The High Level Architecture and Beyond: Technology Challenges

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

Abstract

The High Level Architecture (HLA) provides the specification of a software architecture for distributed simulation. 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.

The HLA was developed over the past three years. It is currently in the process of being applied with simulations developed for analysis, training and test and evaluation and incorporated into industry standards for distributed simulation by both the Object Management Group and the IEEE. This paper provides a discussion of key areas where there are technology challenges in the future implementation and application of the HLA.

1. Introduction

The High Level Architecture (HLA) provides a general software architecture for distributed 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 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. Technical architecture

Figure 1 shows how an HLA federation partitions into its major functional components. The first key components 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. 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.

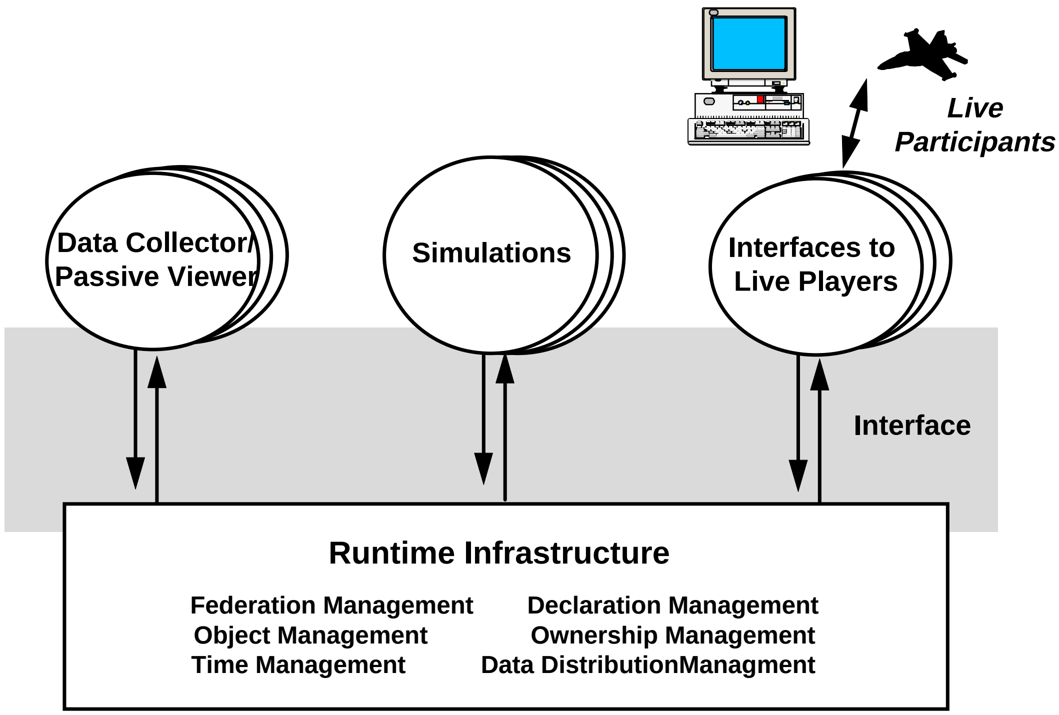

Figure 1. Functional view of an HLA federation

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

2.1. 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.

2.2. 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 specification. These specifications are intended to be the means for open information sharing across the community to facilitate reuse of simulations. 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).

2.3. 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.

3. Technical status and challenges

HLA development began in 1995 and since that time substantial progress has been made in the maturation of the technical specification and implementation of supporting software and user tools. Several implementations of RTI software have been developed by government, commercial organizations and universities in both the US and overseas [4]. The government freeware version of the RTI has been downloaded by over a thousand organizations worldwide. Interfaces to HLA have been integrated into both academic and industrial simulation development environments [5].

At the same time, a formalized process view of the development of HLA federations has been created [6], and this process view has provided the basis for identifying opportunities for application of automation to the use of HLA and has spawned a suite of both government and commercial products. Use of the HLA for training simulations [7], for analytic applications [8] and for support of test and evaluation applications [9] as well for interfaces between simulations and live systems [10], demonstrates the flexibility and broad applicability of the architecture.

Finally, HLA is becoming an open industry standard. In December 1998, the HLA interface specification and RTI services were adopted by the Object Management Group as the Facility for Distributed Simulation 1.0 [11] and the HLA draft IEEE standard 1516 [12] is in the balloting process.

So with the advent of the HLA and with this progress, what are the technical challenges facing the technical community?

Just as HLA provides a framework for reuse of simulations, it also provides a framework for technology development. It is important to reiterate that HLA is an architecture, not an implementation. To date the interest and use of HLA has been primarily with simulations which are workstation based operating over IP networks. The current implementations have focused on support of this class of HLA users. While the use of HLA to date supports the idea that this is a substantial group, it is important to realize that as an architecture HLA is not limited to this implementation area. Attention was paid in the HLA specification process to avoid limiting the possible application of HLA through it design. As interest in HLA grows, new areas for implementation and application have been identified and work is beginning to explore these. With these new areas come new technical challenges. These challenges form the beginning of a technical agenda for the next phase of HLA development process.

3.1. HLA and communications

Current HLA implementations have been workstation-based and have been distributed over IP networks. As an implementation independent architecture, these initial implementation choices reflect the current, immediate application orientation, rather than anything inherent in HLA. Initial research results demonstrate that HLA can be implemented in other communication environments [13] and begin to open the opportunities to extend the use of HLA to other communications environments.

HLA beyond networks. Implementation of HLA federations in shared memory and multi-processor environments is a natural extension for HLA implementation. Some initial work has indicated that multi-processor based simulation engines have implemented a number of the services required in an HLA RTI suggesting the ready opportunity for HLA transition to these computing environments [14]. Use of HLA here offers the promise of promoting the use of these other communications/computing environments to the growing community of simulation developers and users. By offering a common software interface (HLA interface specification) for both workstation and other environments, users gain the flexibility to use the computing environment most applicable or accessible to them without impact on their application software (federate). From a technical point of view, different communications/computing environments offer new possibilities for implementation of key collaborative services in HLA (time and data distribution management). By taking advantage, for instance, of accessible shared state, new options for efficient time management and interest management are possible. Further challenges exist, however, when infrastructure is envisioned which spans different communications/computing environments, allowing for federations that incorporate both networked workstation components and non-networked federates.

HLA on the web. Work has already begun on 'WebHLA' concepts [15], linking the HLA concepts with the concepts of the worldwide web. Using HLA in the web environment is a very natural application. Already, Internet games using HLA are beginning to appear based current infrastructure implementations. As with implementation of HLA in shared computing environments, implementation of HLA in broadly distributed environments opens new possibilities and challenges for collaborative services and RTI implementations. Further, following the adage of 'standards are as standards do', as worldwide web protocols develop, HLA needs to incorporate these into implementations.

Active networks. As network components become more computationally flexible and proficient, opportunities exist for pushing functionality now resident in application software in the RTI down into the network infrastructure. This might include functionality supporting both time management and data distribution management services. HLA specifications establish the interface between the applications (federates) and the runtime infrastructure and specify the services to be provided to the federates, but not how or where those services would be performed. Using network infrastructure to support RTI services to improve efficiency is an area with potential opportunities to both simplify infrastructure and increase efficiency.

3.2. RTI implementation

A small number of production RTI implementations exist and a growing number of research activities [RTI Kit] are beginning which will allow for new initiatives among the research community in algorithms and implementation approaches to providing RTI services. The performance of the current implementations has been very promising and appears to be adequate for the current generation of HLA applications [16]. But as HLA use expands, demands for improved RTI performance will grow and opportunities for application of technology to achieve performance gains will increase. Much gain will likely come from increased computational and communication capabilities, but efficient RTI services will be an important part of the overall performance equation. Further, as is suggested in the preceding discussion, moving RTI implementation into different communications and computing environments opens possibilities and poses new issues for implementation of infrastructure services.

Time management services. Distributed state synchronization algorithms have long been a topic of research in the distributed simulation community. HLA provides a venue for further development, implementation and testing of these and new algorithms in a range of different application and implementation environments. Further, HLA provides a common framework which supports both the 'real-time' networked simulator community and the more traditional 'as fast as possible' discrete event simulation community. HLA federations are increasingly combining these into common applications. This opens consideration of time management algorithms which may be better suited to such applications which are seeking to maintain close mapping between logical simulation time and wall clock time, not traditionally an objective of these algorithms. Finally, as HLA reaches out to a range of different classes of users, a wider range of time management services can be envisioned [17], where event synchronization strategies short of full ordering may be sought out, as users seek an application-based balance between efficiency and correctness. These stresses of maintaining computational correctness (and computability) with efficiency capture the nature of the technical challenges here.

Data distribution management services. As with time management services, issues of data distribution have a research topic in distributed simulation. Again HLA provides a ready, broad based avenue for further development, testing, and application of effective technical solutions. Here, new ideas in the arena of 'mobile' computing may find promising application. Further, the notion of widely distributed, web-based HLA applications, opens wide the options (and needs) for new strategies for data distribution.

Ownership management services. In HLA different attributes of the same object may be computed in different federates and the responsibility to compute these attributes may be exchanged dynamically at runtime. In most HLA applications this 'attribute ownership management' service has been employed to allow for different representational techniques to be selectively applied at different phases of a simulation application. Beyond this, these services have other potential uses in the areas of fault recovery and load balancing. As these application ideas mature, so will the technical approaches to efficient implementation of these services.

Fault tolerance. There is currently no formal failure model for the HLA and hence no focused attention to the design of HLA infrastructure, applications or support tools has been devoted to this area. As HLA matures, this will become a more important arena for attention.

3.3. HLA applications

Most current HLA applications are based on retrofitting simulations built for standalone purposes to operate in a federation. This is an artifact of the maturity of the HLA and the fact that it is a relatively rare event for a substantial simulation to be newly developed. With increased acceptance of HLA and the passage of time, we can expect this to change. New simulation systems will be created as distributed applications with the HLA services as integral part of the design premises. Applications will begin distributing computational elements (scripts, applets, programs, URLs, media) as well as traditional attribute state updates, allowing for more efficient, more dynamic and flexible HLA applications. These applications will pose new implementation challenges for the RTI since it is unlikely that needs for efficiency will diminish as the variety of uses expands. Promotion of new and extended uses of HLA by the advanced simulation technical community is important to keep the pressures for infrastructure innovation ahead of production uses.

'Composability'. HLA distributed system applications are beginning to be developed as federations (rather than standalone simulations) with clear roles for federates as a primary design objective. In some cases, suites of federates will be developed with longer term, broader based common use applications in mind. One can envision such families of federates being developed design with a 'mix and match' capability in mind, as a set of basic components which could be used in different combinations to achieve different goals. In this way one could reasonably imagine that the concept of so-called 'plug and play' composability could become a reality, if this composability were 'designed' into a suite of federates.

Beyond the practical process of design suites of federates as part of a 'product line' approach, what is the potential for broader based or more 'opportunistic' composability? This question raises the general technical issue of composability. What are "necessary and sufficient" conditions to achieving composable models? What constraints on model design are required to achieve composability? Realistically, does one need to become an expert in each of the simulation frameworks / languages used in the federation to be able to effectively create and use a composed simulation model? Or are there ways to design and document federates to allow for easy use without requiring detailed understanding of internal implementation details?

Distributed system management applications. HLA is designed and used as a runtime environment. In a number of cases, HLA is now being incorporated into the infrastructure of standing simulation activities in which a suite of simulations is maintained and used over time. Different simulations are incorporated into federation executions based on the needs of the immediate application. These so-called 'persistent' federations [18] gain some real economies through local reuse over time of federates, federation execution plans and federation object models. Beyond this however, additional opportunities to apply HLA to the pre and post runtime requirements of federation operations are beginning to be apparent, particularly when there are distributed sites which routinely participate in federation executions. Pre-runtime routines of selecting federates, passing key setup parameters, running through technical startup readiness procedures and controlling startup processes, all need to be handled in a distributed fashion. Rather than creating new infrastructure and sets of services, HLA provides an installed base and service inventory which may be readily applied to these requirements, allowing for a more integrated life cycle support environment based on a common architecture. Experimentation in extension of HLA into these areas, as well as more active use of the management object model [19] facilities in the HLA to support distributed simulation management functions will all be important to the next phase of HLA.

Distributed Virtual Worlds. When you step back and look broadly at the development and use of 'simulations, it becomes apparent that there have been several tracks of development which all come to distributed simulation from different technical and cultural premises. HLA offers an opportunity to bring these together, and to leverage the diverse technical perspectives in new ways.

Among the more traditional simulation communities, discrete event simulation development has traditionally been a very separate community from the real-time simulation development world. With HLA, these two can share a common infrastructure and increasingly participate in common applications under HLA. Further afield, the development of graphics-driven virtual worlds technology has placed a premium on the visualization component of simulation and has backed into more complex simulation of behavior and dynamic interaction. With the advent of the worldwide web, this virtual world community is now seeking the mechanisms to distribute their visual creations and to support dynamic realistic interaction among distributed users. Finally, the advanced distributed education community, is looking to both virtual world and interactive simulation to provide a vehicle to enable long-distance education and training possibilities.

As a general purpose distributed simulation architecture HLA brings a common base to these historically diverse, but technically interrelated areas. A major technical challenge is to step back and understand the technical perspectives brought by each community, and to capitalize on these in an open way to create new simulation applications which go beyond what any one of these communities might bring independently. HLA can be a key enabler of this process by providing a common framework to address technical issues facing the community as a whole.

4. Summary and Conclusion

HLA is increasingly been incorporated into a range of applications based on an acceptance of the architecture by the open standards community and by the commercial and academic community as a effective common architecture for distributed simulation. Current HLA infrastructure implementations have supported a set of distributed simulation applications in a set of commonly used computation and communications environments.

As HLA use grows, there will be new technical challenges in extending HLA implementation into

- broader communications and computing environments, notably web-based implementations along with 'non-networked' (shared memory) environments,
- higher performance and innovative implementations of the runtime infrastructure services, and
- extending HLA to capitalize on application opportunities from diverse areas of traditional simulation and simulator development as well as distributed virtual worlds for entertainment and education.

The HLA provides an excellent framework for technology investment. As a reusable common distributed simulation framework, it has brought into broad use technical capabilities that had had only limited user application. HLA provides a framework for building on these technical capabilities in a common environment which has been lacking across the broad simulation technical community. Finally, HLA provides new opportunities for exchange of ideas among a range of technical perspectives in a common framework, a common environment for investigating a range of ideas, and a ready vehicle for transition of viable technical approaches into common practices.

5. References

[1] "High Level Architecture Interface Specification, V1.3" Defense Modeling and Simulation Office, April 1998 (http://www.dmso.mil).
[2] "High Level Architecture Object Model Template, V1.3" Defense Modeling and Simulation Office, April 1998 (http://www.dmso.mil).
[3] "High Level Architecture Rules, V1.3" Defense Modeling and Simulation Office, April 1998 (http://www.dmso.mil).
[4] Bachinsky, Steve, Frank Hodum and J. Russell Noseworthy. "RTI 1.3 NextGeneration Implementation." Paper 99S-SIW-

118, to be presented at the Spring Simulation Interoperability Workshop, March 1999.
[5]Ziegler, Dr. Bernard P. "Implementation of the DEVS Formalism over the HLA/RTI: Problems and Solutions." Paper 99S-SIW-065, to be presented at the Spring Simulation Interoperability Workshop, March 1999.
[6]"HLA Federation and Execution Process Model, V1.4" Defense Modeling and Simulation Office, 14 January 1999 (http://www.dmso.mil).
[7]DLRC: HLA is being used in the Army Digital Leaders Reaction Course (DLRC) to federate the Army Combat Simulation, EAGLE, with Army Command and Control Systems.
[8]Zabek, Anita Adams. "Joint Warfighting Program Information Superiority Experiment Trailblazer." Paper 98F-SIW-228, presented at Fall Simulation Interoperability Workshop, September 1998.
Feinerman, Laura. "Design of JWP Trailblazer Federation in Support of Analysis." Paper 99S-SIW-055, to be presented at Spring Simulation Interoperability Workshop, March 1999.
[9]Wright, Maj. Darrell. "Test and Evaluation Federation Experience and Lessons Learned Using the Federation Development and Execution Process." Paper 99S-SIW-173, to be presented at the Spring Simulation Interoperability Workshop, March 1999.
[10]Nielsen, Jeff. "Challenges in Developing the JTLS-GCCS-NC3A Federation." Paper 98F-SIW-153, presented at the Fall Simulation Interoperability Workshop, September 1998.
[11]"Facility for Distributed Simulation Systems." OMG Document No. MFG/98-06-06.
[12]IEEE Standard P1516. Draft Framework and Rules. IEEE Standard P1516.1. HLA Federation Interface Specification. IEEE Standard P1516.2. Object Model Template (OMT) Specification. (http://siso.sc.ist.ucf.edu/stdsdev)
[13]Christensen, Paul. "HLA RTI Shared Memory Communications." Paper 99S-SIW-090, to be presented at the Spring Simulation Interoperability Workshop, March 1999.
[14]HPCC SPEEDES Project: A project funded by the DoD High Performance Computing Center (HPCC) program is investigating development of an RTI using existing SPEEDES capabilities as a base.
[15]Furmanski, Dr. Wojtek. "WebHLA – An Interactive Programming and Training Environment for High Performance Distributed FMS." Paper 98F-SIW-216, presented at the Fall Simulation Interoperability Workshop, September 1999.
[16]Bachinsky, Steve. "Defining RTI Performance." Paper 99S-SIW-100, to be presented at the Spring Simulation Interoperability Workshop, March 1999.
[17]Fujimoto, R. M. "Exploiting Temporal Uncertainty in Parallel and Distributed Simulations." Proceedings of the Workshop on Parallel and Distributed Simulation, May 1999.
[18]Dahmann, Dr. Judith. "Persistent Federations." Paper 98S-SIW-059, presented at the Spring Simulation Interoperability Workshop, March 1998.
[19]Fullford, Deborah. "A Federation Management Tool: Using the Management Object Model (MOM) to Manage, Monitor, and Control an HLA Federation." Paper 99S-SIW-196, to be presented at the Spring Simulation Interoperability Workshop, March 1999.