A SYNTHETIC ENVIRONMENT FOR C2 SYSTEM INTEGRATION BASED ON EMERGING INFRASTRUCTURE STANDARDS

Dr. George F. Providakes and R. Douglas Flournoy
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420
(781) 377-1760
rflourno@mitre.org

Key Words: Distributed Simulation, HLA, DII COE, C2 System Integration

ABSTRACT

To maintain information superiority as warfighting methods and technologies evolve, increasing quantities of time critical information must be shared between C2 systems. To meet these needs, interoperability between C2 systems must be driven to a new level. A synthetic battlespace testbed environment can facilitate this interoperability by providing a realistic context within which C2 systems can operate to refine information exchange mechanisms and processes. Existing simulations can be linked together to compose these synthetic battlespaces. Migrations to (1) the High Level Architecture (HLA) for DoD simulations and (2) the Defense Information Infrastructure (DII) Common Operating Environment (COE) for C2 systems offer opportunities to ease the composition of synthetic battlespaces and simulation-to-C2 connections. At the Electronic Systems Center (ESC), developing agency for many of the Air Force's C2 systems, a team of researchers is evaluating options for COE-based systems interoperability with HLA-compliant simulations within a framework to support C2 system integration and testing. This paper presents the vision for this framework and lessons learned from prototyping efforts to date.

ACKNOWLEDGEMENTS

This report is based on the results of a project led by co-author Douglas Flournoy. Dr. Providakes, who will be presenting the paper, contributed to the material in an advisory capacity. The vision for the framework discussed in the paper was a direct result of conversations with Mr. Allan Shanahan, a founding member of the ESC Modeling and Analysis Simulation Center (MASC). Many of the design concepts implemented during the project's execution were contributed by Mr. Jonathan Prescott, also an original member of the MASC.

INTRODUCTION

This paper will begin by discussing the trend toward increased interoperability between C2 systems and the challenges this creates for testing and training. Then, overviews of the HLA and DII COE are provided, and an approach is presented for leveraging the two paradigms to enhance simulation-to-C2 system interoperability. A target framework is presented that will provide a layered battlespace environment built around HLA and COE infrastructure capabilities, including pre-runtime tools to assist with the integration of simulations and C2 components into the framework and runtime tools to monitor and control components participating in the framework.

INCREASED INTEROPERABILITY REQUIRES ENHANCED TESTING AND TRAINING CAPABILITIES

In order to provide effective testing and training for C2 systems, it is necessary to represent the set of system input feeds and output connections that occur during battle. Simulations can be used to generate these operational conditions. If, for a given C2 system, the number of data connections involved is small and the variety of possible data scenarios is manageable, simple test jigs and output stubs may be sufficient for certain testing and operator training purposes. Traditionally, such single-system testbeds sufficed where systems were operated in a "stovepipe" manner-- that is, inputs were received, processed, and results generated without significant interaction with other C2 systems.

The current trend in C2 is toward increased inter-system communications or "interoperability." To maintain information superiority in today's warfighting environment, more and more time critical battlespace data must be shared between systems. In response to this need, related C2 systems within the Armed Services are being merged into "systems of systems." Architectures are being specified to aid in the creation of a single next-

generation Integrated C2 System (IC2S). Meanwhile, increasing emphasis on joint operations is forming complex dependencies between systems across different Services. The complexity of data flows between systems is increasing by orders of magnitude. This in turn increases the complexity of tools and processes required to test systems to insure each data exchange is executed properly.

Increased interoperability between C2 systems also brings with it an additional training burden. For interoperability to translate to operational effectiveness, system users must now more than ever "train as they fight," training on operational consoles and making use of all the information available to them from other systems in the battlespace.

Consequently, the days of simple test jigs providing realistic C2 system testing and training are over. A more complete representation of the entire battle, or "synthetic battlespace," is needed. Such a synthetic battlespace provides simulated representations of the other systems in the battle and models the impact of decisions likely to be made by operators of those remote systems. Only by exercising C2 software within such battlespaces can the intended interoperability of the C2 system be verified and users of the system effectively trained to take advantage of the new capabilities.

The Air Force is testing C2 interoperability concepts at ESC's C2 Unified Battlespace Environment (CUBE). Over the past 5 or 6 years simulation capabilities and methods for connecting to C2 systems have been applied with mixed results. Many connections have been made from simulations to C2 systems, and some success has been realized connecting simulations to each other. However very few of the simulation and C2 components were designed with this kind of data exchange in mind. If a system has a capability for exchanging data with external sources, it us usually a different mechanism than many of the other systems in the battlespace. So there have been many custom software development efforts necessary to achieve minimal interoperability at the expense of considerable effort.

Now, with the emergence of the COE and the HLA, there is the potential to leverage these middleware paradigms to provide a framework that encourages simulation-to-C2 interoperability at the infrastructure level. In the three sections that follow, we review the HLA, the COE, and steps that can be taken to integrate or align them to support distributed systems containing both simulations and C2 systems.

HOW THE EMERGENCE OF HLA CAN HELP

Where will synthetic battlespaces come from? Developing them from scratch would certainly be cost- and time- prohibitive. However, many of the building blocks necessary to compose a synthetic battlespace already exist. Theater-level wargames are already used for battlestaff training. Theater- and mission-level simulations are used to analyze operational concepts. More detailed system- and subsystem-level simulations are used for system-specific engineering analysis.

In the past these simulations used a variety of interface techniques to exchange information with remote sources. Soon these simulations or their next generation counterparts will all exchange data via the HLA. The intent of the HLA is "...a structure that will support reuse of capabilities available in different simulations, reducing the cost and time required to create a synthetic environment for a new purpose. An individual simulation or set of simulations developed for one purpose can be applied to another application under the HLA concept of the federation: a composable set of interacting simulations."[1] C2 systems that are able to interface via HLA will be able to connect to a federation of simulations that work together to provide a synthetic battlespace. In this manner, C2 interoperability testing and training needs can be met.

THE HLA: A TECHNICAL FRAMEWORK WITH SUPPORTING PROCEDURES AND TOOLS

The HLA is a software interoperability framework evolving under the guidance of the Defense Modeling and Simulation Office (DMSO) Architecture Management Group (AMG). Figure 1 illustrates key HLA concepts. The HLA provides a specification of Application Programmer Interfaces (APIs) for run-time data interchange services,[2] pre-runtime templates and tools for reconciling data exchange details between applications,[3] and rules for proper use of these services and tools.[4] Within this framework the HLA facilitates a logical context that underpins data interactions so that participating applications know what data is expected of them and in what form data will be delivered to them. The HLA does not provide a list of approved software products, just API specifications for services that any HLA-compliant Run Time Infrastructure (RTI) must provide. The choice of hardware platform, software components, and coding language for the RTI is left to the developer's discretion.

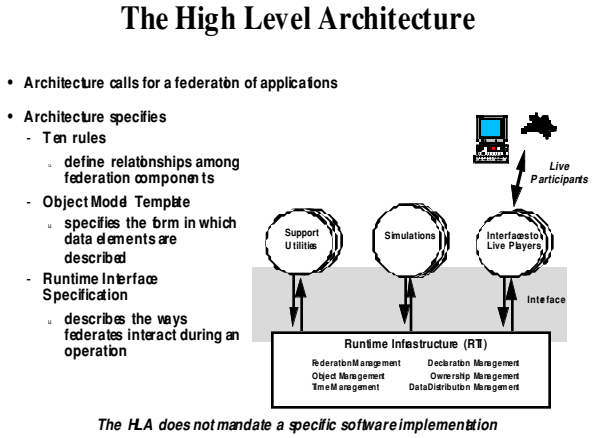

Figure 1. HLA Overview (DMSO figure)

In addition, for a group of software applications that wish to connect with one another, the Federation Execution Process (FEDEP) provides a step-by-step procedure for implementing the HLA.[5] From the point of view of a software engineer faced with enabling his/her single application to participate in an HLA federation, this process includes negotiating with engineers representing the other federates to develop a Federation Object Model (FOM) describing the data-passing needs within that particular federation. The HLA Object Model Template dictates the format for expressing this FOM. Then the software engineer establishes the necessary RTI calls and callbacks, as specified in the RTI Interface Specification, for the application to (1) provide its share of the data exchange capabilities specified in the FOM and (2) maintain proper time synchronization with the other federates.

It is important to note that an application's HLA interface built for the specific needs of one federation does not render the application capable of participating in all HLA federations. For an application to participate in a different HLA federation that has negotiated a different FOM, additional interface development work may be necessary. The simulation community is beginning to realize the value of implementing "FOM-agile" HLA interfaces that can be readily tailored for participation in new federations as the need arises.[6]

THE DII COE: A REPOSITORY OF APPROVED SYSTEM COMPONENTS

The DII COE (see Figure 2) is a collection of approved COTS or GOTS software modules or "segments" ranging from compulsory operating system/kernel software to optional infrastructure services and common support applications. The Defense Information Systems Agency (DISA) is managing the development of the DII COE. Infrastructure services include communications modules, presentation and Web tools, and distributed computing services like the Common Object Request Broker Architecture (CORBA) and the Distributed Computing Environment (DCE). Common support applications include mapping, alerts, correlation, and other capabilities. The goals of the COE are to facilitate interoperability among compliant applications and lower development and maintenance costs via product standardization.[7] Efforts to date have concentrated more on cost-related objectives, leaving most of the interoperability challenges to be tackled in later releases. By (1) following the Joint Technical Architecture (JTA) and its product standardization approach, and (2) requiring that all components be segmented so that they can be installed by a common installation tool, the COE aims to reduce the complexity of systems administration tasks in fielded systems.

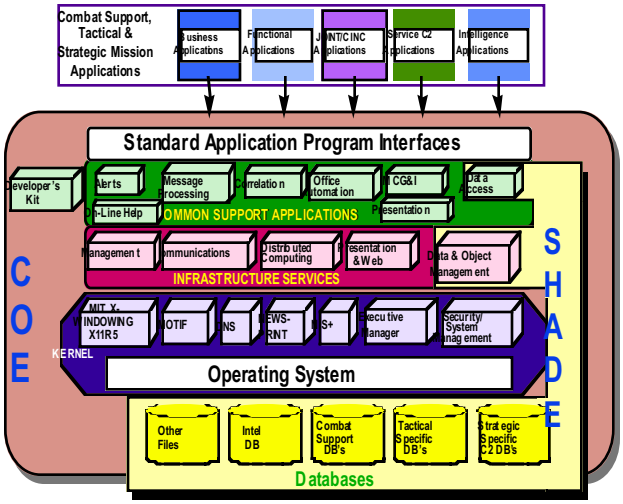

Figure 2. DII COE Taxonomy (DISA figure)

HLA/COE INTEGRATION OPPORTUNITIES

Although the COE and HLA are conceptually very different, there is no reason why they cannot be compatible, even complementary. HLA pre-runtime tools and runtime infrastructure software could be repackaged as COE-compliant segments, or even included in the COE as COE components if so desired. C2 system developers could then use these tools to design and build HLA data entry points to their systems to facilitate simulation support. In the subsections that follow we review MASC research in the areas of (1) COE-compliant HLA software and (2) approaches to HLA/COE information exchange that facilitate interoperability between simulations and C2 systems.

DISA, with active involvement from DMSO and the Navy, Army and Air Force, is promoting a study group to determine what set of simulation-related activities make sense to address within the technical and management structures of the COE. One initial activity being considered is the incorporation of a COE-compliant HLA RTI segment within the COE. A COE-compliant RTI would be especially useful in cases where simulations are employed as embedded C2 decision aids or in C2 system testing and training environments.

The MASC performed a trial COE segmentation of an HLA RTI in order to determine the level of effort required and to discover any technical issues involved with the segmentation process. We used COE Version 3.3, which was the most recent stable release of the COE at the time we performed the segmentation. At the request of the DISA study group, we chose RTI Version 1.3 Next Generation (v1.3NG) for the task. Both COE and RTI releases were available for Windows and Solaris platforms; we chose to work in the Solaris environment.

Because DISA is not supporting the popular Solaris 2.6 release, and the RTI of choice was only available for Solaris 2.6, we were required to perform a tedious set of operating system upgrade and patching steps before we could install both the COE and the RTI on our workstation. Furthermore, insufficient COE documentation hampered our effort. As the COE matures, we expect these inconveniences to lessen. Even so, we were able to segment the RTI in less than 4 weeks. Our efforts resulted in a COE-compliant RTI that can be installed on a properly patched Solaris 2.6 COE workstation via the COE Installer tool and executed from an icon within the user's Application Manager window.

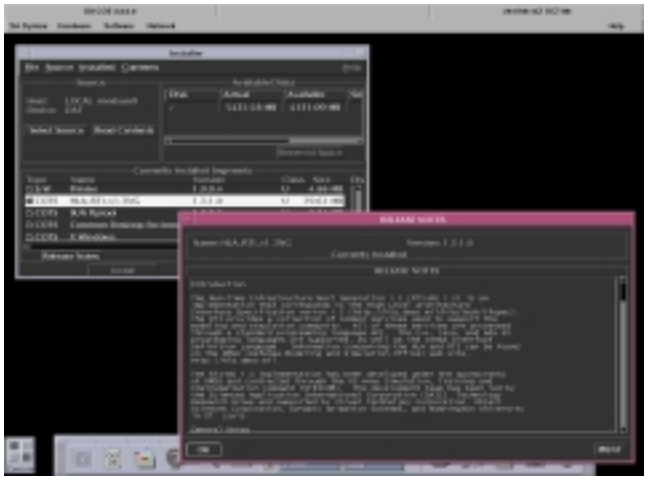

Figure 3: Installation of COE-compliant HLA RTI

Figure 3 shows the COE system administrator's console just after installation of the RTI segment. The window in the upper left-hand corner is the COE Installer software. The system administrator has highlighted the RTI from the list of installed segments, and selected the release notes, the first page of which appears in the window to the right.

A Lightweight Interface Federate Approach to Data Interoperability

The MASC is working with middleware applications that take the integration of the HLA and COE paradigms one step further. These applications, called Lightweight Interface Federates (LIFs), translate information from HLA-compliant simulations into formats or protocols contained in the COE that are commonly used for communication and interfacing within the C2 community. For example, a Track Data LIF is shown in Figure 4 that processes tracks as provided by a simulation via the HLA in the format specified in the governing FOM. The LIF in this configuration translates and repackages the data as Joint Tactical Information Distribution System (JTIDS) tracks (supported by COE Infrastructure Services) and sends them to the receiving C2 system. In theory the LIF could be modified to provide CORBA structures (also supported by COE Infrastructure Services) or database updates (supported by the Shared Data Environment (SHADE) portion of the COE) instead of JTIDS tracks by "swapping out" the C2 interface layer. On the simulation side of the LIF, different federations using different FOMs could be supported by modifying the HLA interface layer.

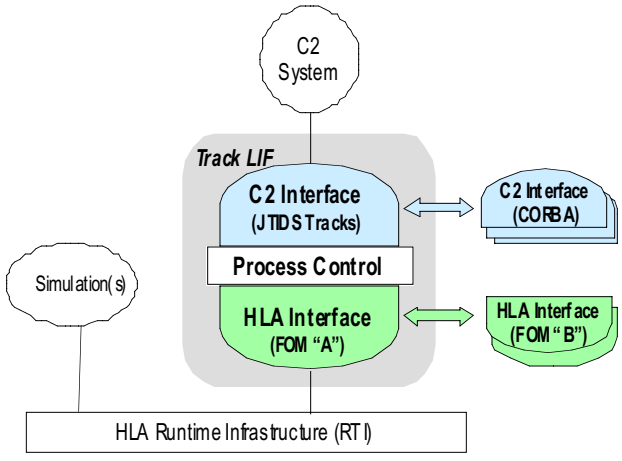

Figure 4. Lightweight Interface Federate Concept

Several LIFs would be employed to cover the variety of data exchange types between the synthetic battlespace and C2 systems. The list of LIFs might include, among others:

- a Battle Orders LIF, for transforming ATO and other battle orders from United States Message Text Format (USMTF) format into simulation-useful mission information,
- a Battle Results LIF, for transforming simulated battle results into USMTF or database updates required by C2 systems,
- a Truth Data LIF, for sending object location data for display or processing by C2 systems,
- a Sensor Data LIF, for providing simulated sensor reports for processing/fusion by C2 system tracking and display applications, and
- a Combat Operations Message LIF, for providing weapons assignments, changes in track reporting responsibility, and other C2-related decisions and events to the synthetic battlespace.

In theory these LIFs provide the HLA connections necessary for C2 systems to operate within a synthetic battlespace without requiring HLA modifications to the actual C2 software. Instead, existing operational interfaces to the C2 software are reused. Limiting each LIF to a specific data exchange type rather than covering all possible data exchange types in a single reconfigurable interface application allows for straightforward, streamlined LIF implementations that address the specific performance and maintainability needs of each data exchange type. These LIFs are central to the MASC's vision for a Simulation-Based C2 Integration Framework (SBCIF), discussed below.

A SIMULATION-BASED FRAMEWORK FOR C2 SYSTEM INTEGRATION AND TEST

Figure 5 shows the presents the SBCIF vision: a layered framework, based on simulation capabilities and tools, to support integration testing of Air Force C2 systems at ESC. The Battlespace Infrastructure layer at the bottom contains the essential components of the synthetic battlespace. Simulations that can collectively model all the military platforms and operations centers in the battle at some low- to mid-level of detail reside here. Also present are runtime support tools that monitor the state of the processes in the framework or collect experimental data for later review. The HLA RTI that ties these applications together with each other and the rest of the layered framework operates in this layer as well.

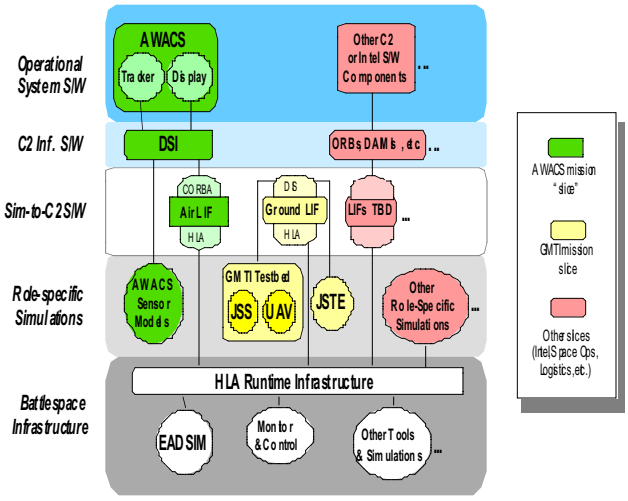

Figure 5. Simulation-Based C2 Integration Framework

The Role-specific Simulations layer contains simulations that provide more detail on specific operations within the battle of interest to support the needs of a specific test or exercise. The Simulation-to-C2 layer contains the LIFs described in the previous section above that translate data between the formats specified in the HLA FOM and COE-based C2 data exchange mechanisms. The C2 Infrastructure Software layer contains COE-based C2 infrastructure software, and the top layer contains C2 system applications.

The applications in the diagram are arranged in vertical slices to represent the MASC's efforts to populate the framework through prototyping. The project concentrated first on software related to the Airborne Warning and Control System (AWACS): that is, simulations and C2 components aimed at addressing the air warfare picture (shown at the left end of Figure 5). A prototype was developed that features the Extended Air Defense Simulation (EADSIM) executing a battle scenario and providing air target location data (truth data). In this instance the LIF employed translates the truth data into CORBA structures for passing to the AWACS real-time CORBA-based infrastructure and on to sensor models, an AWACS tracker, and an AWACS display application.

Next, efforts focused on incorporating the Joint Surveillance Target Attack Radar System (STARS) Simulation (JSS), the Joint STARS Transportable Emulator (JSTE), and the Unmanned Air Vehicle (UAV) Model in the framework. These applications are shown in yellow as the Ground Moving Target Indication (GMTI) "slice" of the diagram. These applications add ground targeting capabilities to the air picture components

already in the framework. Although these applications are shown as simulations in the diagram; they contain much operational and pseudo-operational system code and as such are useful substitutes for actual C2 components. In addition to these applications, we incorporated a DMSO-developed tool for monitoring the HLA federates within the SBCIF.

Future efforts will incorporate C2 software representing other slices of the battle (Intel, combat support, battle planning and monitoring, etc.) into the SBCIF as time and funding permit. As we incorporate applications into the framework, we will emphasize the development of experimental versions of the LIFs needed to populate the simulation-to-C2 system software layer.

LESSONS LEARNED

DISA, DMSO, and the armed services are beginning to investigate the potential utility of COE-compliant HLA support software segments to facilitate simulation-to-C2 system interoperability. We found the technical aspects of segmenting an HLA RTI fairly straightforward; our trial segmentation effort required less than 4 weeks. Much of this time was spent either (a) dealing with operating system upgrades and patching steps or (b) performing rework that could have been avoided with proper COE documentation. Therefore we expect the technical process of segmenting HLA support software to become even easier as the COE matures and some of the inconveniences we dealt with disappear.

Our initial work with EADSIM and the AWACS software showed that many aspects of the SBCIF approach are indeed viable. The design and implementation of the Truth LIF was straightforward, and early indications from test runs indicate no performance concerns with this configuration (although additional testing is needed with more complex scenarios). Interfacing to the AWACS infrastructure, the Distributed Software Infrastructure (DSI) developed by Lockheed Martin, was eased by the fact that the DSI is based on CORBA. The DSI also offers a convenient messaging service that, working hand-in-hand with HLA Time Management Services, provides a ready-made mechanism for keeping EADSIM and the AWACS applications in time synchronization with one another.

Although the data modeling required for our AWACS work was straightforward, indications are that data modeling issues will become one of the biggest challenges as more C2 applications and simulations are added to the framework. Mapping between EADSIM's capabilities and the needs of the AWACS software was easily achieved since the interface was one-way and the data exchanged consisted of a simple set of target data fields. However each additional simulation or C2 application brings with it a different set of data exchange requirements. Mapping between data available from the synthetic battlespace and data required by C2 applications will become more complex, and there may be gaps in the process that must be filled by carefully conceived assumptions and algorithms.

Another challenge will be the proper incorporation of data from the C2 applications back to the synthetic battlespace so that C2 decisions and events influence the outcome of the simulated battle. For instance, the simulation software should be able to send an attack aircraft after a target inside its simulated battle based on a weapons assignment message inbound from an AWACS console. However, many existing simulations are not designed to be influenced by outside data sources in this way. Therefore, realistic representation of certain C2 effects within the synthetic battlespace may not be possible unless enhancements can be made to the simulations involved.

REFERENCES

[1] Defense Modeling and Simulation Office, HLA Overview from Web Site *http://hla.dmso.mil*, March 1999.

[2] U.S. Department of Defense, *High Level Architecture Interface Specification Version 1.3,* Draft 9, 5 February 1998.

[3] U.S. Department of Defense, *High Level Architecture Object Model Template Version 1.3,* Draft, 5 February 1998.

[4] U.S. Department of Defense, *High Level Architecture Rules Version 1.3*, Draft 2, 5 February 1998.

[5] Defense Modeling and Simulation Office, *High Level Architecture Federation Development and Execution Process (FEDEP) Model Version 1.3*, 9 December 1998.

[6] Granowetter, Len. *Solving the FOM-Independence Problem.* MAK Technologies, Inc., 1998.

[7] Defense Information Systems Agency, *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS),* Version 3.0, July 1997.