 version successfully interoperated with federates using RTI 1.0 implemented in C++. Further experiments are underway with RTI 1.3 to support federates operating in shared memory and multiprocessor environments, again to ensure the flexibility of the HLA to support a range of implementation options.

A higher performance, production RTI implementation is in progress to replace the current 1.3 RTI. This software is expected to be available for broad release by the end of calendar year 1998. Other commercial- and university-based RTI developments are underway in both the United States and Europe. These support the HLA interface specification, but vary in other aspects of their designs and implementation.

5.2. Object model tools

As the HLA definition dictates, users of HLA will be developing object models for their simulations, as well as for other federates and their federations. While straightforward in concept, HLA object models can be cumbersome to develop and share as they grow in size and complexity. To aid in this process automated tools to develop, share, and fill object models have been developed. In particular, there are three components in an OM Tool set ([12], [13]) available in the public domain:

5.2.1. Object Model Development Tool Automated support for development of HLA Object Models (OMs), generation of RTI federation execution data, and exchanging OMs with the Object Model Library http://hla.dmso.mil/RTISUP/hla_soft/hla_soft.htm

5.2.2. Object Model Library

Web accessible library of Federation Object Models (FOMs) and Simulation Object Models (SOMs). http://www.omlibrary.epgc4i.com/

5.2.3. Object Model Data Dictionary

Web accessible repository of common data components for OM development mapped to DoD data standards, having linking capability with Object Model Development Tools. The Object Model Data Dictionary is currently undergoing testing prior to its release.

Again these tools are publicly available via the web http://www.dmso.mil/RTISUP/hla_soft/hla_soft.htm . Over 200 users have accessed these since they were first released in the fall of 1997. These tools are supported by a set of standard data interchange formats (DIFs) which allow for the exchange of object models and object model data fill. ([17]). These DIFs have allow for the development of commercial tools to meet these and other HLA object model development functions tailored to the needs of particular HLA users. Several companies in the US and overseas have plans to market comparable tools which use the DIFs for data exchange.

5.3. Federation execution planning and runtime support tools

Automation is also being applied to the process of planning federation execution and to the testing and monitoring of federation runtime operations. A federation execution planners workbook has been developed ([18], [19]) which provides a structured approach to compiling the data need to configure an HLA federation execution. An automated toolset to support the development and exchange of completed workbooks is in development, along with supporting DIFs to support the sharing of this data with other related tools. Tools supporting the verification that federates are appropriately meeting their commitments to the FOM are being developed as are runtime data collection and monitoring tools. These runtime tools monitor the progress of the federation using the HLA interface specification, and hence can be used for different federations.

5.5 Software to support development or adaptation of simulations to work with HLA

As experience with HLA grows, so do the software tools to support the development and adaptation of simulations to use HLA. There has been considerable experience with the adaptation of simulations which had been designed to operate with Distributed Interactive Simulation Protocols ([20], [21], [22], [23]) and several tools have been developed to provide automated support to this transition ([24], [25]). Other tools have been developed to aid in the transition to HLA through the provision of reusable middleware ([25], [26]) and through the incorporation of HLA functions into new or existing simulation development environments and tools ([27], [28], [29]).

5.6 HLA Compliance Testing

A facility to support testing of compliance of federates to HLA has been established. Compliance with HLA was established with the baseline definition ([30]) along with test procedures ([31],[32]). A federation compliance testing process and supporting system was developed and was fielded in the fall of 1997. This web-based test facility http://hlatest.msosa.dmso.mil/step1.html allows federates to be tested for compliance over the network in a straightforward four step process. A comparable approach for testing RTI software for compliance with HLA is in development.

6. HLA support for simulation reuse and interoperability

Simulation interoperability is defined as "the ability of a ... simulation to provide services to, and accept services from, other ... simulations, and to use the services so exchanged to enable them to operate effectively together." ([32]) This definition of interoperability reflects the overall objective of the HLA that different simulations be able to effectively share data towards a common goal.

As the definition indicates, there are two elements in interoperation: effective data sharing and consistent data interpretation. The HLA requires that federates build the functionality required to interface with the RTI and exchange data with other federates via the HLA specified interfaces. This enables federates to participate in federations, to exchange data, and coordinate their operations within a federation. The HLA also requires all federates and federations to document characteristics of their object representations relevant to other potential users of the federate or federation. This documentation, in the form of completed object model templates, facilitates the information exchange needed for a consistent interpretation of shared data.

Universal interoperability (the ability of any simulation to interoperate with any other simulation, regardless of original purpose or technical implementation) is not feasible with today's technology. Realistically, interoperability will be attainable in degrees, with the required level of interoperability determined by the needs of the federation user. Where interoperability deals with the logical exchange of information between distinct federates during runtime, reuse refers to the adaptation of components (e.g., ideas, whole simulations, lines of code) during the development of a new simulation. Reuse is assisted by having well-defined modular components (the federates) which share the common understanding of what is the meaning of the objects contained within them. Further, reuse can exist at the infrastructure and tool level. Through the HLA, RTI software and HLA based automated support tools can be reused across federates and federations.

The HLA rules, interface specification, and object model template provide minimum essential tools for interoperability and reuse. They ensure that the basic capability for information exchange is in place, they establish the mechanisms for runtime data transfer, and they provide the means to identify appropriate simulations for different purposes. Beyond this, additional interoperability requires additional consistency in the internal representation of the simulations themselves. The extent to which this consistency is required for any particular application depends on the characteristics of that application. While the HLA in itself is insufficient to guarantee interoperability, it provides the technical framework for simulation and federation developers to achieve the degree of interoperability needed to achieve their objectives.

7. Access to HLA technical information and support

An on-line HLA resource center is available providing open access to HLA documentation and public domain HLA supporting software. Accessible via the World Wide Web (http://www.dmso.mil), this resource center is intended to be an open source for use by implementors of the HLA. It includes information on HLA training and education which is provided in monthly regional US training events and select overseas events as well as open hands-on RTI training offered twice a month in the Washington DC area. It also includes online user support both on HLA in general and on use of the RTI and other supporting software.

8. References

[1] "High Level Architecture Interface Specification, v1.3" Defense Modeling and Simulation Office, 5 February 1998, http://hla.dmso.mil/hla/tech/ifspec/if1-3d9b.doc .

[2] "High Level Architecture Object Model Template, v1.3" Defense Modeling and Simulation Office, 5 February 1998, http://hla.dmso.mil/hla/tech/omtspec/omt1-3d4.doc .

[3] "High Level Architecture Rules, v1.3" Defense Modeling and Simulation Office, 5 February 1998, http://hla.dmso.mil/hla/tech/rules/rules1-3d2b.doc .

[4] IEEE Project Authorization Request (PAR) 1516 Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules, reviewed and approved by the IEEE New Standards Committee (NESCOM) , 8 Dec 1997.

[5] IEEE Project Authorization Request (PAR) 1516.1 Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification, reviewed and approved by the IEEE New Standards Committee (NESCOM) , 8 Dec 1997.

[6] IEEE Project Authorization Request (PAR) 1516.2 Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification, reviewed and approved by the IEEE New Standards Committee (NESCOM) , 8 Dec 1997.

[7] U. S. Defense Department, Under Secretary of Defense for Acquisition and Technology, USD (A&T), memorandum, Subj: DoD High Level Architecture (HLA) for Simulations,10 September 1996, http://hla.dmso.mil/hla/policy/kaminski.doc .

[8] NATO Working Paper AC/323-WP02, "NATO Modeling and Simulation Master Plan, Draft v 0.2.2", 6 February 1998.

[9] "Verification, Validation, and Accreditation of Distributed Simulations", Simone M. Youngblood, Paper 97S-SIW-164 presented at the Spring 1997 Simulation Interoperability Workshop, Orlando, FL, 3-7 March 1997.

[10] "An HLA Federation Security Engineering Process", E. Taylor Landrum and Jarrellann Filsinger, Paper 97S-SIW-041, presented at the Spring 1997 Simulation Interoperability Workshop, Orlando, FL, 3-7 March 1997.

[11] "HLA Object Model Development: A Process View", Robert R. Lutz, Paper 97S-SIW-010 presented at the Spring 1997 Simulation Interoperability Workshop, Orlando, FL, 3-7 March 1997.

[12] "Planning for the Evolution of Automation in the HLA Tool Suite", Ken Hunt, Judith S. Dahmann, Jack Sheehan and Robert P. Lutz, Paper 97S-SIW-067 presented at the Spring 1997 Simulation Interoperability Workshop, Orlando, FL, 3-7 March 1997.

[13] "High Level Architecture (HLA) Object Model Tool Development", Roy O. Scrudder and Robert P. Lutz, Paper 97F-SIW-164 presented at the Fall 1997 Simulation Interoperability Workshop, Orlando, FL, 8-12 September 1997.

[14] "Object Model Development: Tools and Techniques ", Christina L. Bouwens; Raymond Miller; Susan Harkrider; Robert Lutz; Roy Scrudder, Paper 98S-SIW-142, presented at the Spring 1998 Simulation Interoperability Workshop, Orlando, FL, 9-13 March 1998.

[15] "HLA Federation and Execution Process Model, V1.1" Defense Modeling and Simulation Office, 9 December 1997, http://hla.dmso.mil/hla/federation/fedep/fedep11.doc .

[16] "FEDEP V1.1", Robert R. Lutz , Paper 98S-SIW-236, presented at the Spring 1998 Simulation Interoperability Workshop, Orlando, FL, 9-13 March 1998.

[17] "HLA FOM/SOM Content Standards", Roy O. Scrudder and Jack H. Sheehan, Paper 97S-SIW-180, presented at the Spring 1997 Simulation Interoperability Workshop, Orlando, FL, 3-7 March 1997.

[18] "High Level Architecture (HLA) Performance Framework", Judith S. Dahmann, Jeffrey Olszewski, Richard Briggs, Russell Richardson, Richard M. Weatherly, Paper 97F-SIW-137 presented at Fall 1997 Simulation Interoperability Workshop, Orlando, FL, 8-12 September 1997, also available at http://hla.dmso.mil/hla/federation/fedep/97F_137.doc .

[19] "Federation Execution Planner's Workbook", Defense Modeling and Simulation Office, http://hla.dmso.mil/hla/federation/fedep/fepw.xls .

[20] "Implementation Of The High Level Architecture Into DIS-Based Legacy Simulations", Wesley K. Braudaway and Susan M. Harkrider, Paper 97S-SIW-089 presented at Spring1997 Simulation Interoperability Workshop, Orlando, FL, 3-7 March 1997, also at http://hla.dmso.mil/hla/implement/dis-hla_trans/hlalega.doc .

[21] "The JPSD Experiment Federation Common Software", Richard A. Briggs and Gordon J. Miller, Paper available at http://hla.dmso.mil/hla/implement/dis-hla_trans/jpsdefcs.doc .

[22] "RTI Common Software Framework", Michael Hooks, Rich Rybacki, Charles Koplik, Paper available via http://hla.dmso.mil/hla/implement/dis-hla_trans/rticsf.doc

[23] "Adapting ModSAF Applications To Comply With The HLA", Joshua E. Smith, Paper available via http://hla.dmso.mil/hla/implement/dis-hla_trans/smith.doc

[24] "Integrating DIS and SIMNET into HLA with a Gateway", Andy Cox, Douglas D. Wood, Mikel D. Petty, and Kenneth A. Juge, presented at 15th DIS Workshop on Standards for the Interoperability of Defense Simulations, Orlando, FL, 16-20 September 1996, pp517-525. Paper also available at http://hla.dmso.mil/hla/implement/dis-hla_trans/dissim.doc

[25] "An Approach to HLA Gateway/Middleware Development", Daniel J. Paterson, Eric Anschuetz, Mark Biddle, Dave Kotick, Paper 98S-SIW-005 presented to the Spring 1998 Simulation Interoperability Workshop, Orlando, FL, 9-13 March 1998.

[26] "Transitioning Your DIS Simulator To HLA", Deb Fullford, Sue Hoxie, Ben Lubetsky, Paper available via http://hla.dmso.mil/hla/implement/dis-hla_trans/fullford.doc .

[27] "ASTT-SES: Scaleable Execution Project", Bernard Ziegler, Presentation at the Fall 1997 Simulation Interoperability Workshop, Orlando, FL, 8-12 September 1997.

[28] "TRAXX^{TM}: An Extensible Federate Development Framework to Support HLA Compliant Federation Implementation", Carl Byers & Lily Lam, Paper 97S-SIW-043 presented at the Spring 1997 Simulation Interoperability Workshop, Orlando, FL, 3-7 March 1997.

[29] “OSIM: Experience of an In-Flight Refueling Federation Development using Commercial Tools”, Jean-Pierre Belanger, Carl Byers and Lily Lam, Paper 97F-SIW-108 presented at the Fall 1997, Simulation Interoperability Workshop, Orlando, FL, 8-12 September 1997.

[30] “HLA Compliance Checklist, Federate, Version 1.1”, Defense Modeling and Simulation Office, 26 March 1997, available from http://hla.dmso.mil/hla/policy/cmp_cl11.html/.

[31] “Test Procedures For High Level Architecture Interface Specification, Version 1.1, Defense Modeling and Simulation Office, 14 May 1997, available from http://hla.dmso.mil/hla/policy/pro_v11.doc

[32] “Test Procedures For High Level Architecture Object Model Template, Version 1.1, Defense Modeling and Simulation Office, 13 May 1997, available from http://hla.dmso.mil/hla/policy/pro_v3-0.doc

[33] U.S. Department of Defense, DoD Directive 5000.59, “DoD Modeling and Simulation (M&S) Management,” 4 January 1994, available from http://www.dmso.mil/docslib/mspolicy/directive.html .