Practical Solution of Transitioning a Large Scale Federation from HLA 1.3NG to IEEE 1516

Engin Z. Altan
Science Applications International Corporation
Heathrow, FL
engin.z.altan@saic.com

Don Gordon
Blue Sky Computer Systems, Inc.
Winter Park, FL
blueskycomputer@bellsouth.net

ABSTRACT

The Battle Lab Collaborative Simulation Environment (BLCSE) federation is the Army Training and Doctrine Command's (TRADOC) biggest federation to serve the Army's analytical community. BLCSE has a large, complex, federation-of-federations architecture consisting of 29 different constructive and virtual simulations at 14 geographically distributed sites. The current BLCSE technology environment is comprised primarily of the Distributed Interactive Simulation (DIS) as the primary inter-federate communications protocol. DIS interoperability standards were developed in the late 1980s to support the linkage of simulations exchanging low entity-count data, principally entity-state messages between virtual training devices (e.g., SimNet devices). Active entity counts within BLCSE federations have been steadily increasing as federations grow to support more comprehensive analyses. BLCSE has reached a point where DIS protocol communications cannot reliably manage the federation message load without an externally managed message distribution management scheme. The effects of DIS message saturation, either on the network or at the application itself, are lost messages or incorrectly sequenced messages. Both problems lead to entity state anomalies and lowered data reliability. In view of these challenges, Army Capabilities Integration Center's (ARCIC) Simulations Division Director approved a Simulations Division initiative, in May 2005, to transition the BLCSE federation from DIS (IEEE 1278) to Higher Level Architecture (HLA -IEEE 1516) interoperability standards. However, TRADOC plays an important role in the Army's Cross Command Collaboration Effort (3CE) organization. The 3CE organization currently adopted the Department of Defense (DoD) HLA NG 1.3 standard. In order to provide interoperability with 3CE federation, BLCSE had to implement the NG 1.3 protocol as an intermediate solution. After a year and a half of effort, 20 BLCSE federates are able to communicate in the HLA 1.3 environment. To complete the projects' goal, the development group worked on BLCSE's transition from HLA 1.3 to IEEE 1516. This paper describes the challenges, issues, and problems uncovered during BLCSE's ongoing transition from HLA 1.3 to IEEE 1516 and the lessons learned for transitioning future large-scale, rapid-growth federations.

ABOUT THE AUTHORS

Engin Z. Altan is BLCSE Experimentation Support Team (BEST), Integration Section Manager and leads TRADOC's BLCSE federation transition to IEEE 1516 program. She also has an active role in the 3CE LVC Interoperability task for Object Model development. She holds a Master of Science degree in Electrical Engineering with an emphasis in Meteorology from Technical University of Istanbul, Turkey.

Donald S. Gordon is president of Blue Sky Computer Systems Inc. with offices in Orlando, FL. He has been involved in the simulation industry over 15 years and provides consulting in systems and software development for simulations. Mr. Gordon has a Bachelor of Science degree in Mechanical Engineering and Master of Science degree in Computer Science from the University of Missouri.

Practical Solution of Transitioning a Large Scale Federation from HLA 1.3NG to IEEE 1516

Engin Z. Altan
Science Applications International Corporation
Heathrow, FL
engin.z.altan@saic.com

Don Gordon
Blue Sky Computer Systems, Inc.
Winter Park, FL
blueskycomputer@bellsouth.net

INTRODUCTION

The following sections identify a unique approach to the transition of Battle Lab Collaborative Simulation Environment (BLCSE) federates from the HLA 1.3 simulation protocol to the IEEE 1516 simulation protocol. Innovative use of middleware software that hides the HLA implementation specifics from HLA federate application software allows a seamless migration between the protocols. This paper will address implementation and identify several practical experiences, problems, and lessons learned during the transition that will assist future migration of application software.

BACKGROUND

The BLCSE federation is maintained by Army's Training and Doctrine Command (TRADOC) Future Center to conduct analytical experiments using capabilities from each of the future force's 14 functional areas, shown in Table 1.

Currently the federation uses the DIS protocol for data exchange and communication between federates. However, while BLCSE expands and evolves rapidly, the DIS protocol UDP broadcast communication made repeatable and reliable experiments difficult to execute. Conversely, the basic HLA architecture provides the capability to determine the following:

- What federate information is produced
- What federation information is received
- How to transport the federate information, e.g., reliable, best effort
- How to provide timing between federates, e.g. synchronous or asynchronous

Why IEEE 1516?

When the technical team decided to replace the DIS protocol with High Level Architecture, it was important to take advantage of improved, more robust, commercially standardized set of HLA specifications. It is evident that an open, structured process defined by the IEEE ensures that all user requirements for HLA are being met in IEEE 1516 standards. Furthermore, the rapidly growing BLCSE federation needs processes and procedures for the long-term maintenance of HLA specifications, emerging technology trends, and simulation community user requirements.

The IEEE 1516 specifications provides consistent and precise wording of terminology usage, definitions of terms and acronyms, and accuracy of descriptive text to insure proper interpretation of the rules and accompanying rationale. The specification includes tighter, more robust and consistent set of definitions and acronyms. In addition to the advantages stated above, the decision to select the IEEE 1516 protocol was made primarily for the following two reasons identified below.

- **Technical**

Some services, like Data Distribution Management (DDM), have better flexibility in 1516 than 1.3. In a large-scale federation like BLCSE, bandwidth reduction techniques are extremely important. A BLCSE experiment is very challenging on the communication architecture, which may include:

- 500 instances of the 30 federates shown in Table 2
- 60,000 active entities producing roughly three object updates per second
- 60,000 additional inactive entities across 14 sites nationwide

- **Strategic**

Development and support life cycle for BCLSE federates are quite long. The HLA 1516 standard offers new and the improved features not readily available in older simulation standards. Often, many new simulation projects opt for the IEEE 1516 standard since it is likely to provide support well into the future.

Table 1 - Translation Techniques

	Gateway	Wrapper	Native	Middle ware
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**

Table 2 - BLCSE Federates

Federate Name	Model Manager
Advanced Concept Research Tool-Air Variant (ACRT-AV)	AMBL
Advanced Concept Research Tool-Ground Variant (ACRT-GV)	MMBL
Aggregate-Level Communications Environment Server (ALCES)	Communications-Electronics Research, Development and Engineering Center (CERDEC)
Advanced Simulation Technology Inc. (ASTi)	MMBL
Advanced Tactical Combat Model (ATCOM)	AMBL
Advanced Warfighting Simulation (AWARS)	BCBL-L
Area of Interest (AOI) Network Server	MMBL
Chemical Biological Radiological and Nuclear (CBRN) Defense	MSBL
Collection Manager's Console / Intelligence Request Logger (CMC/IRL)	SMDBL
Comprehensive Mine and Sensor Simulation (CMS2)	CERDEC
Counter Mine Server	MMBL
Data Analysis	MMBL
Data Logger	MMBL
Effect Server	MMBL
Exercise Manager	MMBL
Extended Air Defense Simulation (EADSIM)	SMDBL
Fire Support Simulation (FIRESIM)	DSABL
Joint Approval Console (JAC)	SMDBL
Mobile Command and Control (MC2)	CERDEC
Network Planning Simulation Tool (NPST)	BCBL-G
OneSAF Testbed (OTB)	MMBL
Radio Server	MMBL
Reporter	MMBL
SA Server	MMBL
SIGINT Model Suite	CERDEC
Simulation of Location and Attack of Mobile Enemy Missiles (SLAMEM)	SMDBL
Space Server	SMDBL
SVS™	SBL
System Interface Unit (SIU) Interface	MMBL
Universal Controller (UC)	CERDEC

Transition Approaches

The decision team considered four different techniques , discussed in the following sections, to implement the IEEE 1516 protocol transition.

Gateway Approach

This approach employs an application executing on a dedicated computer external to the simulator, show in Figure 1. The application translates simulation data between DIS and HLA protocols. The implementation of a gateway does not require any modification to the simulator application software. The gateway does add considerable latency to the availability of simulation data as it is translated. However, the single gateway computer is a single point of failure and limits the scalability of the federation to the performance characteristics of the gateway hardware. Finally, many of the advanced features provided by the HLA architecture are not provided by the gateway solution.

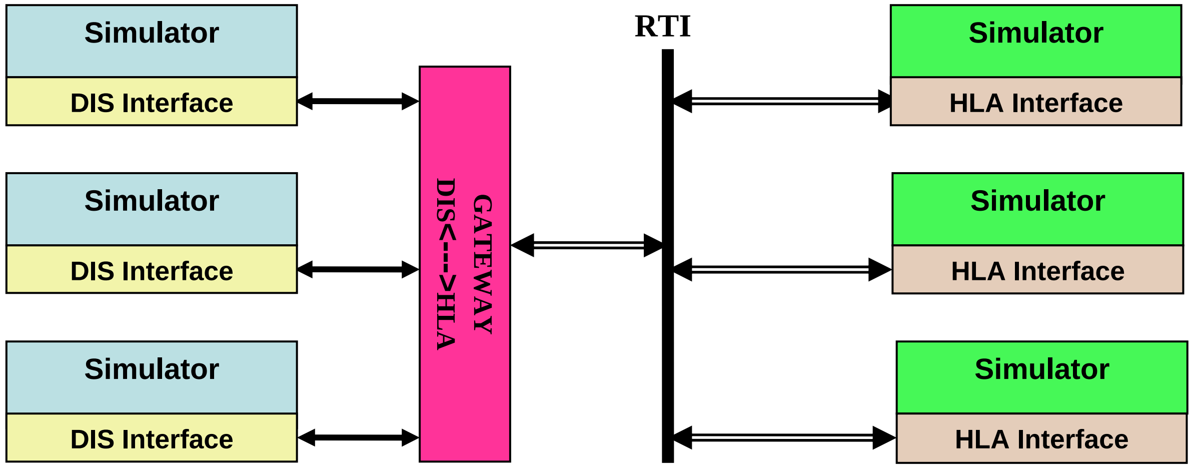


Figure 1- Gateway Approach

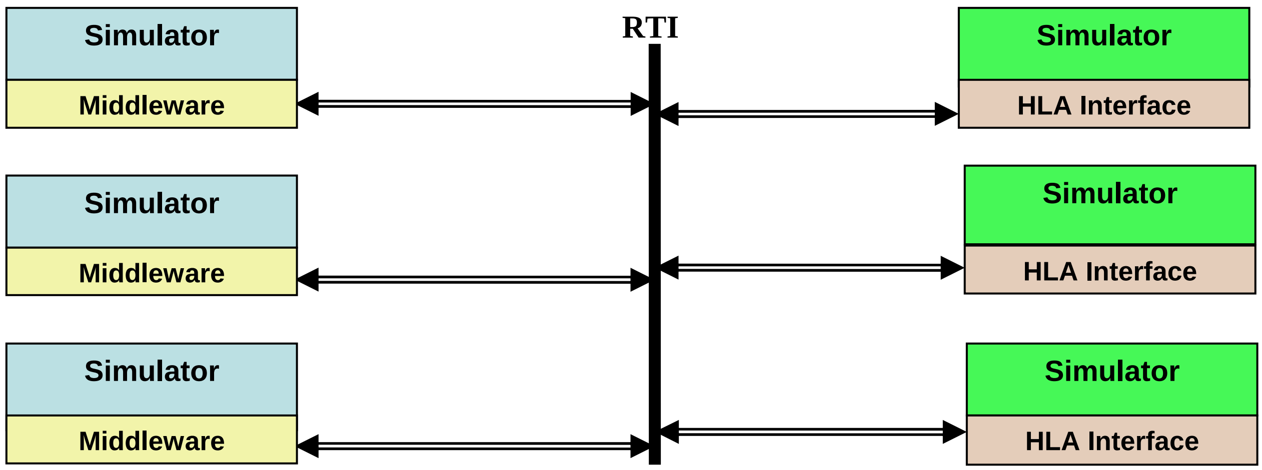


Figure 2 - Middleware Approach

Native RTI API Approach

This approach directly connects the simulator application software to the API provided by the RTI. The approach allows the federate to take advantage of the advanced features of the HLA protocol. However, it requires a substantial modification of the simulator application software and is sensitive to any future changes to the HLA protocol API. Consequently, since the simulator application software must be individually changed, it is a more expensive and time consuming approach. In some circumstances, the legacy application software might not provide a complete set of source code and libraries necessary to rebuild the application preventing this approach altogether.

Wrapper Approach

This approach adds wrapper software underneath the simulation's DIS interface to translate the data from DIS to HLA just before it is sent, and to translate from HLA to DIS just after it is received. The wrapper software does not require any external hardware to run and requires very limited changes to simulator software. However, forward and backward compatibility requires additional application software changes. Finally, the wrapper approach does not take advantage of advanced HLA protocol features.

Middleware Approach

This approach interfaces with the simulation software to the HLA federation using a translation library between the application software and the HLA client software, shown in Figure 2. This translation layer will henceforth be referred to as the Middleware (MW). The Middleware provides both a direct interface to the Federation simulation data and an indirect interface using a DIS like API interface through a DIS Adapter software layer.

The BLCSE HLA transition project selected the MW approach.

IMPLEMENTATION

After the decision was made to use the MW approach, the next step was selecting the most suitable RTI. BLCSEs geographically distributed federates and the demand of high entity count for experiments forced the team to search for an RTI with the following:

- Minimum CPU, bandwidth and memory requirements
- Increased client-side processing at remote sites
- Multi-threaded architecture which could use segmentation
- Additional processing power to support growing federations.

The other important factor was the customer's limited budget for the HLA transition which included:

- The price of RTI's licenses per federate
- Testing, debugging, and data collection tools to support project and experiments
- Service and upgrade charges

Another factor was using the selected RTI in classified environments. Since all the BLCSEs battle laboratories are classified, the customer was very sensitive about using any commercial products in the laboratories. Considering both technical and business factors, the team selected a commercial RTI suitable for the project. When the team was ready to sign the agreement with a commercial RTI vendor, yet another challenge appeared. TRADOCs participation in the newly developed Cross Command Collaboration Effort (3CE) organization established the expectation that the BLCSE federation must be interoperable with other 3CE federations which include:

- The Army's Research, Development and Engineering Command's (RDECOM)
- Future Combat Systems (FCS) Lead System Integrator's (LSI) Modeling Architecture for Research, Technology, and Experimentation (MATREX) federation
- Army Test and Evaluation Command's (ATEC) and Distributed Training Environment (DTE) federation which utilize Test and Training Enabling Architecture (TENA) protocol

Contrary to the direction BLCSE was heading, the 3CE federation would implement the HLA 1.3 standard using the MATREX RTI based on Defense Modeling and Simulation Organization (DMSO) RTI 1.3NGv4.2.

The Transition Strategy

In order to provide interoperability with 3CE federation and meet the BLCSE HLA Transition projects aggressive schedule, the best solution was to implement NG 1.3 protocol first and then transition to IEEE 1516. However, the two-step transition had to incorporate a one-step development. Therefore, the team selected the Middleware Approach for the transition from the DIS protocol to the HLA 1.3 protocol and eventually the IEEE 1516 protocol.

During this phase, the team developed a multi-layer middleware library which provides a seamless migration between the protocols. The Middleware architecture is comprised of several layers, illustrated in Figure 3. The bottom-most layer is the Middleware protocol layer. This layer provides an abstraction of the HLA RTI version and vendor specific protocol. The layer includes an HLA API that provides data types that are not vendor specific. No inclusion of the vendor-specific include files are allowed at this layer.

The Middleware protocol layer provides an infrastructure that allows plug-in libraries to implement the abstract layer using the vendor specific RTI services. During runtime, the Middleware selects a plug-in library by name. This allows the federate developer to select the RTI vendor software at runtime without the recompilation of federate software. This is especially important since all federates in a federation must use the same RTI vendor at runtime. The plug-in protocol libraries implement each abstract service using the specific data types and services provided by the RTI vendor.

Running on top of the Middleware protocol layer is the Middleware layer. This layer provides abstraction of the objects and interactions in the Federation Object Model (FOM). The initial implementation of the

Middleware was MATREX FOM centric and many of the MATREX FOM objects and interactions are modeled in the Middleware software. During later versions of the Middleware, objects and interactions from the BLCSE FOM were added as well. This layer allows the federate developer to create objects and interactions without knowledge of the underlying HLA architecture details. This layer also handles issues such as byte swapping, data marshalling and dismarshalling, attribute request/provide responses, and federate ambassador callbacks, to name a few.

The final layer in the Middleware software is the DIS Adapter. Since nearly all legacy BLCSE federates are DIS-based, a means to seamlessly integrate existing federates into the BLCSE federation is required. The DIS Adapter provides an API that is very similar to the socket interface that federates use to transmit and receive UDP network packets. The DIS Adapter translates the DIS packets into HLA transactions and makes the proper HLA service calls. Likewise, upon receipt of an HLA simulation event, the DIS Adapter translates the HLA data into the proper DIS PDU and notifies the federate.

The flexibility of this Middleware architecture was instrumental in transitioning the BLCSE federates from HLA 1.3 to IEEE 1516. No federate code was required to change for this transition. At runtime, a federate specifies the name of the new simulation protocol library (IEEE 1516) and the benefits of the new protocol were provided by the underlying Middleware protocol library.

RTI Vendor Selection

The selection of an RTI vendor for IEEE 1516 is important to the overall success of the transition. The IEEE 1516 C++ API provides binary compatibility between different vendor implementations. This allows a single code base to work with all IEEE 1516 vendors. However, different vendors RTIs are not interoperable within the execution of a single federation. Therefore, a single RTI vendor must be used throughout the entire federation.

Issues such as license schemes and cost must be addressed to handle all federate members. Import/export issues of foreign developed software must also be addressed. Finally, vendor supplied tools and access to vendor support facilities such as email and phone support are important. MAK's support of the old federation file format was important since the support of a single federation file for both HLA 1.3 and IEEE 1516 was desired.

Middleware Protocol Development

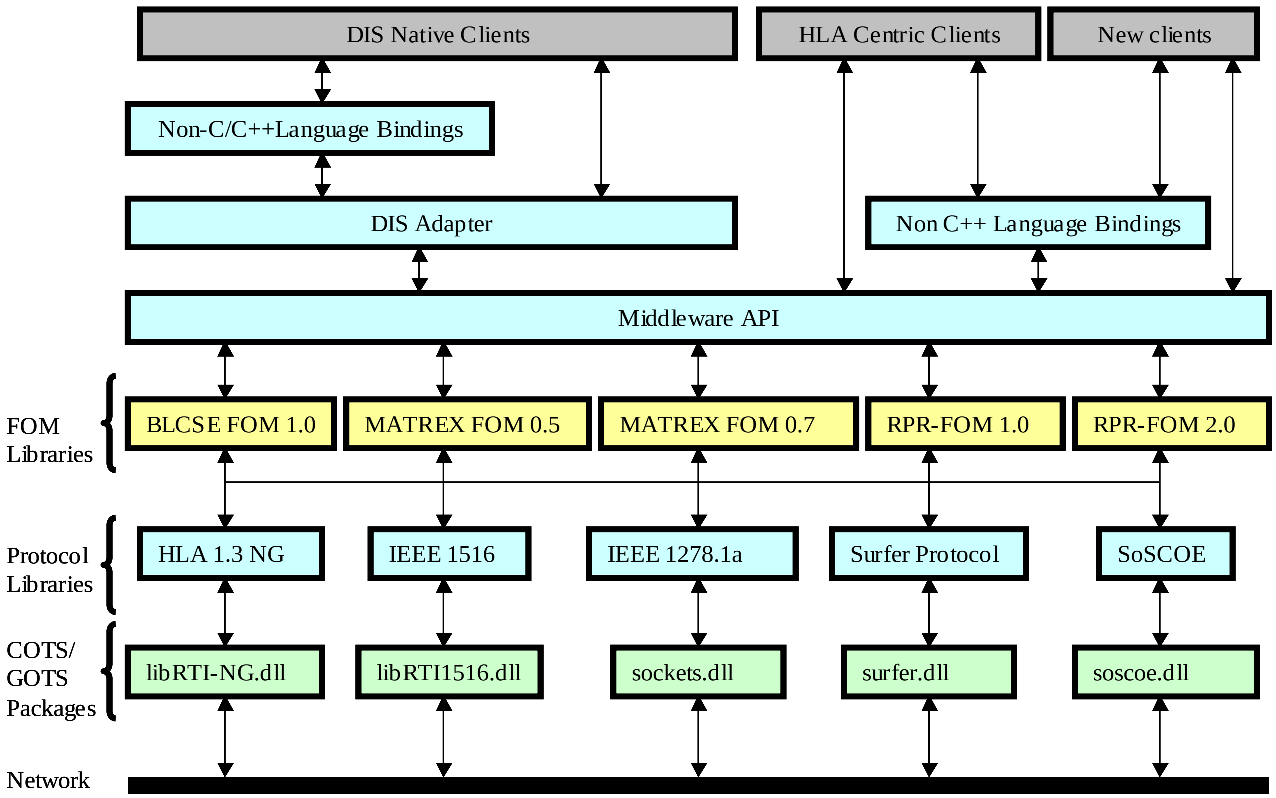
The transition of the existing BLCSE Middleware architecture from the HLA 1.3 protocol to the IEEE 1516 protocol consisted of the development of a new Middleware protocol plug-in library. Initially, a shared common source code base was attempted between the two protocols with minor differences handled with C++ conditional compilation macros. However, the sweeping changes in data types used between the APIs necessitated two separate and distinct code bases between the protocols. The federate developer selects, at runtime, the HLA protocol plug-in that is used by the Middleware software. Several key issues arose during the development of the IEEE 1516 protocol plug-in from the HLA 1.3 protocol plug-in and they are discussed in the following subsections.

C++ API Changes

The IEEE 1516 API is very similar to the HLA 1.3 API. However, some important differences impacted the time required to develop the IEEE 1516 protocol plug-in. The APIs both offer, for the most part, the same HLA services. The method names and argument list used to implement these servers are identical except for a few instances. The C++ name space for the RTI changed as well but these changes were easily integrated into the new protocol plug-in.

The major change is the data types used by the services. The HLA 1.3 standard used custom data types for sets and maps which are used extensively to pass lists of handles, attributes, and parameter data. The IEEE 1516 standard uses the C++ Standard Template Library (STL) for these data types.

Figure 3 – Middleware Architecture Diagram



Any references to these data types had to change for the IEEE 1516 protocol plug-in. Conversion to the new STL data types consumed the majority of the development effort.

The major advantage of the new IEEE 1516 API is the binary compatibility with all vendor RTI implementations. The IEEE 1516 protocol plug-in is developed using the MAK RTI. However, this protocol plug-in will work with all IEEE 1516 vendors without any code changes.

RTI Handle Changes

The change in the representation of handles used throughout the RTI API to identify object instances, attributes, and parameters was the most difficult technical problem during the transition. The goal of the Middleware, and especially the protocol layer of the Middleware, is to isolate protocol-specific data types and nomenclature from the rest of the architecture. Handles in the HLA 1.3 API are numeric values (unsigned long) and some existing knowledge of these handle types was used in the Middleware protocol. Handles were used as keys in maps to identify object instance mapping to data values. With the change of these handles from numeric values to a C++ class, these mappings became problematic.

The handle problem was solved by creating an abstract C++ class that represented the handle data type. Each protocol implemented the abstract class as a concrete class that provided conversion between the native API data type and the Middleware data type. This solution creates overhead during data type conversion. This conversion overhead was unavoidable if the Middleware API was to remain simulation protocol independent. In hindsight, the original Middleware protocol implementation with HLA 1.3 should have not identified RTI handles as a numeric type.

DDM Changes

The IEEE 1516 DDM API represented the biggest difference between the RTI APIs. Some service names where changed or removed and new service names added. However, the majority of the DDM service architecture remains remarkably similar to the HLA 1.3 API. The biggest change is the absence of DDM routing spaces in the IEEE 1516 DDM. The transition to the new DDM API in some respects made the DDM implementation easier. Since the MAK RTI offers backward compatibility with the HLA 1.3 federation file definition, no rework of the FOM was necessary to eliminate the DDM routing spaces.

Vendor Specific RID File Changes

The RID file identifies the vendor specific configuration parameters for the RTI. While many of the RID file parameters are the same or similar between the protocols, there are vendor specific parameters that must be properly set to enable specific RTI features and optimize performance for the runtime environment. Trial and error was often the best way to evaluate the effects of different RID file settings. Vendor support for the RTI is also critical while configuring the RID file.

LESSONS LEARNED

With today's large-scale composite federations (like BLCSE), which are open to continuously increasing demand for experimentation, it is risky to use DOD developed HLA 1.3. Even using the most enhanced version of it, NG 1.3-based RTI may still need continuous maintenance and upgrade. The large military experiments are very expansive, involving hundreds of computers and extensive labor. Therefore, even a small problem in the RTI may create a big problem in the federations' data transfer or communication. Even one or two RTI crashes during an 8 hour daily run results in wasted time and increases the cost of the experiment. For example, while testing the MW in the BLCSE federation with 10 different federates using NG 1.3-based RTI, we discovered that at least one third of the problems were RTI related. Thus, the RTI developers' timely support is very important. Vendor developed support tools such as test and data collection tools play an important role in the development phase and during the experiments. Otherwise, integration of different vendor developed tools in the environment is costly. We suggest, if you are transitioning from 1.3 to 1516, to use the same commercial vendor, if possible. If you are transitioning from government owned RTI 1.3 to a commercial 1516 RTI vendor, evaluating the support tools is as important as evaluating the RTI itself. Again, if possible, stay with the same vendor.

SUMMARY

A large–scale federation transition from HLA NG1.3 to IEEE 1516 requires planning, flexible design, rapid implementation and seamless migration from one protocol to another. The selection of a transition approach is extremely important. It has to be very flexible, while incurring the least amount of risk and cost. The transition may take more time than anticipated. Therefore, the federate should be able to maintain its previous protocol compliance vital to ongoing projects, while providing a planned upgrade to transition. The program managers should recognize that there are reasonable tradeoffs of risk versus benefits. However, the cost associated with migration is extremely minimal compared to potential reduction in program technical, schedule, and cost risks. The long-term benefits provide a solution that is more robust, complete, and widely accepted commercial standard.

REFERENCES

Hanover, Paul, 2004, *Progress Report on the Battle Lab Collaborative Simulation Environment* Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Paper 1563

Defense Modeling and Simulation Office, 2001, *Transition of the DOD High level Architecture to IEEE Standard 1516*

Jones, E Jennifer, 2005, *Transitioning Large-Scale Federations from IEEE 1278 to IEEE 1516* Simulation Interoperability Standards Organization (SISO) Paper 05F-SIW-030

Deb, Fullford, 1998, *Transitioning your DIS Simulator to HLA*, MAK Technologies, White paper

Björn Möller, Lennart Olsson, 2004**,** *Practical Experiences from HLA 1.3 to HLA IEEE 1516 Interoperability.* Simulation Interoperability Standards Organization (SISO) Paper 04F-SIW-045