Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

APPLICATION OF THE ANALYSIS FEDERATE IN THE JOINT ADVANCED DISTRIBUTED SIMULATION JOINT TEST FORCE ELECTRONIC WARFARE PHASE II TEST

William S. Murphy, Jr.

TRADOC Analysis Center-Monterey
Post Office Box 8692
Monterey, CA 93943, U.S.A.

Michael L. Roane

Joint Advanced Distributed Simulation JTF
11104 Menaul Blvd NE
Albuquerque, NM 87112-2454, U.S.A.

ABSTRACT

The Analysis Federate is a general purpose High Level Architecture (HLA) data collection, analysis, and visualization tool that was developed by the U.S. Army TRADOC Analysis Center in Monterey. The Analysis Federate was designed to be composable across HLA federations that use different object model abstractions in their Federation Object Models (FOM). This composability is provided by a conceptual framework that includes fourth generation development tools that automate the procedures required for a federate to subscribe, publish, and interpret federation data. Many of these automated procedures could not be used when the Analysis Federate was applied in support of the Joint Advanced Distributed Simulation (JADS) Joint Test Force (JTF). The functionality of these automation procedures is based on the premise that federations will include detailed and meaningful object, attribute, interaction, and parameter definitions in the FOM. The JADS federation did not take this mainstream HLA implementation approach. Instead, they developed and implemented federation specific policies that were necessary to provide the interoperability support that exceeded the HLA baseline. The approach they took was to include objects and interactions in the JADS FOM that were named byte streams of specified sizes. The data structure definitions that were required to decode these byte streams were not documented in the FOM. Instead, they were defined in two separate paper specifications that are not part of the baseline HLA definition. In the absence of these automation procedures, the Analysis Federate was manually composed for application in the JADS federation. Analysis of the manual composability procedures that were used suggests that they could have been automated if the problem was looked at from a different level of abstraction. A discussion of the application of the Analysis Federate in the JADS federation, and a system design for automating the composability of a federate in federations whose FOMs use named data structures is presented.

1 INTRODUCTION

The principles and practices of simulations have evolved significantly since the first manual simulations were introduced. The advent of the digital computer, the development of simulation programming languages, and the development of simulation support environments are significant components of this evolutionary process. The United States (U.S.) Department of Defense (DoD) has played a major role in these advancements. The most recent significant advancement in the science of simulation is the DoD initiative of developing simulation support environments that are designed to facilitate the interoperability of simulations that are distributed over a computer network. This advancement allows a complex system to be modeled as a collection of subsystems. These individual subsystems are implemented as simulation components that are distributed over a computer network. The composite of the interactions among the individual distributed components replicates the functionality of the complete system that is being modeled. The DoD uses the term Advanced Distributed Simulation (ADS) to describe these simulations and their support environments. The concept of ADS includes any means to interface live, virtual, and constructive simulations and systems so they can interact with each other on a common battlefield. One definition of ADS is "any application or architecture which employs the characteristics of distribution and networking in a way which permits a number of nodes, entities, or devices (at least two) to interact with each other for some common or shared purpose…" (Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF) 1996)

The DoD's involvement in ADS research originated with the development of dedicated specialized environments that were designed to satisfy unique one-time requirements. The DoD recognized the inefficiencies in this approach and attempted to develop standardized distributed simulation support environments. The Simulator Networking (SIMNET) (Leathrum and Stoughton 1996) simulation support environment was the first of these efforts. Its more general successors were the Distributed Interactive Simulation (DIS) (DIS Steering Committee 1994) and the Aggregate Level Simulation Protocol (ALSP) (Weatherly et al. 1993) simulation support environments. DIS and ALSP were designed to support separate domains within the military simulation community. DIS was designed to support the interoperability of entity level combat simulations while ALSP was designed to support the interoperability of aggregate level combat simulations. The design premises of the DIS and ALSP interoperability support environments embraced their separate narrow domain specific focuses. These narrow focuses precluded their adaptation for use as a domain independent ADS interoperability support environment.

The DoD realized that greater efficiencies could be achieved if a general solution to distributed simulation interoperability could be developed. They attempted to solve this problem by developing an architecture-based ADS interoperability support environment. This approach separates the functionality that is generic to the interoperability infrastructure from the specifics of the distributed simulations. It treats the distributed simulations as reusable components and their common object model as a parameter. The DoD uses the term High Level Architecture (HLA) to describe its architecture-based ADS interoperability support environment.

The DoD testing and evaluation (T&E) community expressed widespread interest in using ADS to support testing and evaluation. The community viewed ADS as a technology that could potentially relieve some of the long-standing constraints and limitations that are historically associated with T&E. This need caused the Office of the Secretary of Defense, Director of Test and Evaluations to charter the Joint Advanced Distributed Simulation (JADS) Joint Test Force (JTF) in 1994 to investigate the utility of ADS for developmental and operational test and evaluation. The JADS mission calls for analytical tests of applications and broader evaluations to determine when ADS is an appropriate test tool for T&E. JADS is also responsible for developing meaningful legacy products to enhance both the T&E and the acquisition processes. (Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF) 1996)

JADS developed a series of three tests to satisfy its charter and mission. The System Integration Test (SIT) uses air-to-air missile testing to determine if ADS can be used to test areas where historical testing methods cannot be used because of safety or cost constraints. The End-to-End Test (ETE) explores how well ADS could be used to potentially overcome the Command, Control, Communications, Computers, and Intelligence (C4I) testing problem of limited numbers of test assets that do not reflect anticipated battlefield conditions. The Electronic Warfare (EW) test investigates the potential for using ADS to generate a more realistic test environment for EW systems. (Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF) 1996)

The emphasis of each of the JADS' tests is on the performance of the ADS components and their contribution to testing, rather than any particular system under test or class of weapon system. Areas of interest include network performance, relationships between data latencies, and ADS induced data anomalies. It is important to emphasize that JADS is not evaluating any of the weapon systems that are used as test cases. Time, cost, and complexity as well as validity and credibility of the data are part of the evaluation. JADS selected tests which will allow this comparison.

JADS must plan, conduct, and report on analysis of ADS data and applications from the tests to determine the utility of ADS for T&E. A central issue that must be resolved is whether or not ADS architectures can generate data that are valid, accurate, and at the appropriate level of fidelity for T&E applications. This is relevant because networking distributed live, virtual, and constructive simulations and systems imparts errors in the generation of test data. (Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF) 1996) Additionally, standard methods and tools to support data collection and analysis are not components of the new architecture-based HLA ADS interoperability support environment. The HLA ADS will not be useful for supporting T&E if data cannot be collected from the composite of the distributed components for analysis. (Buss et al. 1999)

Various data collection and analysis strategies and tools have been proposed and are being tested by JADS. This technical report describes the application of the Analysis Federate (Murphy 1998) to provide data collection, visualization, and analysis capabilities in support of the EW Phase II testing.

2 JADS ELECTRONIC WARFARE TEST

The JADS EW Test consisted of three test phases. Phase I, the baseline, used conventional EW test methods and technology to collect baseline data but did not use ADS. Phases II and III used ADS technology to link the key components in the test scenario and recreate the baseline from Phase I. In each phase, the data were used to calculate developmental and operational EW test measures of performance (MOPs). The MOPs calculated from ADS phases were then correlated to the MOPs produced by the non-ADS phase to determine what impact ADS had, if any, on the results. The first of the ADS test phases (Phase II) was completed December 11, 1998. That test recreated the earlier open-air range (OAR) and hardware-in-the-loop (HITL) testing JADS accomplished in Phase II. Although the test imitated a self-protection jammer (SPJ) effectiveness test, the purpose of the test was to evaluate the utility of ADS. The JADS EW SPJ Test applied the HLA process and components in the first ADS-based test (Phase II). The HLA is an integral component of the JADS EW Test design and represents a key element of the building block approach JADS is using to determine the current capability ADS provides for EW T&E. (Wright and Zimmerman 1999)

3 HLA DATA COLLECTION BASICS

The architecture of the HLA ADS interoperability support environment is defined by three components: the HLA Rules, interface specification, and object model template (OMT). The terms, federations, federates, and run-time infrastructure (RTI), are used in the descriptions of these components. Federations are the set of component simulations, models, or tools that interoperate with each other during the execution of a HLA distributed simulation session. Federates are the individual distributed component simulations, models, and tools that interoperate with each other in a federation. The RTI provides the functionality of a distributed operating system for HLA federations. The HLA Rules specify the responsibilities of the federation, federates, and RTI. The interface specification defines the interface services between federates and the RTI. The OMT specifies the format of the object models that are used to document the object and interaction information for federations and federates. These object models are the federation object model (FOM) and the simulation object model (SOM) respectively. The FOM and SOM serve as the HLA interface language for a federation. They facilitate the consistent interpretation of exchanged data by providing a description of objects, attributes, associations, interactions, and level of resolution. A small subset of the data from the FOM is extracted and stored in the Federation Execution Data (FED) file for use during federation execution. The actual FOMs and SOMs are not used online during a distributed simulation session, which is referred to as a federation execution. Instead, the FOM and SOM object models are paper specifications that are used as references to assist in the development of a federate's computer code that invokes the RTI services. This computer code is referred to as the federate's local RTI module.

The premise for the HLA design asserts that it is unreasonable for a single simulation to handle the technical complexity and diverse user needs that are represented in existing simulations, and that future technological innovations and simulation uses and requirements cannot be predicted. The HLA designers addressed this by developing an architecture-based ADS interoperability support environment. This interoperability support environment treats the distributed simulations (federates) as modular subsystems or components that are assembled to form a larger system (federation). The SOMs of the federates that are composed into a common federation must all be abstractions of a single common FOM that is structured around the federation's object view of the world. The HLA design includes an object oriented subscription-based communications structure that uses the federation's common object model as a parameter.

The RTI implements the HLA's subscription-based communications structure during federation execution. Federates must use the RTI publication and subscription services to produce and consume distributed data. Federates must subscribe to and will only receive the distributed objects, interactions, and parameters that they require in order to function in a distributed environment. Similarly, they publish only the data that has been subscribed to by one or more other federates. The RTI filters and routes published data to the appropriate subscribing federates.

The HLA architecture provides a baseline level of interoperability among the federates in a federation. This includes the ability to establish a federation of federates, exchange object data between federates, and coordinate federate operations. Interoperability requirements that exceed the HLA baseline must be implemented as policy atters for each federation.

The HLA design ensures that the modular federates will have well-defined functionality and interfaces that are separated from the supporting RTI. It also helps to ensure that the HLA ADS interoperability support environment can be applied in any simulation domain. This desire for generality introduced an impediment to interoperability into the HLA ADS interoperability support environment. The requirement for a federation specific FOM and corresponding, compatible, SOMs for each federate in the federation limits the potential for federate composability. The HLA design limits the composability of federates to only those federations that use a FOM that is consistent with the federate's SOM. In practice, a federate is not composable with a federation if its SOM is not an abstraction of the federation's FOM. However, any federate can potentially be modified for use in federations where it is not considered as a composable component. Unfortunately this requires that the federate developers design and implement a new SOM that is an abstraction of the federation's FOM. The functionality advertised in the newly developed SOM must be accompanied by the appropriate changes to the federate's local RTI module. Mainstream HLA implementation procedures dictate that these local RTI module changes be implemented by writing computer code. The procedures required to convert a federate for use in a federation, when the federate's SOM is not an abstraction of the federation's FOM, are onerous. They are simpler than but analogous to the procedures required to convert a legacy non-HLA compliant simulation for use in a federation. Thus, neither legacy non-HLA compliant simulations nor federates whose

SOMs are not abstractions of a federation's FOM can be considered as composable components of the particular federation in question.

The architecture requires that federate designers write computer code to build the functionality required to interface their federate with the RTI in order to participate in a federation and exchange data with other federates. This code is the mechanism that allows the HLA architecture to limit the transfer of information to only the objects and interactions that the individual federates need in order to satisfy their unique requirements. A federate's data consumption requirements and production capabilities must be identified as subscription and publication service invocations in the code that implements its local RTI module.

These obstacles to federate interoperability in distributed HLA simulations must all be addressed and solved in order to develop a general purpose tool that is required to collect, store, and analyze the composite of the distributed simulation data in any HLA federation. This general-purpose tool must be supported by a methodology that allows the tool to be reusable in a variety of federations that use different object model abstractions in the FOM. Some legacy ADS interoperability support environments use passive data loggers that capitalize on the ADS' universal object model and its broadcast network communications structure to satisfy the data collection and analysis need. The subscription-based communications structure of the HLA ADS interoperability support environment precludes the use of this passive data logger strategy in HLA simulations. Instead, a new general-purpose methodology must be developed to provide the required data collection and analysis capabilities.

The design of the HLA ADS interoperability support environment does not include a universal object model that serves as the FOM for all federations. Instead, the HLA is designed to support interoperability in any federation that uses any unique object model abstraction in its OMT formatted FOM. The non-existence of a universal object model causes many analysts to conclude that separate, specialized data collection and analysis tools must be developed for each federation that uses a unique object model abstraction in its FOM.

The U.S. Army's Training and Doctrine Command (TRADOC) Analysis Center (TRAC) in Monterey, California did not make this assumption. Instead, they developed a conceptual framework that enables a federate to be treated as a composable component of any federation that uses any object model abstraction in its FOM. This conceptual framework was used to support the development of the Analysis Federate, a HLA data collection, archival, and analysis tool.

4 ANALYSIS FEDERATE

The Analysis Federate (Murphy 1998) is a general-purpose data collection and analysis tool that is designed to function as a composable component in any HLA federation that uses any object model representation in its FOM. This requires an extension of federate functionality beyond what is called for in the HLA design principles, which promote the use of a federate as a composable component in federations that uses a single common object model abstraction in their FOMs.

The conceptual framework that supports this composability and functionality includes tools and procedures that are used to automate the development of a federate's local RTI module. These automated procedures eliminate the requirement for federate designers to write code to develop FOM specific local RTI modules. The conceptual framework accomplishes this by providing fourth generation development tools that automate the procedures required for a federate to subscribe, publish, and interpret federation data; as well as the ability to automatically generate the federate's federation specific SOM. These fourth generation development tools provide a general solution to the requirement that each federate implement FOM specific HLA service invocations in a local RTI module. The tools provide graphical user interfaces (GUIs) that eliminate the need for federate developers to write the local RTI module computer code that is required for a federate to interact with the RTI. Instead, a user's interactions with the GUI are treated as run-time parameters by the fourth generation development tool's general-purpose local RTI module.

The application of the conceptual framework that provides for the composability of federates across federations is not restricted for use only with those federates that are capable of using arbitrary object model representations in their internal algorithms and databases. Instead, the conceptual framework provides a process that allows the user to specify the mapping between the FOM object model representation and the federate's internal object model representation.

The Analysis Federate uses the object model mapping approach. The visualization and analysis functionality in the Analysis Federate uses the Vision XXI (Tapestry Solutions, Inc. 1998) GUI and database. Vision XXI uses a proprietary internal object model that cannot be modified by the user. Thus, Analysis Federate users must provide the information that is required to map a federation's objects, attributes, and interactions into Vision XXI database objects. This mapping must be done the first time the Analysis Federate is used in any federation that uses a unique object model abstraction in its FOM. This mapping would not have been required if the database and algorithms that are used by Vision XXI were originally designed to work with any arbitrary object model representation.

5 APPLICATION OF THE ANALYSIS FEDERATE IN THE JADS' EW TEST

The JADS JTF desired to use the Analysis Federate to collect, process, analyze, present, display, store, access, and transfer HLA data to provide real-time and post-simulation visualization and analysis support for the JADS EW Phase II testing. The development effort for the EW Phase II testing was drawing to a close when JADS became aware of the existence and capabilities of the Analysis Federate. They approached TRAC to determine if the Analysis Federate could be used to provide a full-featured data collection, analysis, and visualization capability on a time schedule that only allowed three weeks of development effort. The factors that were considered to determine if this time schedule could be met are discussed below.

The Analysis Federate functionality and capabilities that were described above may mislead the reader into thinking that the three week time schedule is not a significant constraint because the Analysis Federate is composable across federations and because that composability is supported by automated techniques, tools, and procedures. Normally the reader would be correct in assuming that the Analysis Federate can be fully integrated into federations in very short periods of time. However, that is not the case with the JADS implementation for the reasons that are described below.

The desire for a full featured JADS data collection and analysis solution required that the Vision XXI developer, Tapestry Solutions, Inc., make changes to the Vision XXI source code. These changes were needed because JADS required that all of the EW Phase II analysis algorithms, down to and including the Measure of Effectiveness (MOE), be incorporated directly into the Vision XXI application that serves as the Analysis Federate's integrated GUI and analysis tool. Adding this functionality directly into the Analysis Federate's GUI would allow the MOEs to be continuously updated on a real-time basis during the actual distributed simulation tests. JADS wanted to display these calculated MOE values in the same application and on the same screen that displayed the locations of the distributed entities on the exercise map during the distributed simulation session. The required visualization and calculation functionality was specified by JADS in a requirements document. (Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF) 1998b) Equations that were not included in that document were available directly from the JADS Analysis Team.

The Vision XXI application had never been previously used to support T&E prior to the JADS testing. Likewise, none of the detailed JADS analysis algorithms were ever previously incorporated into the Vision XXI application. Instead, the Vision XXI internal algorithms and the supporting database were designed to support combat maneuver visualization, analysis, and after action review for the training community. However, there are many parallels between visualizing combat operations and visualizing missiles and aircraft in a Joint T&E.

Fortunately for JADS, the Vision XXI design provides the developer with the ability to extend the basic visualization and time management functionality for use in any application area. It was well within the contractor's ability to implement the analysis algorithms and visualization capabilities during the allocated three-week time frame if that was the only task they were required to accomplish, but that was not the case. The Vision XXI developer is also the Analysis Federate contractor. Thus, they would also be required to develop the Analysis Federate's mapping between the object representations in the JADS FOM and the object representations in the Analysis Federate's Vision XXI database. An estimate of the level of effort that would be needed to develop this mapping required a review of the JADS FOM.

The federation that was established by JADS to support the EW Phase II testing was well documented. Unfortunately, many of the JADS interoperability requirements exceeded the HLA interoperability baseline that was provided in early RTI releases. Testing requirements demanded updates every twenty milliseconds, with strict controls on transmission latencies. Since the HLA and JADS development and specification efforts were being completed simultaneously, the JADS JTF was concerned that the subsequent RTI releases also might not be able to provide the performance that was required to support the EW Phase II testing. The test required a closed-loop low latency application that optimized the amount of data being passed. The JADS development team feared that the RTI would impose too much overhead in terms of both processing and bandwidth. Thus, in accordance with the HLA design principles, the JADS JTF developed and implemented federation specific policies that were necessary to provide the interoperability support that exceeded the HLA baseline. The approach they took was to include objects and interactions in the JADS FOM that were named byte streams of specified sizes. The data structure definitions that were required to decode these byte streams were documented in two separate paper specifications that are not part of the baseline HLA definition. The use of named byte streams reduced the RTI overhead which enhanced efficiency and therefore helped minimize latency. This strategy was similar to the strategy that used by the HLA Engineering Protofederation.

The use of named byte streams in the FOM posed a problem for the Analysis Federate, obviating some of its automated features. It required that the Analysis Federate source code be modified in order to provide the mapping between the object representations in the JADS FOM and the object representations in the Analysis Federate's Vision

XXI database. The code changes that were required to provide this mapping would consume the majority of the available three-week development effort if the project were pursued.

The two unique JADS specifications that document the federation's interoperability requirements that exceed the HLA baseline are the Interface Control Document (ICD) (Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF) 1998a) and an unpublished table that enumerates the meanings of the codes that represent the radar and missile guidance modes. The data structures that are documented in the ICD provide federate developers with the information they need to construct and take apart the named byte streams that are specified as the objects in the JADS FOM. Other federations whose requirements can be satisfied by the HLA's baseline level of interoperability do not include these types of byte streams in their FOMs. Instead, they use the mainstream approach of developing object model abstractions and a corresponding FOM that includes detailed objects, attributes, interactions, and parameters.

The Analysis Federate's automated subscription and publication tools work with differing degrees of success in both approaches. The JADS approach of using named byte streams in the FOM requires that the contents of the ICD and the mode enumeration table be included as code in the Analysis Federate in order to provide the mapping between the FOM objects and the more detailed and meaningful parameters that describe the objects, attributes, interactions, and parameters that are used internally by the federates. The mainstream approach of using the objects, attributes, interactions, and parameters in the FOM eliminates the need to write the mapping code, and allows the Analysis Federate to use the equivalent of a table to map between the two object representations.

The classified JADS testing environment was another factor that was considered by the TRAC and Vision XXI development team prior to accepting the mission to support the JADS experiment. The primary Analysis Federate programmer did not have a security clearance. This would require that the modifications of the Analysis Federate would have to be done outside the classified environment, without the benefit of a live data stream from the RTI. Modified Analysis Federate code would have to be brought into the classified environment for testing by a member of the development team who was not the primary programmer. The performance characteristics and proposed modifications to the Analysis Federate source code would then have to be described to the primary programmer who would make the appropriate changes which would subsequently be brought into the classified environment for testing. This was not the ideal development scenario for a project that was limited to a three-week development effort.

TRAC and Tapestry Solutions considered the above factors and decided to accept the mission of integrating the Analysis Federate into the JADS EW Phase II test. The three-week development cycle was a severe constraint, but the development team determined that the project could be accomplished on schedule with minimal risk. It was well known that another contractor was unable to provide the required functionality during their two-year development cycle. However, the existing Analysis Federate and Vision XXI products use innovative technologies that are well suited for use in these types of applications. They provide the basic HLA and visualization functionality that can be easily extended for use in new application areas.

The EW Phase II testing began on schedule three weeks after the Analysis Federate integration effort began. The Analysis Federate provided the required visualization and MOE calculation capabilities on a real-time basis during the distributed simulation test. It was also successfully used to support post-test analysis efforts. Specifically, the Analysis Federate significantly improved the JADS EW Phase II test control and analysis capabilities in the following areas:

- Real-time displays of all MOPs - This was by far the largest improvement over JADS' original analysis/display capability. Other than calculating some MOPs by hand in real-time, there was previously no way to display all MOP information on the same screen during a real-time test run. Not only did the Analysis Federate display the raw MOP numbers, it presented them cumulatively so the analyst knew exactly which portions of the engagements generated which MOP data points.
- Real-time display of missile fly-outs - This capability helped in correlating the EW MOPs of tracking error (TE) and jamming-to-signal ratio (J/S) data during threat missile fly-out.
- Better graphical display of TE and J/S - The data for the entire engagement were displayed, as opposed to about 30 seconds of the engagement when using the test team's original analysis/display capability.
- Real-time display of aircraft range from threat - This capability greatly enhanced the test controller's situational awareness by calculating and displaying the numerical distance of the aircraft from the threat from test run start to test run finish.
- Display of missile distance from aircraft - The Analysis Federate improved test controller situational awareness by displaying

real-time missile trajectories and distances to the aircraft.

- Aircraft script validation - The Analysis Federate provided confirmation of the validity aircraft scripts. For instance, the analysts and test controller could immediately determine if the aircraft flight profile was correct, or if Time-Space-Position-Information (TSPI) data from the aircraft platform federate had been lost.
- Threat missile system status - The Analysis Federate displayed the on or off status of threat missile systems. This capability helped validate that threat systems were behaving in accordance with rules of engagement.

6 ANALYSIS FEDERATE LESSONS LEARNED

An analysis of the federation that was used in the JADS EW Phase II testing, and the corresponding development effort that was required to integrate the Analysis Federate into that federation was conducted. That analysis revealed that the problem of mapping the named byte streams from the JADS FOM into the Analysis Federate's database could have potentially been automated if the problem was approached from a different level of abstraction.

This new level of abstraction acknowledges that the data that are transferred between the RTI and the federates in a HLA federation are fundamentally named byte streams. Furthermore, the key for decoding these named HLA byte streams are the descriptions of the objects, attributes, interactions, and parameters that are contained in the federation's FOM. The Analysis Federate can only decode down to the level of detail that is provided in the FOM. If the level of detail of the FOM objects and interactions is named byte streams that have no context, then that is the lowest level of detail that those objects can be resolved at by using the information in the FOM. However, the key for further decoding those named byte stream objects into actual usable objects must exist as data structures that are recorded in a document whose format is unique to the federation, like the JADS ICD. Two proposed automation procedures will capitalize on the contents of this document.

The first proposed automation concept calls for the Analysis Federate to use two SOMs. The first SOM will be the standard HLA SOM that is required by the HLA Rules. As usual, this SOM will describe the federate's public interface to the federation. The Analysis Federate will use this traditional SOM and the federation's FOM to automate the subscription and publication services, just as it always has. The departure from the existing Analysis Federate techniques is that a second SOM will be developed for use by the Analysis Federate. This second SOM, which will be referred to as the "Data Structure SOM," will be an extension of the FOM, and will incorporate the data structures that define the named byte stream objects that are specified in the FOM. This will be accomplished by replacing the named byte streams with objects, attributes, interactions, and parameters. The names, native data types, and other essential object information will be extracted from the document that defines the data structures that describe the named byte stream objects that are defined in the FOM. The Data Structure SOM will represent the objects, attributes, interactions, and parameters that would have been in the federation's FOM if the HLA baseline definition was sufficient to satisfy the federation's interoperability requirements.

The Data Structure SOM will be used internally by the Analysis Federate as an automation tool. The Analysis Federate will use the Data Structure SOM to construct and take apart the byte streams that are transferred between the RTI and the federates. This procedure eliminates the need to write the mapping code, and allows the Analysis Federate to use the equivalent of a table to map between the two object representations. The internal use of the Data Structure SOM by the Analysis Federate will be transparent to the RTI and the other federates in the federation. The Analysis Federate will still use the naming conventions in the federation's FOM to describe the byte streams that are constructed from the objects in the Data Structure SOM and exchanged with the RTI.

A potential problem with the Data Structure SOM approach is the that workstation manufactures and their operating system developers use diverse big and little Endian implementations. These diverse representations use different byte packing schemes which must be addressed.

The second proposed automation concept requires extensions to the fourth generation development tools. These tools could be modified to include an interface that allows the user to manually input the mapping between the byte streams and the underlying object, attribute, interaction, and parameter representations. The fourth generation development tool would treat this user provided mapping as a run-time parameter which would allow the tool to automatically take apart the byte streams and apply the appropriate context to the data. This approach eliminates the need to develop the Data Structure SOM. It is analogous to the existing Analysis Federate functionality that allows the user to manually input the mapping required to automatically decode enumerated data during run-time.

A significant time savings benefit can be achieved in the federate development process when either of the proposed approaches is used. Manually writing code to construct and take apart byte streams is a labor-intensive process that can be eliminated by the software reuse benefits that are associated with the proposed procedure. Use of the fourth-generation development tool approach by will be more straightforward and less labor intensive. Developing Data Structure SOMs will require the user to be knowledgeable about HLA development procedures and standards. The most appropriate approach will depend on the intended application and future changes to the OMT standard.

Future applications of the Analysis Federate into federations that use named byte streams as FOM objects should include any development efforts that are required to incorporate this new automation functionality into the Analysis Federate. The overhead associated with incorporating the Data Structure SOM capability into the Analysis Federate will satisfy what appears to be a need that will widespread within the T&E community.

7 CONCLUSIONS

The Analysis Federate was successfully incorporated into the federation that was used in the JADS EW Phase II testing. This integration effort was accomplished within the available three-week development cycle, but was not as automated as one would expect. The primary reason for limited automation of the development process is that the objects in the JADS FOM are named byte streams with no inherent contextual representation. The data structures that define those object byte streams are recorded in separate documents that are unique to the federation. The lessons learned from this research include a system design could be used to provide additional automation to the federate development process when the objects or interactions in the FOM include named byte streams. This functionality could be incorporated into the Analysis Federate or any other HLA federates that requires the ability to be treated as a composable component of any federation that uses any object model representation in its FOM.

REFERENCES

Buss, A. H., Knight, S. D., Murphy, W. S. and Jackson, L. A. 1999. Practical Use of an HLA Analysis Federate, *Proceedings of SimTecT99*, Melbourne, Australia, The Simulation Industry Association of Australia (SIAA): Vaucluse, NSW Australia, March 1999.

DIS Steering Committee. 1994. *The DIS Vision: A Map to the Future of Distributed Simulation Vsion 1.0*, Orlando, Florida: University of Central Florida Institute for Simulation and Training, May 1994.

Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF). 1996. Joint Advanced Distributed Simulation Joint Test and Evaluation Program Test Plan, Albuquerque, New Mexico: Department of Defense, February 1996.

Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF). 1998a. *Joint Advanced Distributed Simulation Electronic Warfare Test Phase High Level Architecture Interface Control Document (ICD) Version 1.3*, Albuquerque, New Mexico: Department of Defense, June 1998.

Joint Advanced Distributed Simulation (JADS) Joint Task Force (JTF). 1998b. *Analysis Federate Requirements* , Albuquerque, New Mexico: Department of Defense, September 1998.

Leathrum, J. F. and Stoughton, J. W. 1996. Technology Transfer Applications for Distributed Simulation Technologies, *Proceedings of the 1996 Summer Computer Simulation Conference (SCSC 96)*, Ingalls, V. W., Cynamon, J. and Saylor, A. V., eds., Portland, Oregon, The Society For Computer Simulation International: San Diego, California, July 1996.

Murphy, W. S. 1998. Analysis Federate, *Fall 1998 Simulation Interoperability Workshop -- Workshop Proceedings and Presentations*, Paper 98F-SIW-235, Orlando, Florida, Simulation Interoperability Standards Organization: Orlando, Florida, 1998.

Tapestry Solutions, Inc. 1998. *Vision XXI--An AAR Capability Resulting From a C3 Perspective,* San Diego, California: Tapestry Solutions, Inc., 1998.

Weatherly, R. M., Wilson, A. L. and Griffin, S. P. 1993. ALSP - Theory, Experience, and Future Directions, *Proceedings of the 1993 Winter Simulation Conference*, ed. G.W. Evans, M. Mollaghasemi, E.C. Russell, and W.E. Biles. Institute for Electrical and Electronic Engineers: Piscataway, New Jersey, 1993.

Wright , D. and Zimmerman , P. M. 1999. Test and Evaluation Federation Experience and Lessons Learned Using the Federation Development and Execution Process, *Spring 1999 Simulation Interoperability Workshop Proceedings*, Paper 99S-SIW-173, Orlando, Florida, Simulation Interoperability Standards Organization: Orlando, Florida, 1999.

AUTHOR BIOGRAPHIES

WILLIAM S. MURPHY Jr. is a Major in the U.S. Army Corps of Engineers with 17 years of commissioned service. He graduated with a B.S. from the United States Military Academy in 1982 and with a M.S. in Mathematics and Computer Sciences from the Colorado School of Mines in 1992. Major Murphy is a licensed Professional Engineer (PE) in the state of Virginia. He is currently an operations research analyst at the U.S. Army Training and Doctrine Command (TRADOC) Analysis Center (TRAC) in Monterey, California. He is also the Advisor for the Military Operations Research Society's (MORS) Computing Advances in Military Operations Research working group.

MICHAEL L. ROANE is a Major in the U.S. Army with 17 years of commissioned service. He graduated from the

University of Mississippi in 1981 with a B.A. in Economics and with a Masters in Business Administration from Duke University in 1991. He is currently serving as the Analysis Team Lead with the Joint Advanced Distributed Simulation (JADS) Joint Test Force (JTF) in Albuquerque New Mexico.