# Evolving the High Level Architecture for Modeling and Simulation

**Roy Scrudder**
**The University of Texas**
**Applied Research Laboratories**
**Austin, Texas**
**scrudder@arlut.utexas.edu**

**Gary M. Lightner**
**The AEgis Technologies Group, Inc**
**Orlando, Florida**
**mlightner@AEgisTG.com**

**Robert Lutz and Randy Saunders**
**Johns Hopkins University**
**Johns Hopkins University Applied Physics**
**Laurel, Maryland**
**robert.lutz@jhuapl.edu,**
**randy.saunders@jhuapl.edu**

**Reed Little**
**Software Engineering Institute**
**Carnegie Mellon University**
**Pittsburg, Pennsylvania**
**little@sei.cmu.edu**

**Katherine L. Morse**
**Science Applications International Corporation**
**San Diego, California**
**katherine.l.morse@saic.com**

**Björn Möller**
**Pitch Technologies**
**Linköping, Sweden**
**bjorn.moller@pitch.se**

## ABSTRACT

The High Level Architecture (HLA) was developed to provide a common architecture for distributed modeling and simulation. Initial HLA development began within the US Department of Defense (DoD) in the mid-1990s. International standardization was achieved in 2000, when the HLA was established as a set of Institute of Electrical and Electronics Engineers (IEEE) standards. Today, the HLA is used worldwide to link constructive and virtual simulations as well as interfaces to live components. HLA federations support a wide variety of applications including training, testing, and acquisition analysis.

Subsequent to the 2000 IEEE standardization of the core HLA standard [the HLA Framework and Rules, the HLA Interface Specification, and the HLA Object Model Template (OMT)] many improvements have been proposed. Several of the proposals incorporate emerging technologies. Among the proposed changes are HLA applications programming interfaces which allow dynamic linking (allowing federations to change run-time infrastructures easily), a web enabled HLA interface, fault tolerance services, and additional flexibility in update rates. Proposed changes to the OMT include additional metadata to characterize HLA federates and federations; additional information about the computing hardware, network, and HLA services used by federates; and an update of the data interchange format to take advantage of advances in the Extensible Markup Language (XML). As part of the periodic review of IEEE standards, the HLA standards have been opened for revision with balloting anticipated in 2006.

This paper will provide a brief background on the HLA and then discuss the process used to update the IEEE versions of the HLA specifications. Each of the major improvements that have been made in that process is described. Finally, information is provided on how readers can participate in the evolution of the HLA standards.

## ABOUT THE AUTHORS

**Roy Scrudder** is a program manager at the Applied Research Laboratories, The University of Texas at Austin, Signal and Information Science Laboratory. Mr. Scrudder holds a B.S. degree in Mathematics from the University of Tennessee. Mr. Scrudder has over 25 years' experience in systems analysis and design, with the last 15 years working in support of various data engineering and data management projects in the defense community. Mr.

Scrudder currently provides data engineering expertise to the Joint Strike Fighter; Multi-mission Maritime Aircraft; Naval Common Database; and both Army and Joint command, control, communications, computers, intelligence, surveillance, and reconnaissance and simulation initialization efforts. Mr. Scrudder was a member of the working group for the IEEE 1516-2000 series of HLA standards and chairs the HLA Evolved Product Development Group.

**Gary M. Lightner** is a Senior Simulation Engineer at The AEgis Technologies Groups Inc. where he serves as Director of the Simulation Technologies and Advanced Research Group. Mr. Lightner holds a B.S. degree in Computer Science from the College of Engineering, Arizona State University and a M.S. degree in Computer Science from the Air Force Institute of Technology. With over 20 years of experience in computer modeling and simulation (M&S) systems design, development, implementation and advanced technology R&D, Mr. Lightner establishes, communicates and fosters the vision, goals and agenda for researching, developing and advancing new and advanced M&S technologies, techniques and standards for the AEgis Technologies Group. In addition Mr. Lightner currently provides R&D and technical expertise to the US Defense Modeling and Simulation Office (DMSO) HLA, VV&A and Education programs as well as the Canadian Synthetic Environment Coordination Office. Mr. Lightner was the drafting editor of the IEEE Std 1516-2000 IEEE Standard for Modeling and Simulation High Level Architecture (HLA) – Framework and Rules and as well as the Chair of the Ballot Resolution and IEEE and the DoD HLA Interpretations committees. Mr. Lightner is currently serving as the drafting editor for the IEEE Std 1516-200X specification under the Simulation Interoperability Standards Organization (SISO) HLA Evolved Product Development Group.

**Robert R. Lutz** is a Principal Staff Scientist at the Johns Hopkins University Applied Physics Laboratory (JHU/APL). Mr. Lutz has over 22 years of experience in the design, implementation, and evaluation of computer M&S systems for military customers. Since joining JHU/APL in 1992, Mr. Lutz has assumed leadership roles on several M&S programs, including the Naval Simulation System, Joint Warfare System, and the Simulation Based Acquisition initiative. Currently, Mr. Lutz is the deputy M&S lead for the Multi-mission Maritime Aircraft Program and actively supports the DMSO on several technology projects. Mr. Lutz also served as the technical editor IEEE Std 1516.1-2000 IEEE Standard for Modeling and Simulation High Level Architecture (HLA) – Object Model Template and as working group chair for the IEEE 1516.3-2003 (HLA Federation Development and Execution Process). Mr. Lutz is currently serving as the drafting editor for the IEEE Std 1516.2-200X Object Model Template as part of the SISO HLA Evolved Product Development Group.

**Randy Saunders** is a Principal Staff Engineer at JHU/APL. Mr. Saunders has over 20 years of experience in the design, implementation, and integration of high-fidelity simulations for military and business customers. Mr. Saunders received M.S. degrees in Engineering from Harvey Mudd College in 1980 and in Computer Science from the University of Southern California in 1985. Mr. Saunders has been involved in distributed simulation standardization since the first Distributed Interactive Simulation (DIS) Workshop, both DIS and HLA standards committees, and as a member of the SISO Standards Activities Committee. Mr. Saunders is presently a vice-chair of the HLA Evolved Product Development Group. Mr. Saunders joined APL in 2001.

**Reed Little** is a Senior Member of the Technical Staff at the Software Engineering Institute, Carnegie Mellon University. Mr. Little investigates software architecture representation issues, concentrating on the relationship to and support of software product lines. Mr. Little was the drafting editor of the IEEE Std 1516.1-2000 IEEE Standard for Modeling and Simulation High Level Architecture (HLA) – Federate Interface Specification and as well as the Chair of the Ballot Resolution committee. Mr. Little is currently serving as the drafting editor for the IEEE Std 1516.1-200X Federate Interface Specification as part of the SISO HLA Evolved Product Development Group.

**Dr. Katherine L. Morse** is a Chief Scientist with SAIC. Dr. Morse is a member of the DMSO SETA team, leading the HLA and Live-Virtual-Constructive programs as well as the deputy manager of the Global Information Grid (GIG) M&S Community of Interest. Dr. Morse was a founding eXtensible Modeling and Simulation Framework (XMSF) partner and is the chair of the SISO XMSF Profile Study Group. Dr. Morse received her B.S. in mathematics (1982), B.A. in Russian (1983), M.S. in computer science (1986) from the University of Arizona, and M.S. (1995) and Ph.D. (2000) in Information & Computer Science from the University of California, Irvine. Dr. Morse has worked in the computer industry for over 20 years, specializing in the areas of simulation, computer security, compilers, operating systems, neural networks, speech recognition, image processing, and engineering process development. Dr. Morse's Ph.D. dissertation is on dynamic multicast grouping for Data Distribution Management, a field in which she is widely recognized as a foremost expert.

**Björn Möller** is the Vice President and cofounder of Pitch, the leading supplier of tools for HLA 1516 and HLA 1.3. Mr. Möller leads the strategic development of Pitch HLA products. Mr. Möller serves on several HLA standards and working groups and is currently the vice chair (Europe) for HLA Evolved Product Development Group. Mr. Möller has a wide international contact network in simulation interoperability. Mr. Möller has twenty years of experience in high-tech R&D companies with an international profile in areas such as modeling and simulation, artificial intelligence and Web based collaboration. Mr. Möller holds an MSc in Computer Science and Technology after studies at Linköping University, Sweden and Imperial College, London.

# Evolving the High Level Architecture for Modeling and Simulation

**Roy Scrudder**
**The University of Texas**
**Applied Research Laboratories**
**Austin, Texas**
**scrudder@arlut.utexas.edu**

**Gary M. Lightner**
**The AEgis Technologies Group, Inc**
**Orlando, Florida**
**mlightner@AEgisTG.com**

**Robert Lutz and Randy Saunders**
**Johns Hopkins University**
**Johns Hopkins University Applied Physics**
**Laurel, Maryland**
**robert.lutz@jhuapl.edu,**
**randy.saunders@jhuapl.edu**

**Reed Little**
**Software Engineering Institute**
**Carnegie Mellon University**
**Pittsburg, Pennsylvania**
**little@sei.cmu.edu**

**Katherine L. Morse**
**Science Applications International Corporation**
**San Diego, California**
**katherine.l.morse@saic.com**

**Björn Möller**
**Pitch Technologies**
**Linköping, Sweden**
**bjorn.moller@pitch.se**

N*ote to readers: Because of the volume of acronyms defined and reused in the paper, an acronym list has been provided as Appendix A.*

## INTRODUCTION

The High Level Architecture (HLA) was developed to provide a common architecture for distributed modeling and simulation. As such, HLA can be used to link simulations and interfaces to live systems (collectively known as federates) into federations. HLA is defined by three core specifications:

- HLA Framework and Rules – A set of ten rules, five applying to federates and five applying to federations, that together ensure the proper interaction of federates in a federation and define the responsibilities of federates and federations.

- HLA Federate Interface Specification – A specification of the services and interfaces that a runtime infrastructure (RTI) must implement to ensure the correct operation of federations and the callbacks functions that federates must provide. The Federate Interface Specification also includes language-specific application programming interfaces (APIs) for the services and callbacks.

- HLA Object Model Template (OMT) – A template to be used by federates to specify their capabilities to exchange data (known as a simulation object model or SOM) and by federations to specify the data to be exchanged during federation execution (known as a federation object model or FOM).

### A Brief History of HLA

The development of HLA began in the US Department of Defense (DoD) in 1995 with the initial specification of the architecture. During the next 18 months, a series of prototype RTIs were developed and prototype federations were implemented within a range of application domains. At the end of the prototyping period in August 1996, the DoD released Version 1.0 of the three HLA specifications. An Architecture Management Group was established with representation from across the DoD and coordination by the Defense Modeling and Simulation Office (DMSO). Over an 18 month period, the HLA continued to evolve, culminating in the final release of the DoD-controlled specifications, Version 1.3, in March 1998.

In 1997, the evolution of the HLA moved to the Simulation Interoperability Standards Committee (SISC) of the Institute of Electrical and Electronics Engineers (IEEE). Impetus for this move included DoD policy on the adoption of open commercial standards and the interest outside the DoD in evolving the HLA standards. An IEEE Working Group formed by SISC conducted a series of review rounds including the submission of comments and change proposals and public meetings to resolve changes to the specifications. The revised draft was subsequently balloted under IEEE rules and, in 2000, passed on the second ballot. The resulting specifications form the current baseline for HLA (IEEE, 2000):

- IEEE 1516-2000, IEEE Standard for Modeling and Simulation (M&S) HLA – Framework and Rules

- IEEE 1516.1-2000, IEEE Standard for Modeling and Simulation (M&S) HLA – Federate Interface Specification

- IEEE 1516.2-2000, IEEE Standard for Modeling and Simulation (M&S) HLA – Object Model Template

### Current HLA Evolution Process

By IEEE policy, all IEEE standards must be evaluated every five years and either reaffirmed as they stand, revised, or retired. Subsequent to the establishment of the IEEE standards in 2000, the potential for improving the standards became evident in the first three years. The DoD was called upon to make over 200 interpretations to the specifications to enable the certification of RTIs as compliant with the specifications. Most of these interpretations clarified ambiguities in the specification. During this same period, the Dynamic Link Compatible (DLC) API effort commenced with the Simulation Interoperabilty Standards Organization (SISO). The DLC effort produced APIs which enable federations to change RTIs without recompiling and relinking federate code. These changes are described in more detail later.

Given the immediate opportunities for improvement, a Product Development Group (PDG) was established in early 2004 with the goal of immediately undertaking the improvement of the IEEE HLA standards. The HLA user community deemed this preferable to waiting for the IEEE renewal cycle to begin in 2005.

During the period between the establishment of the IEEE HLA standards in 2000 and the creation of this PDG in 2004, responsibility for maintaining the IEEE HLA standards shifted. In 2003, IEEE's Standards Activity Board (SAB) approved the SISO Standards Activity Committee (SAC) as an IEEE Sponsor Committee. As such, SISO is able to develop and maintain IEEE standards. As a result, SISC (the IEEE body that initially established the 2000 series of HLA standards) was disestablished, and SISO took over as the standards developer for HLA. Consistent with the agreement between IEEE and SISO, the HLA standards are being evolved by a SISO PDG and will be balloted under IEEE procedures. The resulting standards will be published as IEEE Std 1516-200X, IEEE Std 1516.1-200X and IEEE Std 1516.2-200X, where "200X" will denote the year of publication, most likely 2006.

This effort is colloquially known as "HLA Evolved." Currently, there are over 150 participants in the HLA Evolved PDG. Membership of the group includes government, industry, and academic personnel; both producer (e.g., RTI developer) and user (e.g., federation manager) personnel; and is very much international in its membership. Current membership includes representatives from the United States, United Kingdom, Sweden, France, Canada, and Japan, among other nations.

At the time this paper was written, the PDG had completed two rounds of review and changes, evolving all three specifications. The current projected schedule calls for one additional round of review before moving to balloting in 2006. The exact point at which open review cycles of the document end and balloting begins must be voted upon by the PDG membership and approved by both the SISO SAC and IEEE SAB.

Each round of review commences with the circulation of draft documents to the entire PDG and the submission of both technical and editorial comments by members. Drafting Groups formed for each of the specifications from the large PDG address the comments as a whole for each document and make recommendations to the PDG as to how they should be resolved. Where necessary, the PDG establishes small Tiger Teams to make recommendations on specific focused issues. Taking the Drafting Group's and Tiger Teams' recommendations into consideration, the PDG votes on the changes to be implemented. The result can be the adoption of a comment as submitted, an alternative resolution, or a decision to not make the recommended change. The Drafting Group then updates the specifications consistent with the PDG's decisions. At the end of each review round, the PDG decides whether to hold an additional review round or to move to balloting.

## CHANGES TO THE SPECIFICATIONS

The following sections will describe the major changes approved thus far by the HLA Evolved PDG. Many more changes including minor technical changes and editorial changes have been approved by the PDG.

### HLA Framework and Rules

There were no major changes to the HLA Framework and Rules Specification. The ten rules have remained unchanged. There was some additional wording added to the rationale section for Rule 2 to emphasize the importance of the RTI knowing nothing about the objects being simulated. Such knowledge would result in tailored, application-specific RTIs and thus defeat the purpose of a generalized standard. Several definitions and acronyms were added to the specification, including ones for “delete”, “remove”, “subscribe”, “publish”, and Base Object Model (BOM). These definitions and acronyms sections apply to all three specifications in the 1516 series. In addition, the IEEE Std 1516.3-2003 HLA Federation Development and Execution Process (FEDEP) recommended practices specification was added to the references list as was the BOM specification.

### HLA Federate Interface Specification

Incremental improvements were realized in evolving the IEEE 1516.1-2000 standard. The changes were minor with the existing services and callbacks remaining intact. IEEE 1516.1-2000 compliant federates will require modification to come into compliance with IEEE 1516.1-200X, but the extent of the changes should not be significant. The publish/subscribe, register/discover, update/reflect, and send/receive general HLA mechanisms are almost completely unchanged. The changes induced by the 2005 standardization process can be thought of as being around the fringes. One significant change was the removal of the Ada API. No use of the Ada API has been reported from the user community, and thus the PDG voted to remove it rather than maintain it. The removal of the Management Object Model (MOM) was proposed, but the PDG has yet to find sufficient impetus to remove this section. The major changes made as part of this evolution are described in the sub-sections.

### Dynamic Link Compatible (DLC) API

In 2004, the SISO DLC HLA API PDG created revisions to the IEEE 1516.1-2000 Java and C++ APIs to produce a dynamic link compatible set of HLA APIs (SISO, 2004). The primary objective of the DLC standard was to provide a mechanism to permit federates to utilize any RTI developed in compliance with the High Level Architecture, IEEE 1516.1-2000 and the DLC specification, without recompiling or relinking federate code. This capability is extremely valuable to owners of federates that participate in multiple federations using different RTIs. The DLC API was not intended to facilitate functional compatibility. The DLC standard addresses both statically and dynamically linked RTIs.

The general approach to achieving dynamic-link compatibility for C++ was to define a standard set of header files to which all compliant RTI implementations build. A federate compiles against the standard headers and dynamically links against any RTI that complies with this standard. The federate can then switch to a different RTI implementation by utilizing any DLC RTI library.

Java link compatibility required a solution that provided an implementation-independent way to get access to implementation-specific RTI classes. The solution in the DLC API was to have a common RtiFactoryFactory class that returns an implementation-specific class that implements the standardized interface RtiFactory that in turn can provide the actual class instances that are needed for a specific RTI.

A modified DLC standard was incorporated into the 1516.1-200X Java and C++ APIs. Modifications were necessary to take into account other changes in 1516.1-200X and to solve some issues discovered subsequent to the approval of SISO-STD-004.1-2004.

### Fault Tolerance

As HLA has moved out of the closed lab to widespread deployment in real-life environments, the need to be able to handle faults has been increasingly important. To address this need this, supporting fault tolerance functionality for the HLA Interface Specification was proposed and approved.

As part of the HLA Evolved effort, a fault was defined as “a problem that occurs in the federation or its environment that prevents the entire federation from interoperating in an HLA compliant manner.” There are two types of faults as illustrated in Figure 1:

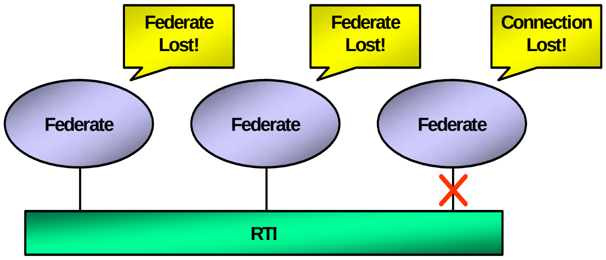

**Figure 1. Types of Faults**

*Federate Lost* occurs when a federate has been unexpectedly lost from a federation as a result of a fault. This fault is signaled using the MOM interaction “HLAreportFederateLost” to any federate in the remaining federation that subscribes to this interaction. The MOM interaction also provides information about the last known-good timestamp to which the lost federate was granted. This timestamp value is provided on a best effort basis and may not represent the highest time advance value the federate was actually granted.

*Connection Lost* occurs when a federation has been unexpectedly lost from a federate as a result of a fault. This fault is signaled through a callback from the RTI to the federate. The federate then enters the Not Connected state. The federate now needs to do a Connect to the RTI again; otherwise a Not Connected exception will be thrown when an RTI call is made. Note that this callback is done to a federate experiencing a fault. It is done on a best-effort basis and may or may not succeed, since this call may be done by a local RTI component that may have lost contact with the rest of the federation or is trying to execute on a computer without power.

When a federate has been lost, the RTI is responsible for reporting this and then resigning on behalf of the lost federate using the "Automatic Resign Directive." This directive is specified in the FOM and can also be accessed using support services. The MOM information and relevant advisories are updated as with a usual resign action. This also means that all federation-wide synchronized operations like time management are recovered. If for example all federates are waiting for a specific federate to advance time and that federate has crashed, this would otherwise have prevented the entire federation from advancing. The final result of the resign and recovery will be a reduced but HLA compliant federation.

The fault detection implementation may vary from RTI to RTI. Heartbeats and broken TCP link detection are some possibilities. The need for and implementation of fault tolerance is also expected to vary from federation to federation. A number of design patterns for fault tolerant federations to match different needs have also been developed to facilitate for federations that want to incorporate this (Möller, Löfstrand, and Karlsson, 2005), (Pearce and Möller, 2005)

### Smart Update Rate

In many federations it is difficult to find a balance for the update rate. Typical examples of this are:

- Mixing older federates with new, high-performing federates. Some subscribing federates may want a higher update rate; some want a lower update rate. Some federates in a federation by be unable to handle fast update rates. A good example would be a command and control surrogate federate in a federation involving aircraft position updates. The federates implementing the aircraft will want position updates at a much faster rate than the command and control federate.
- Different capacities for wide-area network links versus local link. In a wide-area network it may be crucial to provide updates at a lower rate over long-haul links with limited capacity. In this case, it is desirable to subscribe to data for the same class with higher update rate from local federates than from remote federates.
- Focus on some entities. The desire is to get a higher update rate for some instances but not for others. Again the previous example involving aircraft positions would apply. It is possible that a federate would be interested in receiving updates about threat aircraft on a more frequent basis than updates about friendly aircraft.

Some federates are also capable of producing updates at different data rates, i.e., they rate at which they provide updates can be adjusted during federation execution. It is useful to know at what data rate updates are really consumed by subscribing federates so that the production rate can be adjusted accordingly.

A method for smart update rate reduction has been added to the 1516.1-200X Federate Interface Specification. It allows a federate to subscribe to updates and provide a requested update rate. Figure 2 shows an example where the desired update rates for some federates do not match the publication capabilities of other federates.

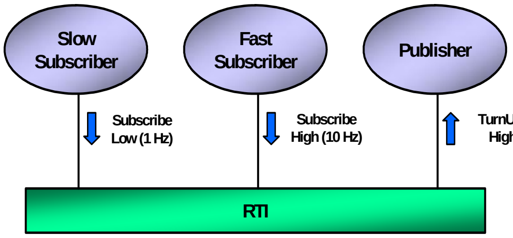

**Figure 2. Smart Update Rate Example**

This will cause the RTI to deliver incoming updates according to the following:

- For best-effort transportation, no two updates will be delivered closer than (1/MaxRateValue ) in time. Superfluous updates will be dropped by the RTI. There is no guaranteed minimum update rate since there is no guarantee from the owning federate about minimum update rate.
- For reliable transportation, the RTI will not drop any messages but inform the owning federate according to the description below.

Data Distribution Mangement (DDM) offers an alternative method to get different update rates for different entities of the same class, e.g., aircraft. DDM can be used to differentiate between airplanes on the local site and airplanes on remote sites. DDM may also be used to get different update rates for friendly or enemy airplanes.

So that a producing federate can restrict its publication rate to match the interest of subscribing federates, Interface Specification callbacks were added. The RTI will call to the producing federate to inform it about the maximum requested update rate.

The smart update rate reduction will make it possible to make HLA federations even more scalable and distributed. It is also useful when mixing older and newer simulation systems. Another application area is when mixing systems with fundamentally different update rate requirements, for example C4I systems and high-resolution platform simulations (Möller and Karlsson, 2005).

### Web Supportive HLA

The concept of making the HLA available as some form of a web service (Möller and Löf, 2005) was first proposed and prototyped as part of the eXtensible Modeling and Simulation Framework, XMSF (Morse, Brunton, and Drake, June 2003) using an RTI compliant with the DoD HLA 1.3 specification. The efficacy of the technology was demonstrated in a series of prototypes (XMSF, 2005; Morse, Brunton, and Drake, April 2003; Morse, Brunton, and Drake, June 2004; and Morse, Brunton, Drake, Busch, and Jilson, 2005). These prototypes used a Web Enabled RTI (WE RTI). Because it was only a prototype, this WE RTI included development of the Extensible Markup Language (XML) Schema Definition (XSD) for the DoD 1.3 Java APIs, but not the Web Services Description Language (WSDL). That is, it described the data types to be exchanged, but not the full APIs as would be required for a web service. Further, it was compliant with DoD HLA 1.3, not IEEE 1516.

One of the lessons learned with the WE RTI was that DDM proved to be particularly challenging because of its use of opaque data types. Because IEEE 1516 has even more of these, a greater challenge was anticipated in extending the approach to IEEE 1516.

This challenge notwithstanding, the technology had already proven to be valuable as an extension to existing federations that were not fundamentally web enabled. Furthermore, the importance of web services for the Global Information Grid (GIG) and the growing opportunity for M&S in the GIG indicated that tackling this challenge would be worth the returns. RTI vendors were engaged in preliminary prototyping and analysis to determine if extending the concept to IEEE 1516 was technically feasible (Möller and Löf, 2005). This preliminary investigation proved to be promising, so the following comment was submitted in the HLA Evolved effort:

> The evolving and expanding emphasis on the use of web services in the GIG encompasses a need to make simulations available as web services, to operate within a web services environment, and to interoperate with other web services. To support this need, the interface specification should be evaluated and updated as necessary to ensure the technical feasibility of developing web-enabled RTIs. This evaluation is anticipated to impact the signatures of some services and may also entail the addition of another subsection to section 12, "Programming Language Mappings," for a web service binding.

The PDG approved formation of a Tiger Team to develop the WSDL API and language mappings. This is a non-trivial task considering that the result would have to cover all the APIs and would be longer than any of the existing APIs due to the verbosity of XML and WSDL.

By March 2005, the Tiger Team had considered the broader question of what it means for the entire HLA to be "web-supportive," but decided to focus on modifications to the Interface Specification to enable RTIs to be called as web services. The Tiger Team also realized that some of the technologies required for the "best" solution, e.g. publish-subscribe web services, were not yet standardized. The Tiger Team concluded that if they proceeded on their own path, they ran the risk of adopting a non-standard approach. Tiger Team chose to standardize those aspects for which the underlying technology is most standardized, i.e.:

- XSD for mapping the data types in the services
- WSDL definition of the services
- Simple Object Access Protocol (SOAP) binding for WSDL

The additions to the Interface Specification will include a WSDL API annex analogous to C++ and Java annexes including the SOAP binding, and a new subsection of 12, "Programming language mappings," that includes the XSD for mapping the data types, although the mapping also could have been in the WSDL annex because of the nature of WSDL. The latter decision was made to preserve the general structure of the Interface Specification.

Appendix B shows a “short” example of the WSDL API. The first few lines are the Java API for Create Federation Execution. The Java API is followed by the considerably longer WSDL API for the same service.

### Fuzzy DDM

The current IEEE 1516-2000 series standards provide DDM services that may be used by federates to reduce both the transmission and the reception of irrelevant data. One of the goals of DDM is to aid scalability and performance. However, the exact matching of publication and subscription regions for each message (as DDM is specified in IEEE Std 1516.1-2000) may cause scalability and performance problems in some implementations.

The PDG agreed to establish a Tiger Team to investigate adding an option to have DDM matching be ”fuzzy,” i.e., inexact in certain situations. The Tiger Team was to determine if “fuzzy DDM” should be allowed, and if so, how it would specified (e.g. in the FOM or at run-time) and what its semantics would be. The Tiger Team work is ongoing and the exact nature of the solution, or if one is feasible is not known at this time.

### Transportation Types

Some minor modifications are under consideration in order to support special additional transportation types. The current IEEE 1516-2000 series standards only support reliable and best-effort transportation types. If these changes are implemented via the HLA Evolved effort, fall back to a default behavior will be specified; this facilitates federate portability when implementing new transportation types (quality of service, high/low priorities, IPv6, etc.). The proposed changes include modification of the OMT Transportation Type Table and the addition of four Interface Specification support services.

### Encoding Helpers

The DLC Java API contained encoding/decoding helpers. These helper classes assist a federate developer with encoding and decoding of data according to the specification in the HLA OMT standards document. These Java helpers were included in the update to the 1516.1-200X Java API.

However, there were no encoding/decoding helpers in the DLC C++ API. The HLA Evolved PDG decided it would be beneficial for the 1516.1-200X C++ API to also have encoding/decoding helpers. A HLA Evolved PDG Tiger Team is developing a general description of C++-based encoding/decoding and proposed C++ encoding/decoding helpers for the 1516.1-200X C++ API. It is highly likely that the PDG will approve the addition of these helpers once the details have been specified.

## HLA Object Model Template

The OMT defines a standardized structural framework for describing the information that can be produced or consumed by individual federates (in a SOM) or exchanged among multiple federates (in a FOM). The OMT is an essential component of the HLA for the following reasons. The OMT provides a commonly understood mechanism for specifying the exchange of data and general coordination among members of a federation. It provides a common, standardized mechanism for describing the capabilities of potential federation members. Additionally, the OMT facilitates the design and application of common tool sets for development of HLA object models.

The OMT experienced a significant amount of change during the development of the original IEEE 1516-2000 series of specifications. Examples of improvements incorporated into the OMT included the addition of new tables to extend the capability to document federate/federation-relevant information, improvements in the mechanisms used to support datatyping, and the migration of the OMT Data Interchange Format (DIF) to an XML representation.

The efforts of the HLA Evolved PDG are now producing the next generation of OMT improvements. The following subsections will focus on two areas in which user needs have resulted in important new OMT capabilities.

### Object Model Identification Table Extensions

The Object Model Identification Table provides a means of specifying basic high-level information about the object model (i.e., metadata) for the purpose of facilitating object model reuse. This metadata also aids in the selection of federates for incorporation into federations. The name and version of the object model, the creation date, the sponsoring agency, and relevant points of contact are all examples of object model metadata.

In January 2003, a SISO PDG was formed to develop a specification and guidance document for BOMs. BOMs are discrete patterns of simulation interplay intended for describing reusable components or “piece-parts” of a simulation or system. To facilitate reuse of BOMs, discovery metadata was an extremely important consideration. The BOM PDG used the IEEE 1516.2 Object Model Identification Table as its starting metadata framework, but quickly discovered that the thirteen categories supported by this table were insufficient for fully describing BOMs such that the desired level of reuse and understanding could take place. To address this deficiency, the BOM PDG defined several extensions to the Object Model Identification Table based upon a mapping of their needs to specific elements defined in other established metadata standards including Dublin Core and the Defense Discovery Metadata Specification (Gustavson, Bachman, Lutz, and Scrudder, 2005). In recognition of the fact that these extensions were of potential value to the broader HLA community, the BOM PDG submitted their extensions as comments into the HLA Evolved process. Approval of these comments by the HLA Evolved PDG provided a much richer, more complete set of object model metadata for the Object Model Identification Table. Examples of some of these extensions include release restrictions, use history, keywords, and security classification.

**XML Schema**

The OMT DIF is an intermediate format for electronic exchange of HLA object model data. In the DoD HLA Version 1.3, the DIF was an ASCII text tagged-value format, formally defined in the Backus Naur form (BNF). While the use of the tagged-value format was adequate for early implementations of HLA object model development tools (OMDTs), participants in IEEE 1516.2 development voted to migrate to XML, which had emerged as a W3C recommendation. At that time, Document Type Definitions (DTDs) were the established method for defining the valid structure of XML documents and thus provided the foundation of the OMT DIF representation. Subsequent to the release of the original IEEE 1516-2000 series specifications, XML Schemas emerged as the recognized successor to DTDs. Some of the reasons for this include explicit support for datatyping, greater extensibility, and support for namespaces (XML Schema, 2005). The HLA Evolved PDG has approved the transition to a Schema-based DIF representation, taking advantage of many of these features. Future implementations of HLA OMDTs will include the capability to automatically generate this DIF representation based upon user input into the various table structures.

**Conformance Specification**

The IEEE 1516-2000 series standards are presently structured to an all-or-nothing assessment of RTI conformance; however, no part of the standards specifically defines what conformance or testing is required. DMSO has developed testing capabilities to support the US HLA community and NATO requirements, but they are not coupled directly to the IEEE HLA standards. Experience with Federate Compliance Testing (FCT) shows that most federates are not capable of using all the HLA services (Crooks, Igarza, Löf, and Turrell, 2004). Significant parts of HLA, such as DDM or deterministic time management, raise conformance questions since only a small percentage of federations depend on them.

The HLA Evolved PDG has addressed the technical and policy questions involved with various conformance strategies. A discussion of conformance testing will be added to the OMT along with an optional table to capture the services that a federate or federation uses.

**Federation Agreements**

Federation integrators have used a wide variety of conventions and rules to prescribe the behavior of federates in federations. These rules, commonly called Federation Agreements as defined in FEDEP Step 4, are expressed as textual documentation for the federation. Significant complications have arisen in the interpretation and enforcement of these text agreements (Ceranowicz, Torpey, Helfinstine, Evans, & Hines, 2002).

Two efforts have attempted to transform agreements from free text to more structured forms. Both have taken structured metadata approaches similar to those used in the OMT to describe object models. The FCT system uses keyword-based metadata to define which Interface Specification services a federate can utilize. Based on this data, the FCT automatically conducts a verification test to demonstrate the consistent performance of the federate. Although the FCT domain of application is small, the same concept was used in the Federation Execution Planner’s Workbook (FEPW). The FEPW captures a broader spectrum of information about the federation, but automatic testing has not been developed.

The HLA Evolved PDG has considered proposals to add tables incorporating the FEPW data for FOMs. Both tables would be optional. However, the HLA Evolved PDG found that these concepts were not mature to the point that they should be added to the OMT standard at this time.

## PARTICIPATING IN HLA EVOLVED

Readers are encouraged to participate in the evolution of the HLA Standards by subscribing to the HLA Evolved discussion board at www.sisostds.org. There is no additional requirement to participate during the comment rounds, but IEEE Standards Association membership will be required to ballot the revise standards.

## CONCLUSIONS

Although the core of the HLA specifications remain unchanged, it is obvious that the changes the HLA Evolved effort is bringing forth will result in a greatly improved architecture. The web enabling of HLA paves the way for the future of simulation in network environments such as the GIG. The fault tolerance, transportation type, and smart update rate enhancements will allow for considerable improvements in ensuring quality of service in HLA. Other changes too numerous to mention here will improve the clarity and usability of the HLA specifications.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the contributions of the entire membership of the HLA Evolved PDG. Without the PDG's efforts the improvements that are being made to the HLA would not be possible.

## REFERENCES

Crooks, M., Igarza, J., Löf, S., & Turrell, C. (2004). HLA Compliance Certification: Lessons learned from the US past activity and current developments within NATO and PfP nations, *Proceedings of 2004 Euro Simulation Interoperability Workshop*, 04E-SIW-071, Edinburgh, Scotland, June 2004.

Ceranowicz, A., Torpey, M., Helfinstine, B., Evans, J., & Hines, J. (2002). Reflections on Building the Joint Experimental Federation, *Proceedings of 2002 Interservice/Industry Training Simulation, and Education Conference*, Orlando, Florida, December 2002.

Gustavson P., Bachman J., Lutz R., & Scrudder R. (2005). Understanding the BOM Metadata and Making It Work For You, *Proceedings of 2005 Spring Simulation Interoperability Workshop*, 05S-SIW-084, San Diego, California, April 2005.

Institute of Electrical and Electronics Engineers (2000), *IEEE Standard for Modeling and Simulation (M&S) HLA – Framework and Rules* (IEEE Std 1516-2000).

Institute of Electrical and Electronics Engineers (2000). *IEEE Standard for Modeling and Simulation (M&S) HLA – Federate Interface Specification* (IEEE Std 1516.1-2000).

Institute of Electrical and Electronics Engineers (2000). *IEEE Standard for Modeling and Simulation (M&S) HLA – Object Model Template* (IEEE Std 1516.2-2000).

Möller, B & Karlsson, M. (2005). Developing the Fault Tolerance Support Extensions for HLA Evolved, *Proceedings of 2005 Euro Simulation Interoperability Workshop*, 05E-SIW-019, Toulouse, France, June 2005.

Möller, B. & Karlsson, M. (2005). Developing well-balanced federations using the HLA Evolved smart update rate reduction, *Proceedings of 2005 Fall Simulation Interoperability Workshop*, 05F-SIW-87, Orlando, Florida, September 2005.

Möller, B. & Löf, S. (2005). Mixing Service Oriented and High Level Architectures in Support of the GIG, *Proceedings of 2005 Spring Simulation Interoperability Workshop*, 05S-SIW-064, San Diego, California, April 2005.

Möller, B., Löfstrand, B., & Karlsson, M. (2005). Developing Fault Tolerant Federations using HLA Evolved. *Proceedings of 2005 Spring Simulation Interoperability Workshop*, 05S-SIW-048, San Diego, California, April 2005.

Morse, K., Brunton R., & Drake, D. (2003). WMDOA Integration Employing Web Services for Federate Communication – an XMSF Exemplar, *Proceedings of the 2003 Spring Simulation Interoperability Workshop*, 03S-SIW-071, Kissimmee, Florida, April 2003.

Morse, K., Brunton R., & Drake, D. (2003). Web Enabling an RTI – an XMSF Profile, *Proceedings of the 2003 European Simulation Interoperability Workshop*, 03E-SIW-046, Stockholm, Sweden, June 2003.

Morse, K., Brunton R., & Drake, D. (2004). Web Enabling HLA Compliant Simulations to Support Network Centric Applications, *Proceedings of the 2004 Command and Control Research and Technology Symposium*, San Diego, California, June 2004.

Morse, K., Brunton R., Drake, D., Busch, J., & Jilson, E. (2005). Enhanced Distance Learning for DVTE: Real Time Feedback in an Integrated SCORM Environment, *Proceedings of 2005 Spring Simulation Interoperability Workshop*, 05S-SIW-120, San Diego, California, April 2005.

Pearce, T., & Möller, M. (2005). Towards Fault Tolerant RTIs, Federates and Federations, *Proceedings of 2005 Spring Simulation*

*Interoperability Workshop*, 05S-SIW-129, San Diego, California, April 2005.

Simulation Interoperabilty Standards Organization (2004). *Dynamic Link Compatible HLA API Standard for the HLA Interface Specification (IEEE 1516.1 Version)*, (SISO-STD-004.1-2004).

XMSF (2005). http://www.xmsf.org, Science Application International Corporation.

XML Schema (2005). *Introduction to XML Schema*, http://www.w3schools.com/schema/schema_intro.asp, W3Schools.

**Appendix A – Glossary of Terms**

| | |
|---|---|
| API | Applications Programming Interface |
| BNF | Backus Naur Form |
| BOM | Base Object Model |
| DIF | Data Interchange Format |
| DIS | Distributed Interactive Simulation |
| DLC | Dynamic Link Compatible |
| DMSO | Defense Modeling and Simulation Office |
| DoD | Department of Defense |
| DTD | Document Type Definitions |
| FCT | Federate Compliance Testing |
| FEDEP | Federation Development and Execution Process |
| FEPW | Federation Execution Planner’s Workbook |
| FOM | Federation Object Model |
| HLA | High Level Architecture |
| IEEE | Institute of Electrical and Electronics Engineers |
| IP | Internet Protocol |
| JHU/APL | Johns Hopkins University Applied Physics Laboratory |
| M&S | Modeling and Simulation |
| MOM | Management Object Model |
| NATO | North Atlantic Treaty Organization |
| OMDT | Object Model Development Tool |
| OMT | Object Model Template |
| PDG | Product Development Group |
| RTI | Runtime Infrastructure |
| SAB | Standards Activity Board |
| SAC | Standards Activity Committee |
| SISC | Simulation Interoperability Standards Committee |
| SISO | Simulation Interoperabilty Standards Organization |
| SOAP | Simple Object Access Protocol |
| SOM | Simulation Object Model |
| TCP | Transmission Control Protocol |
| US | United States |
| WE RTI | Web Enabled RTI |
| WSDL | Web Services Definition Language |
| XML | Extensible Markup Language |
| XMSF | eXtensible Modeling and Simulation Framework |

## Appendix B – Web Services Definition Language Example for HLA

```
public void createFederationExecution(String federationExecutionName,
  java.net.URL fdd,
  String logicalTimeImplementationName)
throws
  FederationExecutionAlreadyExists,
  CouldNotOpenFDD,
  ErrorReadingFDD,
  CouldNotCreateLogicalTimeFactory,
  NotConnected, RTIinternalError;

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="LocalPlaceHolder"
xmlns:localns="LocalPlaceHolder"
xmlns:wsrtins="http://siso.org/hla/evolved/wsrti"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <!-- Sec 4.2 CREATE FEDERATION EXECUTION -->
    <wsdl:message name="createFederationExecutionRequest">
        <wsdl:part name="federationExecutionName" type="xsd:string"/>
        <wsdl:part name="fomDocumentDesignator" type="xsd:string"/>
        <wsdl:part name="sessionID" type="xsd:base64Binary"/>
    </wsdl:message>
    <wsdl:message name="createFederationExecutionResponse">
    </wsdl:message>

    <wsdl:portType name="RTIServicePortTypes">
        <!-- Sec 4.2 CREATE FEDERATION EXECUTION -->
        <wsdl:operation name="createFederationExecution" parameterOrder="federationExecutionName
          fomDocumentDesignator sessionID">
            <wsdl:input message="localns:createFederationExecutionRequest"
              name="createFederationExecutionRequest"/>
            <wsdl:output message="localns:createFederationExecutionResponse"
              name="createFederationExecutionResponse"/> </wsdl:operation>
        </wsdl:operation>
    </wsdl:portType>

    <wsdl:binding name="RTIServiceSoapBinding" type="localns:RTIServicePortTypes">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="createFederationExecution">
            <soap:operation soapAction=""/>
            <wsdl:input name="createFederationExecutionRequest">
                <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                  use="encoded"/>
            </wsdl:input>
            <wsdl:output name="createFederationExecutionResponse">
                <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                  use="encoded"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl: binding >

    <wsdl:service name="RTIServiceService">
        <wsdl:port name="RTIService" binding="localns:RTIServiceSoapBinding">
            <soap:address location="http://..."/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```