High Level Architecture for Simulation

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

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 motivations for the development of the HLA and the process instituted to develop the architecture well as a technical description of the key elements of architecture and supporting software.

1. Introduction

The High Level Architecture (HLA) provides a common 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.

The HLA is intended to have wide applicability, across a full range of simulation application areas, including training, analysis, and engineering functions, at a variety of levels of resolution. These widely differing application areas include a variety of requirements which had to be 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.

2. HLA development process

The HLA was developed by the US Department of Defense (DoD) based on a process involving Government, academia, and industry. In FY94, the Defense Advanced Research Projects Agency (DARPA) awarded three industry contracts for the definition of a high level architecture. The results of these three contractor efforts were received in January 1995, and a core government and academic team combined the best of those inputs with additional insights from other DoD modeling and simulation (M&S) projects to arrive at the initial definition of the HLA. This definition was presented to the Defense Modeling and Simulation Office (DMSO)-sponsored Architecture Management Group (AMG) on 31 March 1995.

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 the basis for a baseline HLA definition, completed in August 1996 and approved as the standard technical architecture for all US DoD simulations on 10 September 1996. The AMG will continue to evolve the HLA as needed.

The AMG is made up of representatives of major US DoD simulation programs. Programs were selected because they represent the major simulation requirements, ranging from analysis to training to test and evaluation, from detailed engineering representation to campaign-level warfighting to man-in-the-loop interactive vehicle simulators. Members, along with their supporting government and industry teams, participate in regular (every 6 to 8 weeks) meetings to address the key HLA issues, conduct cooperative prototyping efforts with other AMG members (prototype federations and experiments) and participate in technical working groups to evolve crosscutting technical aspects of the HLA.

It is also very important that the HLA be integrated into the broader technical community. Work is underway in a partnership with the IEEE Workshop on Standards for Distributed Interactive Simulation ("DIS Workshop"). The DIS Workshop was represented on the AMG during the HLA baseline development period and has committed to establish industry standards to support the HLA. This effort was initiated at the 14th DIS Workshop in March 1996.

3. Technical architecture

3.1. Motivations

The HLA design has been based on certain assumptions. First, the HLA is based on the premise that no one simulation can solve all the US DoD functional needs for modeling and simulation. The needs of the users are too diverse. The technical complexity of needed implementations is beyond what has been shown to be possible today or is likely in the reasonable future to be handled in a single simulation. Further, with changing user needs, it is just not possible to anticipate how simulations will be used in the future or in which combinations. Perhaps equally as important is the changing state of technology; the HLA will need to allow for leveraging technical innovation.

Since it is not possible to anticipate all the uses of M&S in the US DoD, it is important to think in terms of multiple simulations which can be reused in a variety of ways. This means that as simulations are developed, they must be constructed so that they can be easily brought together with other simulations, to support new and different applications. While the HLA attempts to impose as few constraints as possible on the design of individual simulations, given the drive for reuse, the HLA recognizes that simulations themselves will need to be constructed so that they can easily be extended to support new features needed by unanticipated applications.

These assumptions have affected the HLA design in several ways. Clearly, the architecture itself must have modular components with well-defined functionality and interfaces. Further, the HLA has separated the functionality needed for individual simulations from the infrastructure required for interoperability among simulations.

3.2. 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 range. 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).

The second functional component is the RTI. The RTI is, in effect, a distributed operating system for the federation. The RTI provides a 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 C2 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.
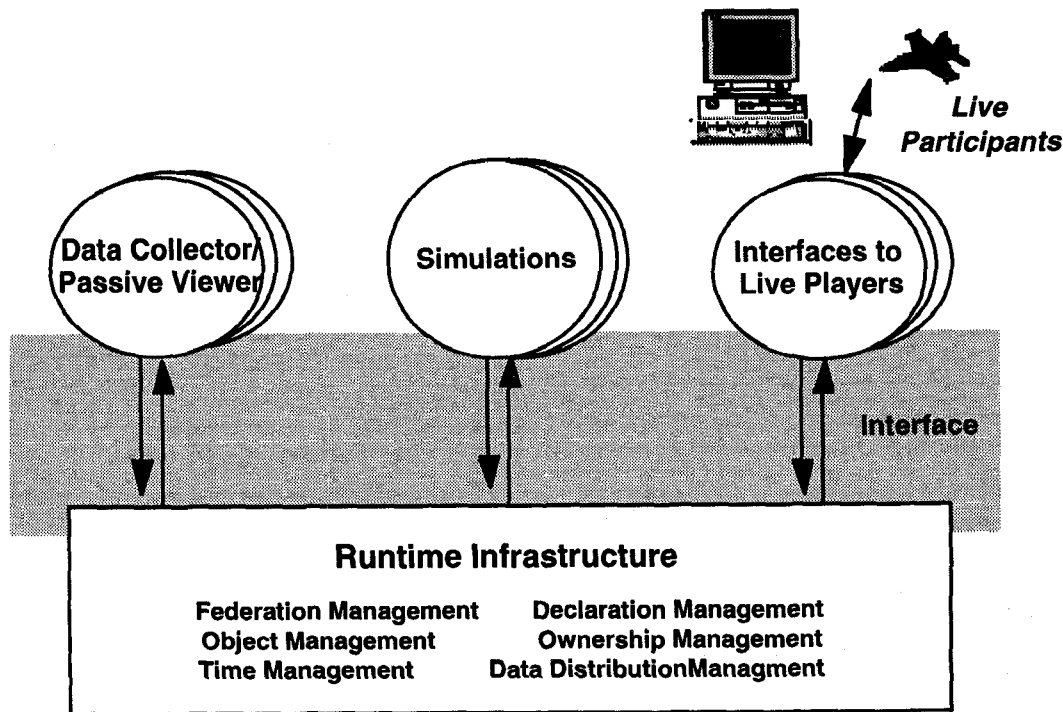
Figure 1. Functional view of an HLA federation

The HLA is formally defined by three components: the interface specification [1], the object model template [2], and the HLA rules [3].

3.3. HLA interface specification

The HLA interface specification 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* supports the dynamic transfer of ownership of object/attributes during an execution. *Time management* services support 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 a programmer's interface.

3.4. 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. 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).

4. HLA rules

Finally, the HLA rules summarize the key principles behind the HLA. These 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 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.

5. HLA prototyping

The purpose of the HLA baseline development period was to take the initial HLA definition to develop the needed specifications and tools to actually use the HLA to support prototype applications. This prototyping was directed toward addressing a set of common issues [4] and to evolve the HLA to meet the broad set of needs. These issues include the technical viability of a single interface specification to support the wide application base of the HLA and the technical feasibility of building an RTI with the necessary range of tools needed by those applications, the impact of the HLA on simulation internal development, the utility of the object model concept and formats throughout the HLA life cycle, the ability to specify common testing methods, the ability to operate HLA federations in a secure mode and, finally, the applicability of the HLA to address the wide range of US DoD simulation applications.

There were four major prototype implementations of federations ("proto-federations") using the HLA to support the HLA baseline development process. The simulations incorporated in these proto-federations were made available, along with their support technical development teams, by AMG members, as part of the HLA development process. Each proto-federation addressed a different corner of the simulation application domain, and as such, generally typifies one way simulations will be used in an HLA context.

Several overall points are warranted before the proto-federations themselves are described. The purpose of these prototypes was to develop the HLA though hands-on experience, not to solve a particular real problem. In several cases the scenarios for the federations had to be stretched to support the mix of federates available from the AMG. Second, all four prototypes used the same architecture, the same interface specification, the same object model template and, in fact, the same RTI software implementation.

For HLA baseline development purposes it was decided to develop one prototype implementation of the RTI and use that implementation with all the proto-federations. This was seen as both the most cost-effective approach and a strategy that would allow for the examination of the desirability of having not only a common interface specification, but also one common implementation which could serve the whole community. The primary objective of the RTI prototyping was to provide the other proto-federations with the needed functionality for them to implement and use the HLA for their application needs. The RTI prototype also provided the mechanism to assess the functional and performance requirements for RTI service provision as input to future RTI development and transition. The RTI software is being re-engineered to incorporate the full set of services defined in the HLA Baseline Definition .

The first proto-federation was called the Platform Prototype federation. This was a federation of real-time platform level simulators and simulations which are current users of IEEE DIS 2.0. The key issues addressed by this proto-federation were the ability of the HLA to provide the capabilities now supported by DIS and to address the issues associated with adapting a current DIS-enabled system to work under HLA.

The second federation, called the Joint Training Prototype, was a federation of discrete event simulations. The key issues for this prototype were the ability to coordinate multiple discrete event simulations using the RTI time management services, the ability to hand-off ownership of attributes in a federation and options for providing dynamic environmental effects across a federation.

The third prototype federation was the Analysis Proto-federation. This proto-federation is comprised of closed form analytic applications; hence, runtime efficiency and replicability are important here, since these simulations will eventually require multiple executions for analysis purposes. The interest in this proto-federation was in the time management and data management services.

The Engineering Prototype federation examined use of the HLA to support simulations to support system acquisition using validated, engineering-level simulations. This proto-federation was particularly interested in object ownership management which affords the option of handing off computation of certain effects during select parts of a scenario to selected simulation facilities and in performance requirements to support hardware-in-the-loop applications.

These prototypes were implemented to apply the HLA to applications which typify its ultimate uses to support its definition. The results of the prototypes were fed into the specification and testing procedures through the working groups. The experiences of the individual federate developments support assessment of the effect of the HLA on simulations and offer an experience base for future users of the HLA.

Finally, the experience of developing these proto-federations provided the basis for the development of a general process view of the HLA. While HLA development focuses on the specification of the technical components of the architecture, it has been recognized that it is very important to understand the process which supports the use of the architecture for different applications. The baseline definition of the HLA was developed based on a set of proto-federations. Using the experience of these proto-federations as a driver, a general process for the development and execution of applications using the HLA is being formulated as a recommended practice to accompany the HLA definition [5]. Understanding this process is viewed as a very important element in the development and application of the HLA.

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." [6] 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 with 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.

The HLA rules, interface specification, and object model template provide minimum essential tools for interoperability. 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. HLA technical library

DMSO has established an on-line HLA Technical Library, accessible through the DMSO Home Page on the World Wide Web (http://www.dmso.mil), which contains not only the HLA Baseline Definition documents, but a wealth of supporting documentation as well. This includes test procedures, design details, an HLA security architecture, examples of the use of the HLA, and an extensive archive of briefings and documents, including all

of the AMG proceedings throughout the HLA prototyping phase and published HLA papers. The HLA Technical Library will continue to grow and evolve as the HLA itself is carried forward. It is intended to be an open source for use by implementors of the HLA.

8. References

[1] "High Level Architecture Interface Specification, V1.0" Defense Modeling and Simulation Office, 15 August 1996 (http://www.dmso.mil).

[2] "High Level Architecture Object Model Template, V1.0" Defense Modeling and Simulation Office, 15 August 1996 (http://www.dmso.mil).

[3] "High Level Architecture Rules, V1.0" Defense Modeling and Simulation Office, 15 August 1996 (http://www.dmso.mil).

[4] "High Level Architecture Baseline Development Plan, V1.7" Defense Modeling and Simulation Office, 1 April 1996 (http://www.dmso.mil).

[5] "HLA Federation and Execution Process Model, V1.0" Defense Modeling and Simulation Office, 6 September 1996 (http://www.dmso.mil).

[6] "Department of Defense Directive on Modeling and Simulation Management (5000-59)" US Department of Defense, 4 January 1994.