Extended Air Defense Testbed (EADTB) Migrates Toward High Level Architecture (HLA) Compatibility

Ronald Sell
Jim Towers
Gary Ballard
Jay Graham
Patty McMahon
The AEgis Technologies Group, Inc.
6703 Odyssey Drive, Suite 200, Huntsville AL 35806
(rsell, jtowers, gballard, jgraham,pmcmahon) @aegistg.com

Keywords:
HLA Migration, EADTB, SOM development, Middleware development, FOM-independence

ABSTRACT: The Extended Air Defense Testbed (EADTB) has completed two progressive migration phases towards High Level Architecture compatibility. The first phase was a proof-of-principle study using a DIS/HLA gateway. The second HLA migration phase for the EADTB was implemented using a HLA middleware component. This middleware option eliminated the use of DIS PDUs, which were used in Phase 1, as a means of communication in the HLA environment. The middleware layer that was integrated between EADTB and the Run Time Infrastructure (RTI) provided a Federation Object Model (FOM) independent capability by introducing a Simulation Object Model (SOM)/ FOM data translation layer. This middleware layer is referred to as the Object Model neutral interface (OMni™). In order to support the integration of the OMni middleware, the EADTB SOM was developed and intended to be an initial EADTB SOM (Version 1.0) documenting EADTB's Capability 4.4 without the use of DIS/HLA gateways. The primary focus used in developing the SOM was to highlight the internal capability and modeling flexibility of EADTB, but other considerations that did impact the SOM development included SOM expandability with regard to future EADTB releases and to further EADTB's prospective analytical role in future federations. With EADTB's modeling flexibility reflected in the SOM and OMni's FOM independent capability, EADTB has the potential to participate in any suitable federation. This flexibility was demonstrated in a Phase 2 demonstration using the existing JWP Trailblazer/Pegasus Build 0 FOM. The OMni middleware was used to provide the necessary translations and mappings between the EADTB SOM and the Pegasus FOM. This paper will present the details of the HLA migration phases for the Extended Air Defense Testbed (EADTB), the SOM development approach taken to develop the EADTB SOM version 1.0, the flexibility of EADTB's OMni middleware, and the results of the HLA demonstrations performed under each migration phase.

1.0 Introduction

The Extended Air Defense Testbed (EADTB) offers a broad range of applicability from the fire unit level to the theater level in a constructive simulation framework. EADTB is an extremely modular, flexible, object-based constructive simulation-representation and simulation-operation framework. It has been particularly configured for the Ballistic Missile Defense Organization (BMDO) to analyze and accurately model the battle management command, control, communications and computer (BMC4I) assets associated with the complex BMDO missile defense family of systems interoperabilty issues. EADTB, therefore, is extendable to other BMC4I experiments and analyses beyond missile defense issues. Specific System Representations (SSRs) may be composed by users who identify component primitives, provide physics-level attribute parameter values, and establish operational behaviors and decision processing by means of programmable 'rule-sets'. This flexibility allows the analyst to apply a high-detail SSR to simulate a key system, with SSR elements to represent components of upper tier (Theater High Altitude Area Defense-like)/ lower tier (PATRIOT-like) enclave at a critical location, while simulating interoperability in the surrounding theater context (including naval and air missile defense components) with a lower-detail and/or higher-aggregation representation. The rule-set-based EADTB SSR language will support growth beyond Extended Air Defense to model explicitly both surface warfare and national missile defense at a similarly broad range of levels of detail and aggregation.[1]

This modeling flexibility of EADTB and the ability for users to continue to define rule-sets made the HLA migration very challenging. The HLA migration of EADTB needed to develop a slow changing and persistent SOM, while maintaining EADTB's wide modeling capability and its ability to provide visibility to the models in an HLA federation. With these objectives in mind, an HLA migration plan [2] was developed that outlined a phased approach for obtaining HLA compliance for EADTB. The following sections discuss each phase of the migration plan in detail.

2.0 EADTB Gateway Migration

2.1 Background

The first HLA migration phase outlined for EADTB was a proof-of-principle study using a DIS/HLA gateway. The purpose of this initial phase was to achieve HLA capability to the same level as their DIS capability. While the vision for EADTB went beyond the current DIS capability, several benefits were anticipated from an HLA/DIS gateway implementation. The following subsections identify the HLA/DIS gateway migration approach. [3]

2.2 Requirements and Objectives

The EADTB program wanted to implement an HLA migration solution that would demonstrate EADTB's HLA capabilities without impacting the current development cycle for the EADTB software release. With this guidance, the following requirements were developed for this HLA migration phase.

1. EADTB will operate in an HLA federation with multiple EADTB's and other non- EADTB federates.
2. EADTB will demonstrate HLA capability using an HLA/DIS gateway.
3. EADTB will use the RPR FOM V0.3 or subset in the federation.
4. Provide multiple HLA federation demonstrations at the Raytheon Software Development Facility (SDF) and the Colsa Advanced Research Center (ARC).

With these requirements defined, the anticipated benefits for the initial migration phase were identified in the following experiment objectives.

1. Develop a scenario to exercise HLA interoperability between EADTB and EADTB.
2. Develop a scenario to exercise HLA interoperability between EADTB and non-EADTB federates.
3. Compare EADTB's DIS and HLA execution of the same scenario.
4. Use AEgis' FedProxy to play one or more scenario elements in the EADTB scenarios during an HLA exercise.
5. Compare the IST and SMOC gateway versions during execution.
6. Identify and execute HLA Interface Specification services to be used by the HLA compliant EADTB.
7. Investigate areas of interest: repeatability, scalability, network latency and traffic, network loads, message loss, and time management schemes.
8. Draft EADTB SOM unconstrained by the use of a gateway reference FOM.

2.3 Federation Design and Execution

The war-fighting scenario, ***Figure 2.3-1***, was developed to contain friendly defended assets whose lower tier TBM defenses were attacked by enemy cruise missiles in an air-defense-suppression raid. The cruise missiles were in turn engaged by friendly SHORAD units. In addition to the cruise missile threat, enemy TBMs' attacked against the upper tier defended assets that were also engaged by the friendly active TBM defense.

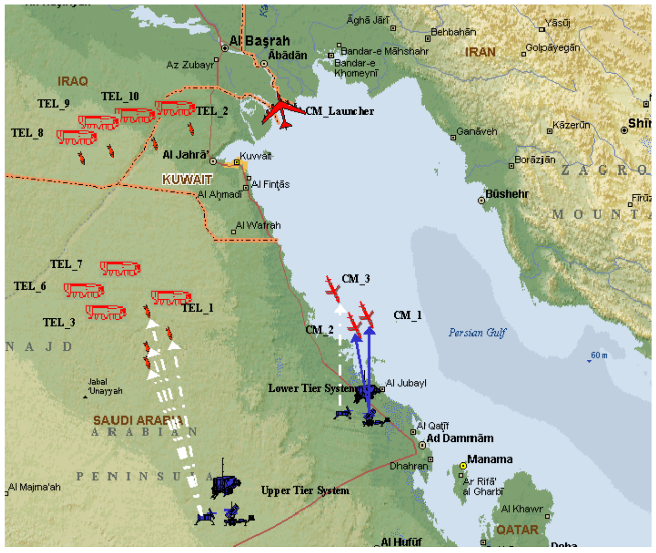

Figure 2.3-1. Scenario Laydown

Figure 2.3-2 shows the federation members that were used to execute the scenario described. Blue forces were modeled by EADTB #1 and Red forces were modeled by EADTB #2 and FedProxy. This division of modeling provided the necessary interoperability defined in the experimental objectives.

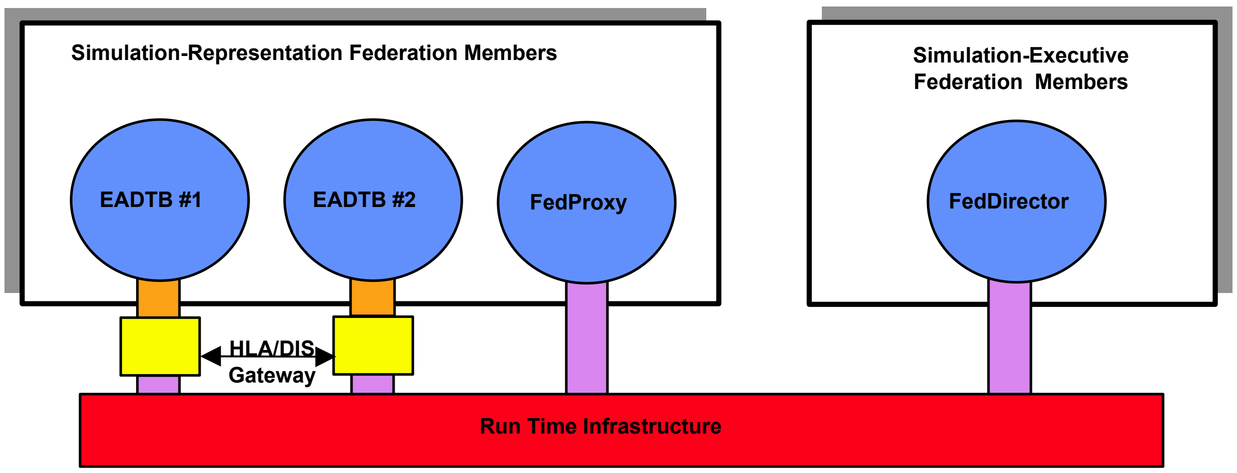
Simulation-Representation Federation Members
EADTB #1
EADTB #2
FedProxy
HLA/DIS Gateway
Simulation-Executive Federation Members
FedDirector
Run Time Infrastructure

Figure 2.3-2 Federation Membership

2.4 Lessons Learned

As a result of the initial HLA migration phase, EADTB proved capable of federating in a HLA federation. The EADTB federation experiments observed the following characteristics during execution.

1. Repeatability of scenario was achieved in DIS and HLA.
2. Minimal time lags occurred resulting in real time execution.
3. No additional latency was observed from introducing an HLA Gateway.
4. Scenario execution was the same in DIS and HLA.
5. Minimal time delays were observed between the FedProxy ghosted cruise missile and the EADTB ghosted cruise missile.

In addition to the federation execution observations, pros and cons were identified for continuing to use the HLA/DIS gateway for EADTB's HLA interface.

Gateway Pros
1. Inexpensive
2. Doesn't require source code modifications to EADTB
3. EADTB will still use its existing native DIS protocols

Gateway Cons
1. EADTB will be restricted to use a version of the RPR FOM.
2. Required to run real time
3. Limited by use of DIS
4. Native HLA federates in federation would need to provide "DIS Type" information, i.e. conform to RPR FOM.
5. Will limit the number of federations in which EADTB can federate in.

As a result of these observations and lessons, a second HLA migration phase was planned to integrate an HLA middleware layer.

3.0 EADTB Middleware Migration

3.1 Background and Requirements

The HLA middleware migration for EADTB was divided into three components. The first component was the SOM development activity, second, the development of the HLA Software Interface Unit (HIU) to interface between EADTB and OMni, and third, the development of the OMni middleware. Each migration component is discussed in the following sections.

3.2 SOM Development

In determining the publishable object classes, consideration was given to the specific functionalities that EADTB could make available to other EADTB federates and non-EADTB federation participants. The Specific System Representations (SSRs) in the EADTB Master Library were reviewed and categorized into real-world entities representing candidate object classes, and the potential behaviors of each of these entities were evaluated for determining candidate interaction classes. For subscription object and interaction classes, consideration was given to entities and actions for which EADTB is capable of accepting and processing meaningful data from other simulations. In conjunction with these activities, data involved in inter-Scenario Element (ScenEl) relationships was extracted from the Experiment Definition Data and the Common Model Set (CMS) Software Requirement Specification (SRS) documents. All of this information was captured in a draft presentation to Raytheon of EADTB's publishing and subscription capabilities for object and interaction classes and their associated attributes and parameters.[4] Through a series of iterations and detailed discussion, the object class structure shown in ***Table 3.2-1*** was developed.

Table 3.2-1 EADTB Object Class Structure

Scenario_Element	Guided_Missile_Element
	Ballistic_Missile_Element
	Winged_Airborne_Element
	Sea_Element
	Space_Element
	Land_Element

For interaction classes, the Fire, Detonation, and Signal PDUs (corresponding to the Launch, Detonate, and Transmit interaction classes) were considered. However, the ability of the signal PDU to adequately support the transmission of all EADTB messages in the HLA environment was less than desirable. As an alternative, the SOM structure for the Transmit interaction class was developed to support the EADTB message set that is represented in the EADTB Message Protocol Table. Four options were considered for representing the message structure in the SOM, summarized as follows: [4]

1. Separate interaction class per message.
2. Separate interaction class per protocol using specific field values as defined by various messages within the protocol.
3. Separate interaction class per protocol using generic real, text, and logical parameters.
4. One interaction class using a protocol parameter and generic real, text, and logical parameters.

Although it was recognized that a generic message representation provided limited visibility to the protocols currently supported in EADTB, option four was chosen. This decision was based mainly on the flexibility provided to define any messages within the Message Protocol Table, including user-defined protocols, without modifying the SOM. As a result, ***Table 3.2-2*** describes the interaction class structure for EADTB.

Table 3.2-2 EADTB Interaction Class Structure

Transmit	
Launch	
Detonate	
Shared_Launch	
Simulation_Management	Start
	Stop
	Pause
	Resume

The relative simplicity of EADTB's SOM was deliberate and by design to maintain three key elements in the ability for EADTB to participate in HLA federations.

1. Generic structure of SOM supports the expandable nature of EADTB software using Specific System Representations (SSR).
2. Generic structure in conjunction with OMni middleware allows participation in virtually all relevant FOMs.
3. SOM structure is easily expandable to support additional EADTB capabilities in HLA federations (i.e. intra-ScenEl interfaces, user defined interaction classes, etc.).

As a result, the simplistic SOM becomes EADTB's strength in an HLA federation. EADTB will continue to have the same flexibility and be capable of utilizing EADTB's potential capabilities in an HLA environment. However, the simplistic SOM also supports the need to have a robust FOM-independent middleware for providing translations to various federation FOMs.

3.3 HLA Interface Unit

The HLA Interface Unit (HIU), shown in ***Figure 3.3-1***, was developed by Raytheon Systems to be the software component that provides the transfer of model data between EADTB and the OMni middleware. [5] The HIU was developed in Ada 95, which allowed for the Ada 95 API of the RTI to be used as the API between the HIU and OMni. Using this API provided a seamless interface between the HIU and the RTI whether or not OMni was utilized as a middleware application. In instances where EADTB only federates with other EADTB's, the translation layer will not be necessary. The HIU Ada function calls to the RTI are invoked to call OMni C methods that in turn issue calls to the RTI. In addition to the programming interface, implementation details of the HIU development have continued to evolve.

The HIU provides all data to OMni in accordance with the SOM and the FOM independence remains the responsibility of OMni.

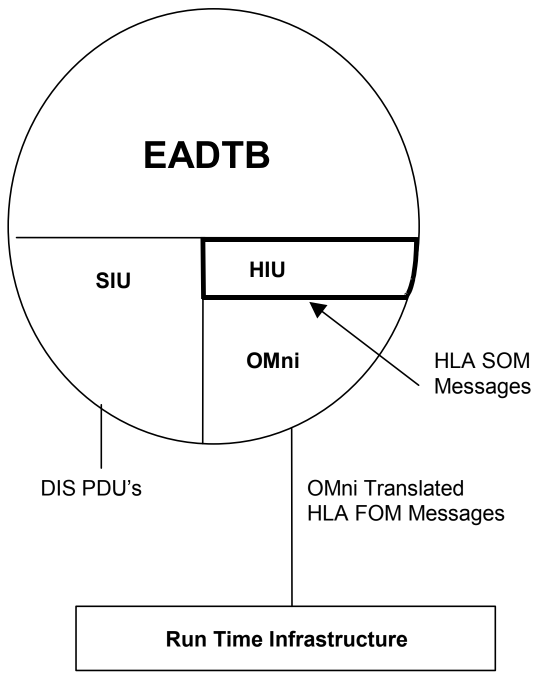

Figure 3.3-1 EADTB HLA Interface Unit

3.4 OMni Integration

OMni (for Object Model-neutral interface) is a set of related software components and applications that together provide a simulation with the means of establishing a FOM-independent interface to the RTI. In addition, it offers a type-safe interface that will tremendously impact the testing and integration process of HLA migration as well as HLA federation development. To accomplish this, OMni is based on the assumption of a slow-changing simulation object model (SOM). The SOM is an HLA object model that describes the internal object view of a particular simulation. As shown in ***Figure 3.4-1***, the OMni Runtime Layer has two primary interfaces. Its first interface is with the OMni-based simulation and is derived from the simulation's SOM. It is important to emphasize that with OMni, it is beneficial for the SOM to actually be indicative of what the simulation is doing internally.

The OMni Runtime Layer's other primary interface is with the RTI. This interface is continually adapting to different FOMs in order to allow the simulation to support different federations. OMni's interface to the RTI is derived from the FOM of the federation being adapted to (referred to as the "target FOM"), and a mapping definition read by OMni during an initialization step. As shown in ***Figure 3.4-1***, the mapping definition is created by an application called the FOM Mapper™. The FOM Mapper reads the simulation's SOM and the target FOM, and allows the user to specify how the classes, attributes, interactions and parameters of one model are manifest in the other. The FOM Mapper allows the user to designate Translator Modules to perform simple operations such as unit conversions, or more complex operations such as buffering attributes from multiple objects in the SOM domain that correspond to a single object in the FOM domain.

The OMni translation layer is responsible for providing the FOM independence for EADTB. The translation layer invokes rules that describe the predefined mappings between the EADTB SOM and the target FOM and execute any conversion routines that are defined by the translation rules. The translation rules and conversion routines are developed prior to execution and independent of executing EADTB. The rules are compiled and linked to OMni during runtime through a shared object library. This implementation negates the need to recompile EADTB source code and OMni source code for each new federation. The language used to develop the translation rules is a combination of SQL based scripting language and C++ algorithms. With the released version of OMni, these translations can be developed and maintained, using FOM Mapper, by the EADTB end user and not the EADTB or OMni developers. Therefore, a single release of EADTB with the integrated OMni middleware would be able to support numerous federations without any modifications to source code or the SOM. [6]

3.5 Federation Design and Execution

Studies were performed to identify federations with which EADTB could federate after the completion of this migration phase. The Pegasus Federation (previously known as Trailblazer Federation) that is being sponsored by the JWP and supported by the Defense Modeling and Simulation Office (DMSO) was identified as the lead candidate, although other federations were also considered.

AEgis Technologies, formerly AEgis Research, obtained a copy of the Pegasus FOM, Pegasus Federation Implementation Document (FID), and the EADSIM (site modified, non-baseline) SOM. These documents were reviewed in detail to evaluate their compatibility with the EADTB SOM

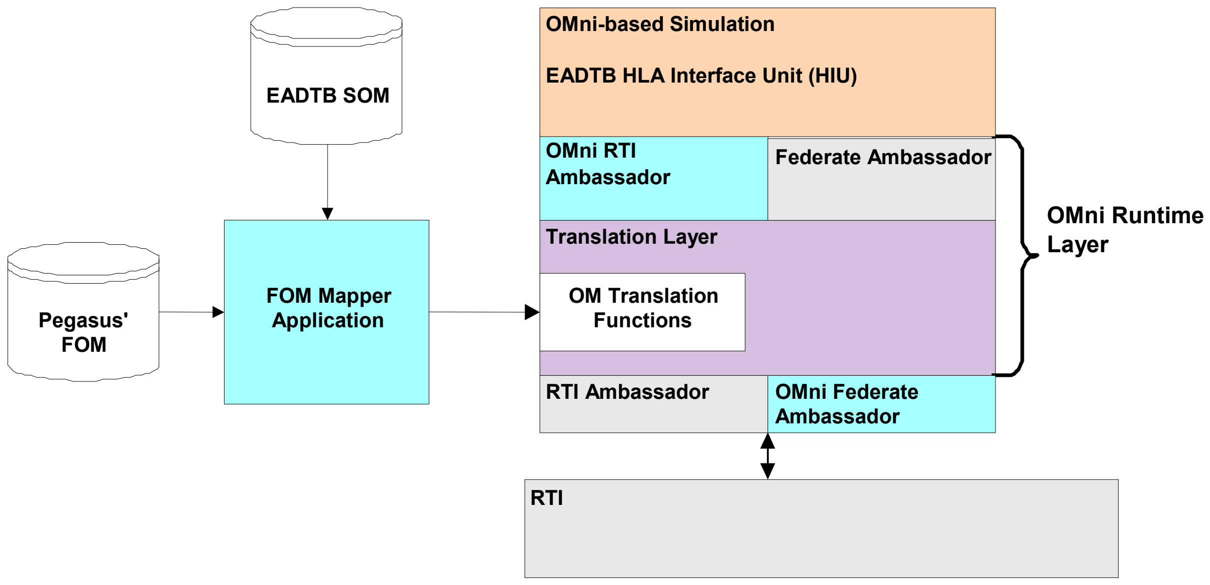

Figure 3.4-1 User's View of OMni

structure. With EADTB's FOM independent capability, the Pegasus FOM appeared to be a suitable candidate for the EADTB's initial federation participation.

As efforts continued with the investigation of the Pegasus federation and the development of the EADTB SOM, initial mappings between the SOM and FOM were recognized. As a result, the object and interaction class mappings were determined, and associated translations for these mappings were developed.

Tables 3.5-1 and ***3.5-2*** describe the object and interaction class mappings between the EADTB SOM and the Pegasus FOM.

Table 3.5-1 Object Class Mapping

EADTB SOM Class	***Pegasus FOM Class***
Guided_Missile_Element	Air.Fixed_Wing.Other
Winged_Airborne_Element	Air.FixedWing
Space_Element	Air
Land_Element	Ground.C2
Land_Element	Ground.Artillery
Land_Element	Ground.SAM.TEL

As the federation integration activities proceeded, it was discovered that the datatypes associated with the Pegasus FOM object class attributes and interaction class parameters were passed through the FOM as type STRING and not the types indicated in the Object Model Template (OMT) format of the Pegasus FOM. Because of this discovery, the SOM to FOM translations that were developed for EADTB were not valid since other federates in the Pegasus Federation were expecting all datatypes to be STRING. Due to this discrepancy between the Pegasus FOM OMT documentation and the Pegasus FOM implementation, the decision was made to continue to use the Pegasus FOM and to modify the scenario to only include multiple instances of EADTB as participating federates instead of federating with the site modified version of EADSIM used in the Pegasus federation.

Table 3.5.2 Interaction Class Mappings

EADTB SOM Class	***Pegasus FOM Class***
Transmit	Transmit.Report.Detection
Transmit	Transmit.Order
Detonate	Engage.Ground_Air
Detonate	Engage.Ground_Ground
Simulation_Management. Start	Simulation_Management. Start
Simulation_Management. Stop	Simulation_Management. Stop
Simulation_Management. Pause	Simulation_Management. Pause
Simulation_Management. Resume	Simulation_Management. Resume

4.0 Conclusions

The HLA migration of the EADTB continued under a Phase 2 effort that focused on the development and implementation of a software middleware layer between EADTB and the RTI to achieve HLA interoperability. In order to support the HLA middleware solution, the EADTB SOM was developed, which is used to define the data passing interface between the middleware layer and EADTB.

For this phase EADTB selected OMni as its middleware solution. By selecting OMni, EADTB gained the ability to establish a FOM independent interface to the RTI. FOM-independence is important when trying to interface EADTB to various HLA federations. OMni provides a stable API for EADTB that is equivalent to the HLA RTI API. Programming libraries are written to provide the translations between EADTB's SOM and the various targeted FOMs. Thus, there is no need to recompile EADTB or OMni source code. The use of the released version of OMni will allow end users, not EADTB developers, the flexibility to federate with multiple federations without the need to recompile EADTB or OMni source code.

As a result, EADTB utilizing its flexible development and modeling capability, its versatile SOM, and the FOM independent capability of OMni, makes EADTB a very powerful and adaptable simulation in an HLA federation environment.

5.0 References

[1] World Wide Web site for United States Army Space and Missile Defense Command (USASMDC), http:\\www.smdc.army.mil.

[2] Extended Air Defense Test Bed (EADTB) High Level Architecture (HLA)Migration Plan -Course of Action Analysis, AEgis Technologies, May 1998.

[3] EADTB HLA Migration - Project Review, AEgis Technologies, presented at EADTB/EADSIM Combined Users Conference, October 5-9, 1998.

[4] EADTB Phase 2 Implementation Analysis Plan, AEgis Technologies, August 1999.

[5] Extended Air Defense Testbed (EADTB) Migrates Toward High Level Architecture (HLA) Compatibility, AEgis Technologies, presented at South Eastern Simulation Conference (SESC), October 6-7, 1999.

[6] Using OMni ^{TM} as an Interface to the HLA RTI, Ken Hunt, Jay Graham, AEgis Technologies, 1999.

6.0 Biographies

RONALD SELL is a Simulation Engineering with The AEgis Technologies Group, Inc, (formerly AEgis Research Corporation) and received his BSEE, MS at Auburn University. Mr. Sell is responsible for the management of several projects related to the DoD High Level Architecture (HLA). Presently, Mr. Sell is the project manager, under contract with Coleman Research, of the HLA migration activities for the Tactical Simulation Interface Unit (TSIU). His activities for the TSIU program include oversight of the SOM development, OMni middleware integration, federation development and execution activities. Previously, Mr. Sell was the project manger for the HLA migration of the Extended Air Defense Testbed (EADTB) under a subcontract to Raytheon Systems Corporation. Mr. Sell was technically involved with the HLA Simulation Object Model (SOM) development and HLA middleware software development and integration for EADTB. These tasks moved EADTB towards HLA compliance. Mr. Sell was the project manager and also technically involved with the initial EADTB HLA migration effort. As a result of the initial HLA migration effort, EADTB was demonstrated in an HLA federation with EADSIM. In addition to the EADTB HLA migration efforts, Mr. Sell coordinated an AEgis Technologies IRAD project which integrated EADTB into an HLA federation with EADSIM, MCS/TSIU C4I system, and HLA LabWorks commercial software products developed by AEgis Technologies. This federation was demonstrated at the I/ITSEC and ITEA conferences in December 1998 that provided extensive visibility of EADTB to the modeling and simulation user community.