Migrating the HLA Object Model Template to an IEEE Standard

Robert R. Lutz

DoD's High Level Architecture (HLA) was established to promote and facilitate interoperability across a wide range of military simulation systems. The purpose of the HLA Object Model Template (OMT) is twofold: (1) to provide a standardized mechanism for HLA federations to formally specify the format of runtime data exchange, and (2) to provide a standard format for specifying the external interface of individual HLA federates. This article reports on recent efforts to transition the current DoD version of the OMT to a recognized commercial standard. The OMT modifications implemented in this transition are highlighted, as is the standardization process that was used to gain community consensus and acceptance. (Keywords: HLA, IEEE, OMT.)

INTRODUCTION

In October 1995, the U.S. Under Secretary of Defense for Acquisition and Technology [USD(A&T)] published a master plan for the use of modeling and simulation (M&S) in DoD applications. The purpose of this master plan was to establish a future vision for DoD M&S. Included in this plan were a set of high-level M&S objectives, including specific actions and responsibilities for accomplishing these objectives.[1]

The first major objective identified in the DoD M&S Master Plan was the establishment of a common technical framework for M&S as a means of facilitating interoperability and reuse. A High Level Architecture (HLA) for M&S was identified as the first and most prominent component of the framework. The need for the HLA was based on the premise that no one simulation system could satisfy the demands of all users, and thus an architecture was essential for composing unified simulation environments from multiple, interacting simulation systems. The HLA had to be widely applicable across the full range of DoD application areas and could not prescribe a particular implementation approach.

An initial definition of the HLA was produced in March 1995 based on the synthesis of industry inputs and previous DoD architecture efforts. This initial definition provided the starting point for a 17-month prototyping phase to test and mature the architecture via active use in several different application areas. An Architecture Management Group (AMG) was established to oversee the process and generally to guide the technical evolution of the HLA. The prototyping period culminated with the baseline release of the HLA in August 1996. The following month, the USD(A&T) formally designated the HLA as the standard technical architecture for all DoD simulations and directed all DoD components to establish plans for transitioning to the HLA.

Since its baseline release, the HLA has continued to mature and evolve. The most recent release of the HLA specifications (V1.3) occurred in February 1998. Although V1.3 is currently a recognized DoD standard, efforts to broaden the HLA user base by producing a commercial standard were initiated in June 1997. This article describes the activities required to transition to a commercial standard and the resulting effect the process had on one of the major components of the HLA, i.e., the Object Model Template (OMT).

BASIC HLA PRINCIPLES

The HLA consists of three major components:

1. HLA Rules: A set of principles and conventions that must be followed to achieve the proper interaction of federates during a federation execution.
2. HLA Interface Specification: A specification of the functional interface between HLA federates and the runtime infrastructure (RTI; a distributed operating system with services that support and control the exchange of information among federates during execution.)
3. HLA OMT: A standard presentation format and syntax for describing HLA Simulation Object Models (SOMs) or Federation Object Models (FOMs).

Although each of these components is equally necessary and important, this article focuses primarily on the HLA OMT.

The most fundamental construct in an HLA application is the concept of a *federation*, i.e., a set of software applications that interact under a common object model and RTI to form a unified simulation environment. Each member of the federation is called a *federate*, which represents a software application with a single point of attachment to the RTI and can be a simulation, a runtime tool, or an interface application to a live participant in the federation (Fig. 1).[2]

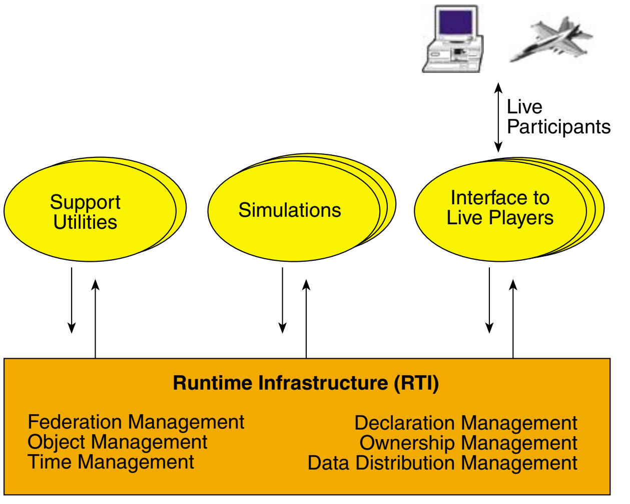

Figure 1. High Level Architecture showing attachment of federates to the runtime infrastructure.

HLA OBJECT MODEL TEMPLATE

The OMT defines a standardized structural framework for describing the information model conveyed in a SOM or FOM. The OMT, as an essential component of the HLA,

- Provides a commonly understood mechanism for specifying the exchange of data and general coordination among members of a federation
- Provides a common standardized mechanism for describing the capabilities of potential federation members
- Facilitates the design and application of common tool sets for the development of HLA object models

The concept of an object model is critical for achieving the ambitious interoperability and reuse goals identified for the HLA. HLA object models can take one of two forms. An HLA SOM defines the information that individual federates can produce or consume when assuming a role in an HLA federation. This information is described according to

- A set of objects that a federate can model or represent internally
- A set of object attributes whose values collectively define the state of each object at any point during execution
- A set of interactions that define the events that a federate can generate or react to during an HLA execution
- A set of parameters that define the important characteristics of each interaction

The second type of HLA object model, a FOM, defines the information exchanged among federates at runtime in support of a specific application. This normally represents the union of the subsets of each federate's SOM that applies to the application at hand, although the need for a common set of shared data characteristics across the federation may require some modifications to the HLA interface of one or more federates. Note that the basic categories of information used to define a SOM are identical to those used to describe a FOM.

In an HLA federation execution, it is through the runtime exchange of object attributes and interactions defined in the FOM that HLA federates can interoperate with one another. Depending on their role in the federation, individual federates can elect to model certain types of FOM objects themselves (sharing the appropriate object state information with other federates via object attribute updates), and/or can elect to receive object attribute updates produced by other federates and reflect the state of these externally modeled objects in their own local simulation environments. This is referred to as *publishing* and *subscribing*, respectively, which also works in an analogous fashion for interactions. Federate subscription requirements and publication responsibilities must be carefully considered during FOM design in order to reduce unnecessary network traffic and relieve federates of having to continually discard irrelevant data.

The HLA OMT defines a tabular format for describing both SOMs and FOMs. A short description of each OMT table is provided below.[3] Note that these table descriptions reflect the V1.3 specifications of the HLA OMT, the current DoD standard.

- **Object Model Identification Table.** This table provides a means of specifying object model metadata for the purpose of facilitating federate and object model reuse. Examples of the types of information supported by this table include the name and version of the object model, the creation date, the sponsoring agency, and relevant points of contact.
- **Object Class Structure Table.** This table provides a means of identifying the object classes that are supported by the federate or within the federation, along with their class/subclass relationships. Each class in this table is associated with an appropriate Publish/Subscribe (P/S) designator to indicate the capabilities of the federate with respect to that class or how the class is utilized in the federation.
- **Attribute Table.** This table provides a means of describing the characteristics of HLA object attributes. Examples of these characteristics include datatype, units, resolution, accuracy, and update rate.
- **Interaction Class Structure Table.** This table provides a means of identifying the interaction classes that are supported by the federate or within the federation, along with their class/subclass relationships. Each class is associated with an appropriate Initiate/Sense/React (I/S/R) designator to indicate the ability of a federate to initiate the interaction, receive or sense the interaction (without reacting), or react to the interaction by modifying the state of affected objects. These designations also apply to federations.
- **Parameter Table.** This table provides a means for describing the characteristics of HLA interaction parameters. Examples of these characteristics include datatype, units, resolution, and accuracy.
- **Enumerated Datatype Table.** This table provides a means of identifying the enumerated values and associated integer representations that define an enumerated datatype.
- **Complex Datatype Table.** This table provides a means of defining the fields (and field characteristics) of complex data structures.
- **Routing Space Table.** This table provides a means of defining the characteristics of routing spaces, the fundamental concept that underlies the HLA Data Distribution Management (DDM) services (services that reduce the volume of data delivered to federates at runtime).
- **FOM/SOM Lexicon.** This set of four tables provides a means of defining all of the object classes, object attributes, interaction classes, and interaction parameters identified anywhere in the object model.

IEEE STANDARDIZATION

Although the HLA was originally developed for DoD applications, it has long been recognized that any community using distributed simulation in its corporate operations (including such businesses as medicine, transportation, and entertainment) can benefit from the HLA. Owing to the desire to make HLA technology available to these communities, an effort was initiated in 1997 to migrate the HLA from a DoD standard to a commercial standard. An existing standards organization, the Institute of Electrical and Electronics Engineers (IEEE), was chosen as the vehicle for producing this commercial standard.

Figure 2 illustrates the organizational structure for the standardization process.[4] At the base, the HLA Working Group is composed of a group of dedicated

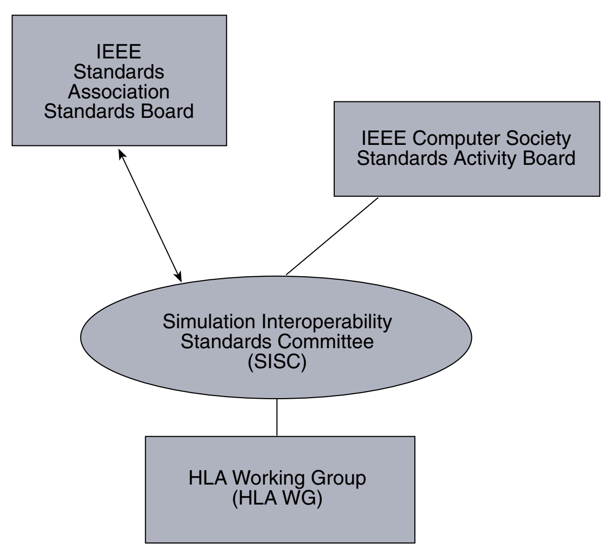

Figure 2. Organizational structure for the IEEE standardization process.

technical professionals responsible for managing drafts of each of the three HLA standards. This group reports to the Simulation Interoperability Standards Committee (SISC), which is charged with developing, coordinating, and maintaining all IEEE standards in the area of distributed simulation. The SISC works with two different but related IEEE boards. The first of these is the IEEE Standards Association Standards Board. The Standards Association is the umbrella organization under which the full range of IEEE standards activities is carried out. The Standards Board approves the initiation of new standards projects and reviews IEEE standards projects for consensus, due process, openness, and balance.[5] The second board is the IEEE Computer Society Standards Activity Board. The Computer Society is chartered by IEEE to advance the theory, practice, and application of computer and information processing science and technology. The Standards Activity Board is responsible for all standards activities in the Computer Society.[6]

The Process

The IEEE standardization process consists of seven major steps. The following describes the activities that have been completed thus far in the process.

Step 1: A Project Authorization Request (PAR) is submitted to the Standards Board for approval to proceed with the development of a new standard or to modify an existing standard. In the case of the HLA, it was decided that each of the three HLA components required its own PAR. The three PARs were formally submitted to the Standards Board and approved in December 1997.

Step 2: A draft of the standard is developed and forwarded to the sponsor with a recommendation that it be "balloted." In this case, the SISC was the sponsor of the standard, and the HLA Working Group was the body tasked to develop the draft. The development period began in February 1998 with an open review of the draft baseline (V1.3) by potential users of the standard. Comments were formally submitted against the initial draft to enhance the specification and/or identify problem areas and potential solutions. These comments were then discussed within the working group, and all accepted comments were addressed in a revised version of the draft. This general process was then repeated three more times in the following months, with open reviews driving needed revisions that were then incorporated into an evolving, maturing draft. The development period ended in January 1999 with the fourth draft of the HLA specifications. The working group, at that time, unanimously recommended that the specifications be delivered to the sponsor for formal balloting. Approval to move forward with the balloting process was granted at the March 1999 meeting of the SISC.

Step 3: The sponsor develops a management approach for ballot resolution. In this case, the SISC delegated the responsibility for resolving ballot comments to a Ballot Resolution Committee (BRC) for each of the three HLA specifications. The BRC represents a knowledgeable and dedicated group of volunteers who interact with the balloters to address and resolve technical issues raised through the balloting process. A proposal for the composition of the BRCs was presented and approved at the March 1999 SISC meeting.

Step 4: An appropriate balloting group is formed. This activity began with an open invitation to various HLA user organizations to participate in the balloting. Those individuals that responded affirmatively (and were members of the IEEE Standards Association) became members of one or more of the three HLA balloting groups. Each balloting group was then examined to ensure that there was a proper balance of interests and that no one interest category (user, producer, academic, or general interest) dominated over any other. The composition of the three balanced balloting groups was finalized in May 1999.

Step 5: The next activity is the balloting itself. It began in June 1999, with a mass mailing of the draft standards and appropriate balloting instructions to the members of each balloting group. Each balloting group member was asked to review the draft standard and categorize each comment as (1) an editorial change, i.e., a minor editorial comment that would not influence the approval or disapproval of the specification; (2) a content change, i.e., a minor technical comment that would not influence approval or disapproval; and (3) an objection, i.e., a major technical comment that would result in the disapproval of the specification. The "pass" criterion for an IEEE standard requires a 75% return of the ballot. Of that 75% returned, 75% must vote in the affirmative. Table 1 shows the results of the balloting for the HLA specifications. Note that although the ballot was valid, none of the specifications passed on the first ballot.

Step 6: Technical issues raised by the balloters are addressed and resolved for each HLA specification. This activity began in July 1999 with the closing of the initial balloting period. The chair of each BRC first assigned responsibility for the committee's response to each comment to an appropriate BRC member. Once the full set of proposed comment responses was collected from all members, the BRC chair was charged with ensuring that the responses as a whole were complete,

Table 1. HLA balloting results.

HLA specification	% returned	% affirmative
HLA Rules	90	71
HLA Interface Specification	91	50
HLA OMT	90	52

consistent, and not in conflict with the comment responses produced by the other BRCs. Responses were then sent to the original balloters.

At this point in the process a period of open negotiation and discussion between the individual balloters and the appropriate BRCs commenced. The primary goal of this two-way correspondence from the BRC perspective was to reach an agreement with each balloter on the resolution of his or her comments and, as a result, reverse as many negative ballots as possible (while maintaining the overall cogency of the specifications). Once each of the BRCs felt that the issues raised by their balloters had been properly addressed, the negotiated resolution of each comment was incorporated into a revised draft of the appropriate specification. This revised draft is now in the process of being recirculated to the original ballot group for a revote. The same balloting rules apply, and comments are allowed only on the changed portions of the specifications.

IEEE P1516.2

Before discussing the last step of the IEEE standardization process, we now examine how it has influenced the form and content of the OMT. We focus first on modifications to existing OMT V1.3 tables, then introduce new tables that have been incorporated into the evolving IEEE standard, and finally briefly mention the deleted tables.[7]

Modified Tables

This first series of tables illustrates modifications that have been incorporated into IEEE OMT Draft 5. New table features are highlighted in blue.

Object Model Identification Table

Category	Information
Name	<name>
Type	<type>
Version	<version>
Modification Date	<date>
Purpose	<purpose>
Application Domain	<application domain>
Sponsor	<sponsor>
POC	<poc>
POC Organization	<poc organization>
POC Telephone	<poc telephone>
POC Email	<poc email>
References	<references>
Other	<other>

Two new rows have been added to the table to support the identification of reference sources and any other information relevant to the reuse of the federate.

Object Class Structure Table

HLAobject Root (<p/s>)	[<class> (<p/s>)]	[<class> (<p/s>)]	[<class> (<p/s>)]	...	[<class> (<p/s>)]
		[<class> (<p/s>)]	[<class> (<p/s>)]	...	[<class> (<p/s>)]
			⋮	...	⋮
			[<class> (<p/s>)]	...	[<class> (<p/s>)]
		[<class> (<p/s>)]	[<class> (<p/s>)]	...	[<class> (<p/s>)]
			⋮	...	⋮
			[<class> (<p/s>)]	...	[<class> (<p/s>)]
		⋮	⋮	...	⋮
	[<class> (<p/s>)]	[<class> (<p/s>)]	[<class> (<p/s>)]	...	[<class> (<p/s>)]
			[<class> (<p/s>)]	...	[<class> (<p/s>)]
			⋮	...	⋮
	⋮	⋮	⋮	...	⋮

The required root class of all HLA object class hierarchies (HLAobjectRoot) has been made explicit in this table, and the optional reference to a continuation table that was previously supported in the last column of the table (in OMT V1.3) has been removed.

Attribute Table

Object	Attribute	Datatype	Update Type	Update Condition	D/A	P/S	Available Dimensions	Transportation	Order
HLA object Root	HLA privilege ToDelete Object	<datatype>	<update type>	<update condition>	<d/a>	<p/s>	<dimensions>	<transport>	<order>
<object class>	<attribute>	<datatype>	<update type>	<update condition>	<d/a>	<p/s>	<dimensions>	<transport>	<order>
	<attribute>	<datatype>	<update type>	<update condition>	<d/a>	<p/s>	<dimensions>	<transport>	<order>
	<attribute>	<datatype>	<update type>	<update condition>	<d/a>	<p/s>	<dimensions>	<transport>	<order>
<object class>	<attribute>	<datatype>	<update type>	<update condition>	<d/a>	<p/s>	<dimensions>	<transport>	<order>
	<attribute>	<datatype>	<update type>	<update condition>	<d/a>	<p/s>	<dimensions>	<transport>	<order>
	<attribute>	<datatype>	<update	<update	<d/a>	<p/s>	<dimensions>	<transport>	<order>

The one required attribute of HLAobjectRoot (HLAprivilegeToDeleteObject) has been made explicit. The "Cardinality," "Units," "Resolution," and "Accuracy" columns from V1.3 have been removed, as this information is now captured in the new datatyping tables, and "Accuracy Condition" has been removed as unnecessary. For consistency with the evolving Federate Interface Specification (P1516.1), "Routing Space" has been replaced by "Available Dimensions," "T/A" (Transfer/Accept) has been replaced by "D/A" (Divest/Acquire), and "U/R" (Update/Reflect) has been changed to "P/S" (Publish/Subscribe). Finally, two new columns were added to support the identification of transportation and delivery order types for object attributes.

Interaction Class Structure Table

HLAinteraction Root (<p/s>)	[<class> (<p/s>)]	[<class> (<p/s>)]	[<class> (<p/s>)]	...	[<class> (<p/s>)]
		[<class> (<p/s>)]	[<class> (<p/s>)]	...	[<class> (<p/s>)]
			⋮	...	⋮
			[<class> (<p/s>)]	...	[<class> (<p/s>)]
		[<class> (<p/s>)]	[<class> (<p/s>)]	...	[<class> (<p/s>)]
			⋮	...	⋮
			[<class> (<p/s>)]	...	[<class> (<p/s>)]
		⋮	⋮	...	⋮
	[<class> (<p/s>)]	[<class> (<p/s>)]	[<class> (<p/s>)]	...	[<class> (<p/s>)]
			[<class> (<p/s>)]	...	[<class> (<p/s>)]
			⋮	...	⋮
	⋮	⋮	⋮	...	⋮

The required root class of all HLA interaction class hierarchies (first column) has been made explicit, and the optional reference to a continuation table previously supported in OMT V1.3 has been removed. Also, the I/S/R designation for interaction classes has been replaced by the same P/S designations used for object classes.

Parameter Table

Interaction	Parameter	Datayype	Available Dimensions	Transportation	Order
<interaction class>	<parameter>	<datatype>	<dimensions>	<transportation>	<order>
	<parameter>	<datatype>			
	<parameter>	<datatype>			
<interaction class>	<parameter>	<datatype>	<dimensions>	<transportation>	<order>
	<parameter>	<datatype>			
	<parameter>	<datatype>			

Changes are similar to the Attribute Table. "Cardinality," "Units," "Resolution," "Accuracy," and "Accuracy Condition" have been removed. "Routing Space" has been replaced by "Available Dimensions," and two new columns, "Transportation" and "Order," were added to support the identification of transportation types and deliver order types for interactions.

New Tables

The following new tables have been incorporated into IEEE OMT Draft 5. An overview of each is provided. As their titles imply, several of these tables relate to datatyping. These were designed to provide more flexibility and datatyping options to federation developers. Note that several of the datatyping tables are preloaded with standard entries. While these entries must appear in all HLA object models, users are not required to use them in their applications.

Basic Data Representation Table

Name	Size in Bits	Interpretation	Endian	Encoding
HLAinteger16BE	16	Integer in the range $[-2^{15}, 2^{15} - 1]$	Big	16-bit two's complement signed integer[a]
HLAinteger32BE	32	Integer in the range $[-2^{31}, 2^{31} - 1]$	Big	32-bit two's complement signed integer[a]
HLAinteger64BE	64	Integer in the range $[-2^{63}, 2^{63} - 1]$	Big	64-bit two's complement signed integer[a]
HLAfloat32BE	32	Single-precision floating-point number	Big	32-bit IEEE normalized single-precision format (see IEEE Std. 754-1985)
HLAfloat64BE	64	Double-precision floating-point number	Big	64-bit IEEE normalized double-precision format (see IEEE Std. 754-1985)
HLAoctetPairBE	16	16-bit value	Big	Assumed to be portable among hardware devices
HLAinteger16LE	16	Integer in the range $[-2^{15}, 2^{15} - 1]$	Little	16-bit two's complement signed integer[a]
HLAinteger32LE	32	Integer in the range $[-2^{31}, 2^{31} - 1]$	Little	32-bit two's complement signed integer[a]
HLAinteger64LE	64	Integer in the range $[-2^{63}, 2^{63} - 1]$	Little	64-bit two's complement signed integer[a]
HLAfloat32LE	32	Single-precision floating-point number	Little	32-bit IEEE normalized single-precision format (see IEEE Std. 754-1985)
HLAfloat64LE	64	Double-precision floating-point number	Little	64-bit IEEE normalized double-precision format (see IEEE Std. 754-1985)
HLAoctetPairLE	16	16-bit value	Little	Assumed to be portable among hardware devices
HLAoctet	8	8-bit value	Big	Assumed to be portable among hardware devices
<name>	<size>	<interpretation>	<endian>	<encoding>
<name>	<size>	<interpretation>	<endian>	<encoding>

[a]The most significant bit contains the sign.

This table provides a means of describing the fundamental building blocks upon which all other datatypes are defined. Encoding information is provided for each basic data representation.

Simple Datatype Table

Name	Representation	Units	Resolution	Accuracy	Semantics
HLAASCIIchar	HLAoctet	NA	NA	NA	Standard ASCII character (see ANSI Std. X3.4-1986)
HLAunicodeChar	HLAoctetPairBE	NA	NA	NA	Unicode UTF-16 character (see Unicode Standard, V3.0)
HLAbyte	HLAoctet	NA	NA	NA	Uninterpreted 8-bit byte
<simple type>	<representation>	<units>	<resolution>	<accuracy>	<semantics>
<simple type>	<represenatation>	<units>	<resolution>	<accuracy>	<semantics>

This table provides a means of associating an essential set of information (name, units, etc.) with a basic data representation from the previous table. Through this association, the level of definition necessary to directly utilize the datatype in describing HLA data elements is provided.

Enumerated Datatype Table

Name	Representation	Enumerator	Values	Semantics
HLAboolean	HLAinteger32BE	HLAfalse	0	Standard boolean type
		HLAtrue	1	
<enumerated type>	<representation>	<enumerator 1>	<value(s)>	<semantics>
		...	...	
		<enumerator *n*>	<value(s)>	
<enumerated type>	<represenation>	<enumerator 1>	<value(s)>	<semantics>
		...	...	
		<enumerator *m*>	<value(s)>	

This table provides a means of describing data elements that can only assume a discrete, finite set of possible values. Columns are provided to specify the name of the datatype, the list of possible enumerations (and associated values), the basic data representation associated with each enumerator value, and the overall meaning of the datatype.

Array Datatype Table

Name	Element Type	Cardinality	Encoding	Semantics
HLAASCIIstring	HLAASCIIchar	Dynamic	HLAveriableArray	ASCII string representation
HLAunicodeString	HLAunicodeChar	Dynamic	HLAvariableArray	Unicode string representation
HLAopaqueData	HLAbyte	Dynamic	HLAvariableArray	Uninterpreted sequence of bytes
<array type>	<type>	<cardinality>	<encoding>	<semantics>
<array type>	<type>	<cardinality>	<encoding>	<semantics>

This table provides a means of describing indexed, homogeneous collections of datatypes. Columns are provided to specify the name of the array datatype, the datatype of each individual array element, the total number of elements in the array, the array encoding, and the overall meaning of the datatype.

Fixed Record Datatype Table

	Field				
Record Name	Name	Type	Semantics	Encoding	Semantics
<record type>	<field 1>	<type 1>	<semantics>	<encoding>	<semantics>
	. . .	. . .	. . .		
	<field *n*>	<type *n*>	<semantics>		
<record type>	<field 1>	<type 1>	<semantics>	<encoding>	<semantics>
	. . .	. . .	. . .		
	<field *m*>	<type *m*>	<semantics>		

This table provides a means of describing heterogeneous collections of datatypes (i.e., complex data structures). Columns are provided to specify the name of the datatype; the name, datatype, and meaning of each field in the structure; and the semantics and encoding for the entire fixed record.

Variant Record Datatype Table

	Discriminant			Alternative				
Record Name	Name	Type	Enumerator	Name	Type	Semantics	Encoding	Semantics
<variant type>	<name>	<type>	<set 1>	<name 1>	<type 1>	<semantics>	<encoding>	<semantics>
			. . .	. . .	. . .	. . .		
			<set *n*>	<name *n*>	<type *n*>	<semantics>		
<variant type>	<name>	<type>	<set 1>	<name 1>	<type 1>	<semantics>	<encoding>	<semantics>
			. . .	. . .	. . .	. . .		
			<set *m*>	<name *m*>	<type *m*>	<semantics>		

This table provides a means of describing data items whose datatypes depend on the value of a discriminant. Columns are provided to specify the name of the datatype; the name, enumerated datatype, and possible enumerator values for the discriminant; the name, datatype, and meaning of each alternative (one alternative per enumerator value); and the meaning and encoding used for the entire variant record.

Dimension Table

Name	Datatype	Dimension Upper Bound	Normalization Function	Value when Unspecified
<dimension>	<type>	<bound>	<normalization function>	<default range/excluded>
<dimension>	<type>	<bound>	<normalization function>	<default range/excluded>
<dimension>	<type>	<bound>	<normalization function>	<default range/excluded>

This table provides a means of describing the dimensions that may be associated with HLA interactions or HLA object attributes when Data Distribution Management (DDM) services are being used by the federate or within the federation. This table may be empty if DDM services are not used.

Time Representation Table

Category	Datatype	Semantics
Time Stamp	<type>	<semantics>
Lookahead	<type>	<semantics>

This table provides a means of defining the time representation abstract datatype (ADT) provided to the RTI when a federate joins a federation. It also provides a means of identifying the ADT specified for “lookahead” values (lookahead is used to increase the “parallelization” of federate processing).

Transportation Type Table

Name	Description
HLAreliable	Provide reliable delivery of data in the sense that TCP/IP delivers its data reliably
HLAbestEffort	Make an effort to deliver data in the sense that UDP provides best-effort delivery
<name>	<description>

This table provides a means of defining the mechanisms used for transporting data between federates. Columns are provided to indicate the name and description of each transportation type. Although two required transporation types are preexisting (HLAreliable and HLAbestEffort), others may be added when an RTI is used that can support them.

User-supplied Tag Table

Category	Datatype	Semantics
Update/Reflect	<type>	<semantics>
Send/Receive	<type>	<semantics>
Delete/Remove	<type>	<semantics>
Divestiture Request	<type>	<semantics>
Divestiture Completion	<type>	<semantics>
Acquisition Request	<type>	<semantics>
Request Update	<type>	<semantics>

This table provides a means of describing the tags that may be supplied with certain HLA services. The first column indicates the HLA service categories for which tags can be provided. The remaining columns are used to define the datatype of each tag and to describe the use of the datatype for that tag.

Synchronization Table

Label	Tag Datatype	Capabilities	Semantics
<label>	<type>	<capability>	<semantics>
<label>	<type>	<capability>	<semantics>
<label>	<type>	<capability>	<semantics>

This table provides a means of describing synchronization points. Synchronization points are used to synchronize federate activities during a federation execution. Columns are provided to indicate the label for each synchronization point, the datatype used for the user-supplied tag (if applicable), the capability of the federate with respect to each synchronization point (for SOMs only), and the usage or overall meaning of the synchronization point.

Switches Table

Switch	Setting
Auto Provide	<auto provide>
Convey Region Designator Sets	<convey region designator sets>
Attribute Scope Advisory	<attribute scope advisory>
Attribute Relevance Advisory	<attribute relevance advisory>
Object Class Relevance Advisory	<object class relevance advisory>
Interaction Relevance Advisory	<interaction relevance advisory>
Service Reporting	<service reporting>

This table provides a means of defining the initial settings of various RTI switches. These switches control whether certain RTI capabilities should be enabled or disabled at the beginning of an HLA federation execution. The first column identifies the names of these switches, and the second column defines the initial setting for each switch.

Deleted Tables

Three OMT V1.3 tables have been removed from IEEE OMT Draft 5. In two, the Enumerated Datatype and Complex Datatype Tables, the information previously specified is now captured in the new datatyping tables (although the Enumerated Datatype Table designation was retained). The third, the Routing Space Table, has been replaced by the new Dimension Table.

STEP 7 AND BEYOND

As of the time this article was written, the results of the ballot recirculation for the revised HLA specifications were still unknown. However, since recirculation cannot occur until a sufficient number of dissenting balloters have agreed to reverse their negative votes (enough to guarantee passage) during the comment resolution period, it is assumed that the ballot will be successful for all three HLA specifications.

Step 7 in the overall process will be for the SISC to forward the HLA specifications to the Standards Association. The Standards Board will then review the specifications development and balloting process, incorporate whatever editorial changes are necessary to be consistent with IEEE style guidelines, and finally approve the specifications as an IEEE standard. The approval process is expected to occur in the fall of 2000.

After the standard is approved, the SISC will assume responsibility for long-term maintenance of the HLA specifications. This task will include working with the user community to identify any technical issues relevant to the standard, and managing the reaffirmation of the standard every 5 years. The HLA AMG must also evaluate the IEEE standard to decide whether it should supercede the existing DoD HLA standard (V1.3). Since the HLA AMG was an active participant in the development of the IEEE standard (via appropriate representation in the balloting groups), it is assumed that this approval will take place.

SUMMARY

This article has provided a summary of activities intended to transition the HLA from a DoD standard to a widely recognized commercial standard. The focus was to illustrate how this process has affected the OMT component of the HLA. For more information, readers may access the Simulation Interoperability Standards Organization (SISO) Web site[8] or consult any of the other reference materials cited throughout the article.

REFERENCES

[1]Under Secretary of Defense for Acquisition and Technology, *Department of Defense Modeling and Simulation Master Plan*, DoD 5000.59-P, Washington, DC (Oct 1995).

[2]Dahmann, J., Kuhl, F., and Weatherly, R., "Standards for Simulation: As Simple as Possible but Not Simpler—The High Level Architecture for Simulation," *Simulation* **71**(6), 378–387 (Dec 1998).

[3]U.S. Department of Defense (DoD), *High Level Architecture Object Model Template Specification, Version 1.3*, Washington, DC (Feb 1998).

[4]Zimmerman, P., and Symington, S., *The IEEE Standardization Process*, HLA Architecture Management Group (AMG) archives (Aug 1999), available at http://hla.dmso.mil/amg/meetings/31-35.html (accessed 3 May 2000).

[5]The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standards Association Operations Manual* (May 1999).

[6]The Institute of Electrical and Electronics Engineers, Inc., *IEEE Computer Society Policies and Procedures Manual* (Jan 1999).

[7]The Institute of Electrical and Electronics Engineers, Inc., *IEEE P1516.2/D5: Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT)* (Apr 2000).

[8]Simulation Interoperability Standards Organization Web site, available at http://www.sisostds.org (accessed 4 May 2000).

ACKNOWLEDGMENTS: The standardization effort described in this article required the combined efforts of many dedicated individuals. The author would like to thank the members of the OMT Balloting Group for their participation (and patience) throughout the balloting process, the OMT Ballot Resolution Committee (BRC) for their invaluable assistance in resolving balloter concerns, the P1516 and P1516.1 specification editors (Mike Lightner and Reed Little) for their help in addressing and coordinating responses to cross-specification issues, and Phil Zimmerman (HLA Program Manager and SISC Chair) for her thoughtful review of this article. The members of the OMT BRC are Deb Fullford (Mak), James Ivers (SEI), Katherine Morse (SAIC), Jeff Nielsen (Mitre), Dave Rosenbaum (GTRI), Roy Scrudder (ARL), Dave Seidel (Mitre), Graham Shanks (Alenia Marconi), and Susan Symington (Mitre).

THE AUTHOR

ROBERT R. LUTZ is a Principal Staff Engineer in JWAD at APL. He has over 19 years of experience in the design, implementation, and evaluation of computer M&S systems for military customers. He received his M.S. degree in operations research from the State University of New York at Stony Brook in 1980. Since joining APL in 1992, Mr. Lutz has assumed leadership roles on several M&S programs, including the Naval Simulation System (NSS), Joint Warfare System (JWARS), and the Simulation Based Acquisition (SBA) initiative. Currently, he is the technical editor for IEEE Standard P1516.2 (Object Model Template), and actively supports several other HLA-related projects for the Defense Modeling and Simulation Office (DMSO). He also serves as the chair of the SISO Federation Development Process Forum and as a guest lecturer in The JHU Whiting School of Engineering. His e-mail address is robert.lutz@jhuapl.edu.