Providing Uninterrupted Training to the Joint Training Confederation (JTC) Audience During Transition to the High Level Architecture (HLA)[1]

Sean P. Griffin: The MITRE Corporation
Ernest H. Page: The MITRE Corporation
C. Zachary Furness: The MITRE Corporation
Mary C. Fischer: U.S. Army STRICOM

Summary The Aggregate Level Simulation Protocol (ALSP) Joint Training Confederation (JTC) -- currently consisting of twelve primary constructive training simulations -- has been used to support several major command post exercises each year since 1992. In a recent memorandum signed by the Undersecretary of Defense for Acquisition and Technology, ALSP and its cousin, the Distributed Interactive Simulation (DIS) protocol are targeted for replacement within the U.S. Department of Defense (DoD) by the emerging High Level Architecture (HLA) for models and simulations. The JTC itself is also scheduled for replacement by the Joint Simulation System (JSIMS). The transition from ALSP to HLA and from the JTC to JSIMS must be carefully planned to provide a "seamless" migration and uninterrupted training support. One of the aspects of this migration path has been the construction of the ALSP Data and Protocol Transfer Over RTI (ADAPTOR). Use of the ADAPTOR requires no change to the existing ALSP-compliant simulations, and permits the JTC to utilize the HLA Runtime Infrastructure.

1. INTRODUCTION

The Aggregate Level Simulation Protocol (ALSP) resulted from a U.S. Defense Advanced Research Projects Agency (DARPA) effort to identify mechanisms suitable to facilitate the integration and interoperation of existing (so-called "legacy") constructive training simulations. The Joint Training Confederation (JTC) – the primary application of ALSP – has evolved from two models in 1992 to a planned twelve for 1997 and supports several large-scale command post exercises (CPXs) each year, including the annual Ulchi Focus Lens, Prairie Warrior and Unified Endeavor exercises.

Both ALSP and the JTC are nearing their respective ends of service. The U.S. Defense Modeling and Simulation Office (DMSO) has sponsored the definition and development of the High Level Architecture (HLA) for models and simulations (M&S). The HLA has been defined to promote interoperability across a wide spectrum of M&S, encompassing all those currently supported by either ALSP or the Distributed Interactive Simulation (DIS) protocol. In a recent memorandum signed by the U.S. Undersecretary of Defense for Acquisition and Technology Dr. Paul Kaminski, the HLA has been endorsed as the standard for all U.S. Department of Defense (DoD) M&S. The HLA standard supersedes both ALSP and DIS and all DoD M&S must comply with the HLA, receive a waiver, or be retired by 2001.

The Joint Simulation System (JSIMS) is currently under development and will serve as the next-generation platform for joint training. JSIMS is scheduled to replace the JTC in 1999 -- with the 1999 JTC capabilities defining the requirements for JSIMS initial operational capability (IOC). The transition from ALSP and the JTC to HLA and JSIMS must be carefully planned to ensure uninterrupted service to the training audience. A variety of alternatives exist – from a "lights-off/lights-on" transition to myriad phased approaches. The challenges to this transition are both technical and operational in nature.

This paper highlights the transition from ALSP/JTC to HLA/JSIMS. In Section 2 the broad aspects of the transition are briefly discussed and the remaining JTC development calendar is summarized. A software utility constructed to facilitate a wide variety of phased transition options is described in Section 3. A few conclusions are given in Section 4.

[1] To Appear in *Proceedings of the 1997 Simulation Technology and Training Conference,* Canberra, Australia, 17-20 March.

2. JTC TRANSITION

The planned 1997 JTC, outlined in Table 2.1, consists of twelve simulations that encompass the major functionality required for Joint training exercises.

The ALSP Infrastructure Software (AIS) provides the mechanism for inter-simulation communication. Information regarding objects, interactions, time flow and so forth is accomplished through the AIS. For a complete description of the AIS refer to [1,2,3]. For purposes of this discussion, two primary components of the AIS merit introduction. The ALSP Common Module (ACM) provides the simulation interface and coordinates the principle activities within an ALSP confederation, including: joining and departing the confederation, time advance, message delivery, and ownership management. The ALSP Broadcast Emulator (ABE) provides multi-cast message distribution within a confederation.

JTC Simulation	Primary Functionality
Corps Battle Simulation (CBS)	Representation of Army combat operations
Air Warfare Simulation (AWSIM)	Representation of Air Force combat operations
Research, Engineering, and System Analysis (RESA)	Representation of Naval combat operations
Marine Air Ground Task Force (MAGTF) Tactical Warfare Simulation (MTWS)	Representation of Marine Corps combat operations
Combat Service Support Training Simulation System (CSSTSS)	Representation of Army logistics
Joint Electronic Combat - Electronic Warfare Simulation (JECEWSI)	Representation of Electronic Warfare
Tactical Simulation Model (TACSIM)	Representation of Intelligence assets and information feeds
Portable Space Model (PSM)	Representation of Satellites, Theater Missile Defense
Air Force Semi-Automated Forces (AFSAF)	High fidelity representation of air combat
Analysis of Mobility Platform (AMP)	Representation of military transportation
Logistics Anchor Desk (LAD)	Provides linkage to Logistics Decision aid
Joint Operational Visualization Environment (JOVE)	Visualization of simulation battle, as portrayed across ALSP

Table 2.1 Planned 1997 JTC Simulations

2.1 JTC Evolution through FY01

The evolution of the JTC through 1999 is guided by the ALSP Master Plan. The ALSP Master Plan is updated annually and documents CINC joint training requirements as well as the means for implementing them within the JTC. The current ALSP Master Plan is based on CINC requirements collected and prioritized by the Joint Warfighting Center (JWFC) during FY96, and forms the basis of concept development for the 1997-99 JTCs. It is envisioned that this list of requirements will change little between now and 1999 and that enhancements to the current JTC will be based on only the highest priority requirements.

In May 1996 the ALSP Review Panel approved continued development of the JTC through 1998, with an option for additional functionality in 1999 if deemed necessary. After 1999, JTC functionality will be frozen and the simulations that comprise the JTC will go into a "maintenance" mode; they will be maintained, without enhancement, until replacement by JSIMS or its components.

Current plans call for a transfer of ALSP Executive Agency from the Simulation, Training and Instrumentation Command (STRICOM) to the Joint Program Office (JPO), which leads JSIMS development, in the spring of 1998. This transition coincides with the completion and fielding of the 1998 JTC. Transition of the JTC management structure should help promote a seamless migration between system functionality -- aligning aspects of the current JTC development cycle with JSIMS development. Coincidental with the change in Executive Agency, the ALSP Interface Working Group (IWG) (see [1]) will be subsumed by the Joint Working Group (JWG) with the intention of unifying the simulation user and development communities for both the JTC and JSIMS.

2.2 Transition Challenges

The most pressing challenge in the transition between the JTC and JSIMS is to replace the current JTC simulations, and the underlying AIS without a loss of training capability to the joint training audience. An effective method to accomplish this "seamless" transition would seem to be a *phased* approach. Specifically, a mechanism is needed to permit ALSP-compliant simulations and HLA-compliant simulations to exist and interoperate within the same federation. A few advantages of this approach are:

- *Addition of HLA-compliant federates*. Over the next several years, a variety of HLA-compliant simulation interfaces are anticipated candidates for JTC membership. One such example is the Modular Reconfigurable C4I Interface (MRCI) project, which is planned for incorporation into the JTC in FY98. MRCI will provide a common, reusable interface for a variety of C4I systems that allows them to exchange information with simulations using the HLA Run-time Infrastructure (RTI). By transitioning the JTC from an ALSP architecture to the HLA, such HLA-compliant simulation interfaces can be accommodated.
- *Cost-effective migration from ALSP compliance to HLA compliance*. A few of the current JTC simulations may become HLA compliant over the next several years. While these simulations could continue

to maintain their ALSP interfaces, a mechanism that would allow them to interoperate with the JTC through their HLA interface would reduce the maintenance costs involved with that simulation.

- *Cost-effective expenditures in both the near- and long-terms*. Many of the existing simulations within the JTC will be retired within the next five years. Enhancements to these simulations will likely be limited to those that are necessary to meet immediate and critical training requirements. Over the next five years, gradual reductions in funding for support of the JTC system engineering activities are anticipated. However, maintenance costs for traditional system engineering activities such as continued documentation of the AIS, ALSP training, and ALSP exercise support will tend to remain constant over that time frame. Transitioning from the AIS to the RTI would remove the burden of AIS maintenance (and associated training, and exercise support costs) from the JTC systems engineer and share that cost across the community by using the same, community supported, DoD simulation infrastructure.[2] Transitioning to the HLA RTI will enable products developed under system engineering support to have utility beyond the JTC. Such products include: automated testing software, database management applications, and diagnostic and management tools used to coordinate simulation operations during a test or exercise.
- *Use of HLA support software*. In addition to systems engineer-developed software, utilizing the HLA enables the JTC to take advantage of the broad set of support tools envisioned for HLA federations. These tools will be developed, tested and maintained outside of the ALSP funding structure.

A software component to facilitate ALSP-HLA interoperability is described in the next section.

3. ALSP DATA AND PROTOCOL TRANSFER OVER RTI

To enable a phased transition to the HLA some mechanism is needed to allow ALSP-compliant simulations and HLA-compliant simulations to exist and interoperate within the same federation. The ALSP Data and Protocol Transfer Over RTI (ADAPTOR) was constructed to accommodate this interoperability. The ADAPTOR coupled with the RTI replaces the ALSP Infrastructure Software (AIS) as the underlying mechanism for data exchange and time management among interconnected simulations. The ADAPTOR receives ALSP protocol messages from JTC simulations and translates these messages into RTI service calls. Similarly, the ADAPTOR receives information from the RTI, constructs ALSP protocol messages (as appropriate) and forwards them to the JTC simulations. Beyond facilitating ALSP-HLA interoperability, the primary requirement for the ADAPTOR is transparency. A JTC simulation that currently has an ALSP interface will not need to build an RTI interface to utilize the RTI.

3.1 ADAPTOR Design Alternatives

Figure 3.1 depicts early design alternatives considered for the ADAPTOR. Figure 3.1(a) depicts the ADAPTOR serving as a bridge between two distinct federations, an ALSP confederation and a HLA federation. These federations exchange data and time management information via the ADAPTOR. In this configuration, ALSP-compliant simulations continue to rely on the AIS for data management and time management services. Therefore, the AIS would have to be enhanced/maintained in conjunction with the ADAPTOR. This configuration would not use the ADAPTOR if there were no HLA-compliant simulations participating with the JTC. Here the ADAPTOR is clearly a bottleneck as all traffic between the two confederations passes through it.

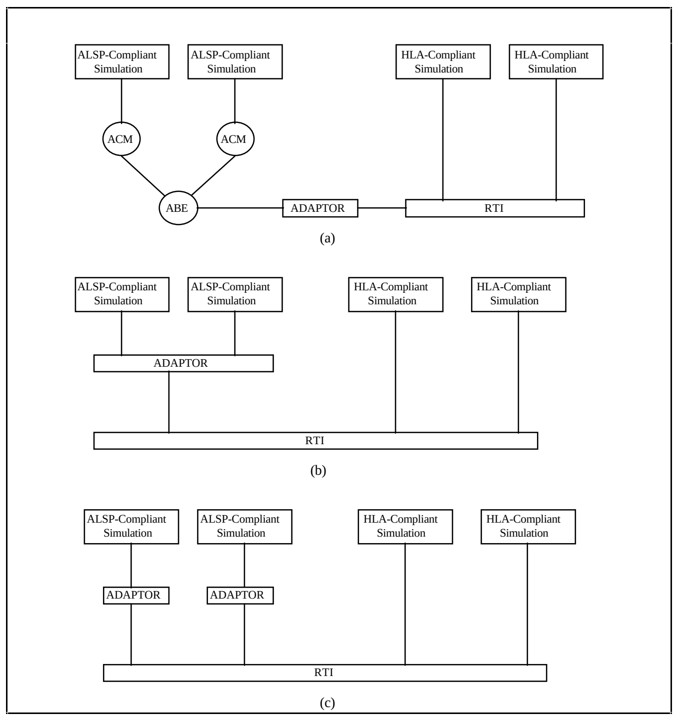

Figure 3.1: ADAPTOR Design Alternatives

Figure 3.1(b) depicts a single instance of the ADAPTOR supporting all ALSP-compliant simulations. HLA-compliant simulations communicate directly with the RTI. The ALSP-compliant simulations utilize the RTI services regardless of the presence of HLA-compliant simulations. However, the ADAPTOR would have to implement some of

[2] Although a potential cost savings in systems engineering support is evident, this savings must be weighed against additional costs associated with fielding the RTI to the various exercise sites.

its own data distribution services to route incoming data from the RTI to the correct simulation. This configuration also exhibits potential inefficiencies due to the amount of network traffic that must be handled by he ADAPTOR.

Figure 3.1(c) represents N instances of the ADAPTOR, one for each ALSP-compliant simulation. HLA-compliant simulations communicate directly with the RTI. This alternative was selected for implementation and exhibits several positive characteristics:

- It fully utilizes the data distribution and time management services provided by the RTI.
- The two main objectives (interoperability and transparency) of the ADAPTOR are met by this approach.
- It "looks" the same from an exercise configuration standpoint as the AIS and admits the widest range of network configurations. A simulation's ADAPTOR can be strategically located to maximize efficiency for a given network topology.

The Implementation language was another design alternative for the ADAPTOR. Two primary candidates were considered: C++ and Java. C++ is a natural candidate since it is the implementation language of both the F.0 and 1.0 versions of the RTI. The application developer must instantiate two C++ objects: (1) the *RTI Ambassador*, which is used to pass data to the RTI; (2) the *Federate Ambassador*, which provides methods for the RTI to invoke. The disadvantages of C++ are primarily its limited portability, and lack of support for both exception handling and garbage collection.

Java, by design, overcomes many of the limitations of C++. Java provides a large degree of platform independence -- compiling Java source code produces “bytecodes” which are interpreted by a Java Virtual Machine (JVM). Bytecodes produced on one platform can run, without modification, on any platform for which a JVM has been implemented. Java as a programming language is fairly robust. Java implements/enforces a variety of features that protect programmers from hard-to-track-down bugs. Java is strongly typed and does not allow pointer arithmetic. In addition, Java performs garbage collection, fully implements exception handling, and has built in thread support.

The primary risk of using Java is its performance. As indicated above, Java is an interpreted (rather than a compiled) language. A layer of software – the JVM – exists between a Java application and the operating system. This interpretation will cause a Java application to run slower than an application developed in a compiled language like C or C++. While performance of the ADAPTOR is an important issue, the ADAPTOR does not have "hard" real-time requirements. The expected introduction of Java Just-In-Time compilers – which compile Java bytecodes to native machine code during runtime -- portend a solution to the Java performance question. With Just-In-Time compilation, the performance of a Java application should be comparable to that of a C or C++ application.

The advantages of Java in the ADAPTOR context, several of which – such as its suitability as a network programming language – are not mentioned above, seem to outweigh its shortcomings and the decision was made to use Java for the ADAPTOR implementation. An additional mitigating factor in this selection is the anticipation of a Java RTI (around mid-summer 1997).

3.2 ADAPTOR Design

The ADAPTOR is composed of three primary classes, the *ALSPProcessor* class, the *JavaRTIAmbassador* class, and the *JavaFederateAmbassador* class, and many smaller supporting classes.

The *ALSPProcessor* class reads ALSP protocol messages from an ALSP-compliant simulation, and performs any processing that is not provided by the RTI but is present in the AIS. The information contained in the message is then translated into an appropriate RTI service call, which is made via the *JavaRTIAmbassador* class.

The *JavaRTIAmbassador* class provides the interface from the Java source code to a C++ *RTI Ambassador* object. Every application that utilizes the RTI must instantiate an *RTI Ambassador* object. Information received from the JTC simulation is passed to the RTI via this class.

The *JavaFederateAmbassador* provides a Java interface to the required C++ *Federate Ambassador* object. It receives service calls from the RTI, performs the required processing, translates the service call into an appropriate ALSP protocol message(s), and sends the message(s) to the simulation. Additionally, several ancillary classes are defined within the ADAPTOR to support data storage, communication, and file management.

The two primary Java classes, the *JavaRTIAmbassador* and *JavaFederateAmbassador* are designed to provide a generic Java interface to an RTI, i.e. any Java-based federate could utilize these object classes. For example, the Federation Management Tool (FMT) is currently under development to manage the operation of a federation. At least one prototype FMT is being constructed in Java and will utilize the object classes built for the ADAPTOR.

3.3 Missing AIS Functionality

Generally speaking, while the RTI provides all of the end-to-end functionality needed by ALSP, several aspects of inter-AIS functionality are not present in the extant RTI. For example, in ALSP a simulation may stipulate a minimally-required subset of attribute values required to create a “ghost” (“reflection” in HLA parlance) of a particular object instance. This set is known as the *create set* for a given class. The HLA Interface Specification provides no such mechanism. Therefore, this functionality is implemented in the ADAPTOR.

The HLA Interface Specification defines a capability for object/attribute filtering, however the syntax for this filtering is vaguely specified, and no filtering is implemented in version F.0 of the RTI. ALSP provides mechanisms to establish attribute filters for any object class of interest to the simulation. These filters limit external object representations to simulation-defined areas of interest, or attribute value ranges. The ADAPTOR implements the attribute filtering present in the AIS and the ALSP Out of Domain Database, which directly supports the AIS filtering algorithm.

Finally, runtime management of the JTC must be provided. Currently, the majority of exercise management data for the JTC exists within the AIS. When the JTC infrastructure is provided by the ADAPTOR/RTI, the FMT will be used to manage the running JTC. The FMT operation is dictated by the Management Object Model (MOM) within the HLA. The ADAPTOR has been constructed to provide the necessary, MOM-consistent "hooks" that allow management via the FMT.

3.4 Implementation Plan

Introduction of the HLA RTI and the ADAPTOR into the JTC will occur over two years. The FY97 efforts are directed to experimentation and proof-of-concept prototyping of the ADAPTOR. During this time, ADAPTOR development and testing will take place. Testing will initially be performed in the lab and subsequently will take place with a subset of existing JTC simulations.

The RTI/ADAPTOR will be fully introduced into the JTC development cycle in FY98. The RTI/ADAPTOR will participate in the 1998 ALSP All-Actor Integration and ALSP Confederation Test. Upon successful execution of these major test events, the RTI/ADAPTOR infrastructure will be available to support training exercises.

Figure 3.1 (part c) depicts how the ADAPTOR is intended to provide interoperability between simulations in the existing JTC, and HLA compliant simulations that are planned to be introduced over the next 2-3 years. The transition from an ALSP architecture to the HLA will be seamless, through the use of the ADAPTOR -- allowing simulations to migrate to HLA if mandated, or continuing to maintain an ALSP interface if not.

The fielding of the 1998 JTC (with the RTI and ADAPTOR) to the various exercise sites, will create the need for additional hardware at these sites. However, these costs may be minimized by the choice of a Java implementation for the ADAPTOR (as well as a potential Java RTI). Due to its platform-independent nature, Java portends flexibility in hardware procurement.

4. CONCLUSIONS

Both ALSP and the JTC are nearing the end of their respective service lives. To borrow a phrase from a recent U.S. election, a "bridge to the future" must be constructed – from ALSP to the High Level Architecture, and from the JTC to JSIMS. The challenges to this transition are both technical and operational. And while no singular "best" transition path can be defined, a need clearly exists to chart a course toward future systems that provides minimal disruption to the training audience, and cost-effectiveness. The construction of a software utility, the ADAPTOR, that permits both ALSP-compliant and HLA-compliant simulations to exist and interoperate within the same federation is a solid foundation upon which to build this bridge.

5. REFERENCES

[1] Fischer, M.C. "Joint Training Confederation," Proceedings of the 1st International Simulation Technology and Training (SimTecT) Conference, Melbourne, Australia, 25-26 March 1996.

[2] Weatherly, R.W., Wilson, A.L. and Griffin, S.P. "ALSP – Theory, Experience, and Future Directions," In: Proceedings of the 1993 Winter Simulation Conference, pp. 1068-1072, Los Angeles, CA, 12-15 December 1993.

[3] Weatherly, R.M., Wilson, A.L., Canova, B.S., Page, E.H., Zabek, A.A. and Fischer, M.C. "Advanced Distributed Simulation Through the Aggregate Level Simulation Protocol," In: Proceedings of the 29th International Conference on System Sciences, Vol. 1, pp. 407-415, Wailea, HI, 3-6 January 1996.

[4] Wilson, A.L. and Weatherly, R.M. "The Aggregate Level Simulation Protocol – An Evolving System," In: Proceedings of the 1994 Winter Simulation Conference, pp. 781-787, Lake Buena Vista, FL, 11-14 December 1994.