

TRANSITIONING YOUR DIS SIMULATOR TO HLA

by Deb Fullford
Ben Lubetsky
617.876.8085
www.mak.com

Soldiers haven't always been able to train like they fight. In its early days simulation was used only in training for tasks such as piloting an aircraft or operating a specific weapons system. A single soldier would train in a stand-alone, high-fidelity simulator with the goal of learning one particular task. The only way for soldiers to train with the team they fought with was to send hundreds or thousands of troops with equipment to remote locations to conduct live training scenarios.

Distributed Interactive Simulation (DIS) has radically changed the process by which soldiers train for combat. By connecting together many types of simulations into a shared virtual world, DIS dramatically increases the training benefit from simulation. Using DIS, soldiers now train like they fight--in teams.

Less than a decade after its introduction, DIS is maturing into a new generation of software technology, known as the High Level Architecture (HLA). Despite its promise, many members of the simulation community are anxious over the impending transition to HLA. Much of their concern is caused by a lack of understanding of what HLA is, its benefits, and how to transition DIS-compliant simulators to HLA-compliant simulators. This article will provide information and clarification about the technology that will help to alleviate some of those concerns.

History of DIS

In 1983 the Defense Advanced Research Projects Agency (DARPA) sponsored the SIMNET (SIMulation NETworking) program to create a new technology to expand the current single task trainers into networked team trainers. SIMNET was tremendously successful, producing over 300 networked simulators with the technology that was to develop into DIS.

The foundation of DIS is a standard set of messages and rules, called Protocol Data Units (PDUs), used for sending and receiving information across a computer network. The most common message is the Entity State PDU which represents all of the state information about a simulated entity that another simulator needs to know.

For example, an Entity State PDU contains data about an entity's position and velocity. By using the position, velocity, acceleration, and rotational velocity data, a receiver is able to extrapolate, or dead reckon, a vehicles' position before the arrival of the next PDU, thereby reducing consumption of network bandwidth. Us-

ing this technique, DIS is able to limit the amount of data an average simulator transmits to approximately 250 bytes per second. Optimizations, such as dead reckoning, permit very large virtual battles to take place. The largest DIS exercise, part of DARPA's Warbreaker program, had 5,400 simulated entities interacting in a single DIS virtual world.

The fact that there is no central server is perhaps the most surprising DIS characteristic. DIS is strictly a peer-to-peer architecture, in which all data is transmitted to all simulators where it can be rejected or accepted depending on the receivers' needs. By eliminating a central server through which all messages pass, DIS dramatically reduces the time lag needed for a simulator to send important information to another simulator. This time lag, known as latency, can seriously reduce the realism, and therefore the effectiveness, of a networked simulator. For example, it is vital that when one simulator fires at another simulator the target is made aware of the incoming munition as soon as possible to allow it to take the appropriate defensive action. Any delay introduced by the training device results in negative reinforcement to the trainee.

What is HLA?

As DIS matures, the DoD is looking ahead to the next generation of modeling and simulation software that will support a wider range of applications with more functionality. The DoD's Defense Modeling and Simulation Office (DMSO) is leading a DoD-wide effort to establish a common technical framework to facilitate both the interoperability between the wide spectrum of modeling and simulation applications and the reuse of the modeling and simulation components. This common technical framework includes the HLA and is considered the highest priority effort within the DoD modeling and simulation community. Recently, the DIS community has voted that the next generation of DIS will be HLA-compliant

The HLA defines a set of rules governing how simulations, now referred to as federates, interact with one another. The federates communicate via a data distribution mechanism called the Runtime Infrastructure (RTI) and use an Object Model Template (OMT) which describes the format of the data. The HLA does not specify what constitutes an object (objects are the physical things that are going to be simulated, such as tanks and missiles), nor the rules of how objects interact. This is a key difference between DIS and the HLA.

The RTI lets different types of systems interact. These systems can include simulations which run faster than real-time and simulate objects which are hierarchical aggregates of individual entities (platoons, companies, or battalions) all the way to high-fidelity engineering models which run much slower than real time and simulate individual subsystems with very high accuracy. In the DIS paradigm these two applications would not be able to interact.

However, the strength and flexibility of HLA is also its weakness—unless all the HLA simulators in an exercise agree on a single Federate Object Model (FOM) they will not be able to interoperate even though they are HLA compliant. The FOM describes the objects and interactions involved in the federation execution.

Besides facilitating interoperability between simulations, the HLA provides the federates a more flexible simulation framework. Unlike DIS where all simulations receive every piece of data broadcast, the federates now have the ability to specify:

- What information they will be producing
- What information they would like to receive
- The data's transportation service, e.g. reliable, best effort
- Whether or not the federation's timing mechanism is synchronous or asynchronous.

The above points make it possible to have more simulations on a network at one time because the amount of data being sent is reduced. The simulation software is also simplified because it does not need to process extraneous information.

How Does HLA Work?

The two major components of HLA are the Runtime Infrastructure (RTI) and the Object Model Template (OMT). The OMT provides a standard format for describing a simulation in terms of its objects and the interaction between objects. Again, objects are the physical things that are simulated, and interactions are the events that occur in simulations, such as detonations and collisions.

As previously stated, the RTI's primary function is that of a data distribution mechanism. Federates send information through the RTI which distributes the information to the appropriate parties. The RTI does not maintain information about the state of the federation. Nor does it handle any semantics associated with the interaction between the federates, such as what coordinate system to use, what happens during a collision, or how to dead-reckon remote vehicles. Also, the RTI does not specify the exact byte layout of data sent across the network.

The RTI provides a common set of services to the federates. They can be divided into six categories:

- Federation Management: Handles the creation, dynamic control, modification, and deletion of a federation execution.
- Declaration Management: Enables federates to declare to the RTI their desire to generate (publish) and receive (subscribe/reflect) object state and interaction information. Federates can subscribe to only the objects they want (or have the capability) to receive, e.g. tanks might

need only data pertaining to ground movement, or airplanes might need only data pertaining to flight activities

- Object Management: Enables the creation, modification, and deletion of objects and interactions. These services comprise most of the network traffic during runtime.
- Ownership Management: Allows federates to transfer ownership of object attributes to other participants in the simulation.
- Time Management: Provides useful services for setting, synchronizing, and modifying simulation clocks. Time Management services are tightly coupled with the Object Management services so that state updates and interactions are distributed in a timely and ordered fashion.
- Data Distribution Management: Federates can provide conditions governing when to start or stop transmitting and receiving certain pieces of data.

Options for Transitioning Your Simulator from DIS to HLA

As the training and simulation industry moves toward HLA compliance, current DIS simulators will need to be updated in order to keep from becoming obsolete. There are four techniques for making the DIS to HLA transition—translator, wrapper, native, and protocol interface unit (PIU). Some of these techniques are simpler and more cost-effective than others, and each has its own advantages and disadvantages. Table 1 illustrates the benefits of the four approaches.

	Translator	Wrapper	Native	PIU
Forward Compatibility	X	X	X	X
Backward Compatibility	X			X
Ease of use	X			X
Low Latency		X	X	X
Scalability		X	X	X
Takes full advantage of HLA			X	X

Forward compatibility: Technique's ability to be upgraded to newer versions of HLA

Backward compatibility: Technique's ability to switch between HLA and DIS

Easy of use: Requires only limited modifications to existing simulation software

Low latency: Technique does not cause a delay in between sender and receiver

Scalability: Technique's ability to interface with a large number of simulations

Each of the techniques are discussed in detail below:

Translator: Using this technique, a separate application, often another computer, is placed on the network to translate network traffic between the different protocols.

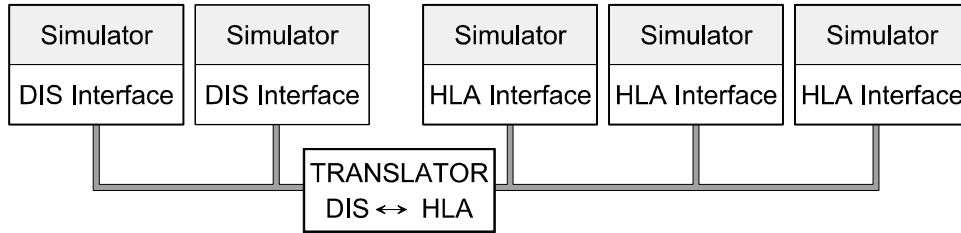


Figure 1. The *Translator* technique requires a separate application or hardware device to manage communications between applications that use different protocols.

A translator requires no software modification to the simulator, but because data must travel through this extra piece of hardware, the simulator's latency increases by roughly a factor of ten. Having all traffic pass through one computer is risky since it puts a single point of failure into an otherwise distributed system. The translator does permit limited forward and backward compatibility, but limits the scalability and flexibility of the simulator. Also, when using a translator, the simulator cannot take advantage of future HLA features.

Wrapper: With a wrapper approach, software is added underneath the simulation's DIS interface to translate the data from the old protocol (DIS) to the new protocol (HLA) just before it is sent and to translate the data from HLA to DIS just after it is received.

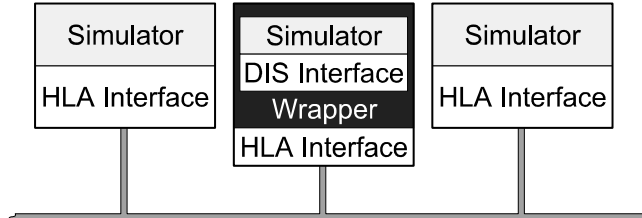


Figure 2. The *Wrapper* technique links additional code to a DIS application to provide interoperability with HLA applications.

Unlike the translator, a wrapper does not require additional hardware. All the changes are made via limited modification to the simulator's software. However, forward and backward compatibility requires further software changes, and like the translator, the wrapper does not allow the simulator to take advantage of HLA specific features.

Native: Creating a native HLA simulation implies that all interfaces to the network are contained within the simulation software.

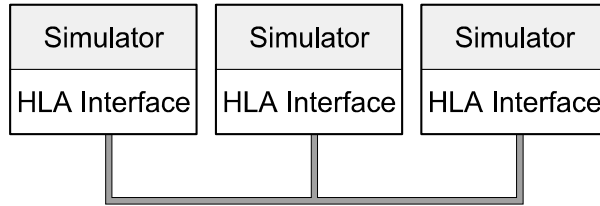


Figure 3. The *Native* technique requires that the simulation software contain all necessary interfaces to the network.

A native HLA simulator can take full advantage of all HLA features. However, these advantages come at the expense of huge software modifications at the initial transition and then additional modifications for any future protocol changes. Also, there is no backward compatibility.

Protocol Interface Unit: The simulation interfaces with the network via a software system known as a Protocol Interface Unit (PIU). A good PIU, such as MÄK Technologies' VR-Link™, will have one API (the simulation's interface) which supports all the features in both protocols, DIS and HLA. A less capable PIU will limit functionality to the lowest common denominator, and the simulator will be unable to take advantage of any features unique to either protocol.

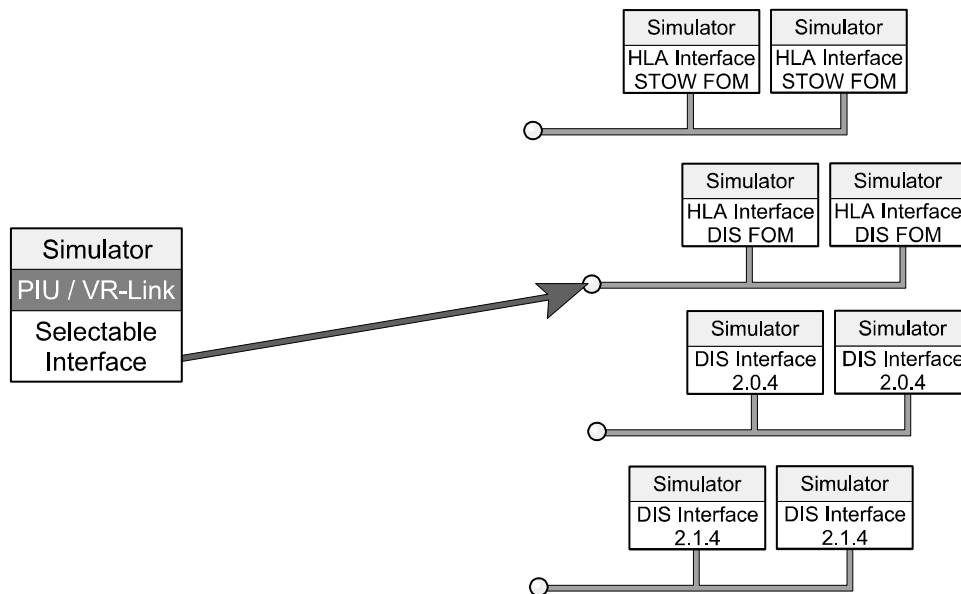


Figure 4. The *Protocol Interface Unit* technique supports all features of both protocols, and allows switching among different FOMs within HLA.

The PIU approach may be the best technique to update a DIS simulator to HLA. It provides an easy upgrade path to HLA, while maintaining backward compatibility with DIS. Using a PIU also permits a simulator to switch among different FOM's within HLA and even different versions of the DIS protocol. A PIU requires only minimal modifications to the simulation software and provides the most

flexibility when designing a new simulation. On the downside PIUs can be complex and expensive to write and maintain.

Conclusion

DIS has proven to be a valuable tool to train soldiers, perform weapons assessment, and conduct mission rehearsals. As DIS moves into the HLA paradigm, it provides an even more powerful and flexible framework for the modeling and simulation community. Over the next four years, as DIS simulations transition to take advantage of the HLA, their value will increase through the integration of a wide range of existing and planned simulations.

We've demonstrated several techniques available to transition DIS-compatible simulations to this new paradigm. One of these, the Protocol Interface Unit, stands out as offering the most flexibility, while incurring the least amount of risk and cost. By using a PIU, such as VR-Link, simulations can maintain their DIS compliance vital to ongoing projects, while providing a clear upgrade path for future HLA simulation programs.

Visit MÄK's web site at www.mak.com

MÄK Technologies provides cutting-edge commercial products for the entertainment and World Wide Web markets through research and development in the Distributed Interactive Simulation (DIS) technology area. MÄK's flagship product, VR-Link™, the world's most widely deployed simulation networking software, is used extensively by most major U.S. military simulation programs, the Department of Transportation, as well as by video game and Web product companies.