18th ICCRTS
"C2 in Underdeveloped, Degraded and Denied Operational Environments"

Integrating CPOF, JSAF and ONESAF through CBMS

Topics
Experimentation, Metrics, and Analysis
Modeling and Simulation
Architectures, Technologies, and Tools

Names of Authors
Saikou Diallo, Ross Gore, Anthony Barraco
Virginia Modeling Analysis and Simulation Center

Amy Warren
General Dynamics Information Technology

Point of Contact
Saikou Diallo
1030 University Blvd, Sufflok, VA 232435
757-638-6320
{Sdiallo, Rgore, Abarraco} @odu.edu
Amy.Warren@gdit.com

Abstract

Web services, solutions and standards to enable interoperability are becoming more and more prevalent in both the Command and Control (C2) and the simulation world. Recently, the Simulation Interoperability Standards Organization (SISO) released the Military Scenario Definition Language as the standard for simulation initialization and is working on a Coalition Battle Management Language (C-BML) as a standard unambiguous language to support Command and Control (C2) to Simulation interoperability. In this paper, we show how to use web based solutions to support the integration of the Command Post of the Future (CPOF) Sandbox with JSAF and OneSAF. We focus specifically on the technical challenges of using multiple standard languages such as the military scenario definition language (MSDL) and C-BML simultaneously in support of initialization and execution of federations that include C2 systems.

Introduction

Interoperability between military command and control (C2) systems and simulation systems is critical for efficient planning, training, experimentation, and operational support needs. While interoperability of C2 systems is enabled by C2 standards and interoperability of simulation systems is enabled by simulation standards, the interoperability of C2 systems and simulation systems was not addressed in a coherent and standardized way until recently. The international standardization group on Coalition Battle Management Language (C-BML) is working on defining an "*unambiguous language used to command and control forces and equipment conducting military operations and to provide for situational awareness and a shared, common operational picture"* applicable for information exchange between C2 systems and simulation systems (Blais, Galvin & Hieb, 2005) However, the idea of using C2 systems to drive simulations and the emergence of Live Virtual Constructive environments (LVC) is moving operational environments into a more and more complex and interconnected domain where systems and humans must work together to accomplish a set of goals (Diallo, Tolk, Graff & Barraco, 2011).

The current state of the art in the M&S community is to use various standards ranging from the Protocol Data Units in the Distributed Interactive Simulation (DIS) (IEEE, 1998) to federation object models in the High Level architecture (IEEE, 2000) to the use of the eXtensible Markup Language (XML) based information as evaluated in various experiments conducted under the Extensible M&S Framework (XMSF) (Blais, Brutzman, Drake, Moen, Morse, & Tolk, 2004). The state of the art in the C2 community is to use structured message formats or various common reference models such as the NATO adopted Joint consultation Command and Control Information Exchange Data Model (JC3IEDM). Due to the variance between C2 and simulation interoperability standards and best practices, current interoperability solutions must be adapted to support a wide array of heterogeneous systems, languages, protocols and information exchange requirements. As a result, new interoperability solutions based on the World Wide Web and web services are emerging to support both C2 interoperability and simulation interoperability respectively. If successful, these web based frameworks will allow the integration of logistics, C4ISR and M&S systems into a cohesive whole thus providing the planner the ability to leverage a new array of capabilities.

In this paper, we present an approach for implementing C2-M&S federations using a set of interoperability web services and a set of interoperability languages such as C-BML and MSDL. We show how the proposed approach can be generalized to support system of systems interoperability. The balance of the paper is organized as follows: in Section 2 we present the motivation for C2-M&S integrated environment and present the proposed approach; In Section 3 we present the implementation of a C2-M&S federation using CPOF, OneSAF and JSAF; In Section 4 we conclude and present areas of future work.

Motivation and Proposed Approach

Commanders must have the ability to command, control, and coordinate an integrated and interoperable force in rapidly changing conditions involving complex, distributed, simultaneous, or sequential operations. Command, control, and coordination within DoD and with external mission partners requires employment of integrated and interoperable capabilities that allow assigned forces to have visibility and easy access to information to effectively organize, understand, plan, decide, direct, and monitor the execution of operations in support of a commander's intent. The Joint Training Community publishes a “Program Goals and Objectives (PG&O)” document every year to provide strategic guidance on capabilities development.). An identified PG&O requirement is “Enhance Integration with Partners” that states: sustain and improve the ability to integrate with allies, coalition members, international partners, and non-governmental organizations. Another venue to identify joint training requirements is the Training Gaps Analysis Forum (TGAF). TGAF Program Area (PA) 44 titled “Exercise Design, Integration, Standards, and Data Management” states:” revamp the management of both the federation and the simulation applications within the federation so users can plan and conduct exercises to include unilateral, coalition, and partner nations.” What this means for coalition operations is the time and costs of architecting, integrating, and testing systems of systems impedes the timely deployment of technology solutions. This cost is associated with the technical complexities of satisfying data, software, and hardware interoperability within the coalition environment. Lacking guidelines and tools for integrating heterogeneous environments, the ability to deliver new capabilities and functionalities is hindered. This will be magnified as information sharing, security, and force cohesiveness is anticipated to increase over time.

In order to fulfill the “train as we fight” objective, we often combine the advantages of live training, in which real people with real systems participate in a simulated operation, virtual training, and constructive training, in which all aspects are simulated. The means to combine live, virtual, and constructive (C2-M&S) training components are provided by architectures that support a common integration infrastructure. This infrastructure must ensure that all information needed is provided in time to the respective system (effectiveness) while at the same time ensuring that only the information needed is provided (efficiency). In addition, when addressing what information to exchange, we need to ensure that the information is structured correctly (syntax), the interpretation of the information by the participating elements is unambiguous (semantics), and the information is used as intended by the receiving elements (pragmatics). Improvements in interoperability and how integrations of systems is achieved will also reduce “time to market” for new C2 and simulation systems, better enable reuse and repurposing of solutions and improve interoperability between systems and organizations The proposed framework has the following components:

- *A set of interoperability languages*: An emerging approach to ensure the understanding of information exchanged is the definition of a *common language* or common reference

model (CRM). The Coalition Battle Management Language (C-BML) is an example for such efforts. Using agreed protocols and communication services, C-BML expressions derived from an accepted formal representation of modeled doctrine are exchanged (Schade, 2004);
- *A set of interoperability services*: The services are integrated into a web service-based infrastructure that allows message sending, receiving and manipulating (update and delete) between command and control systems and simulations or robotic forces.

In addition, we separate the C2-M&S interoperability problem space into an initialization space and an execution space. The initialization space deals with how to share information between systems: (1) pre execution including order of battle information, (2) initial tasking, control features, situational context (enemy positions, status, etc.), environmental and weather conditions. The execution space covers anything that happens after all systems are initialized and the simulation is started. This includes fragmentary orders, reports generated by the simulated entities and additional orders that tasks. Figure 1 shows a set of C2 systems and simulation systems connected through a middleware providing a set of services that allows them to exchange and use information.

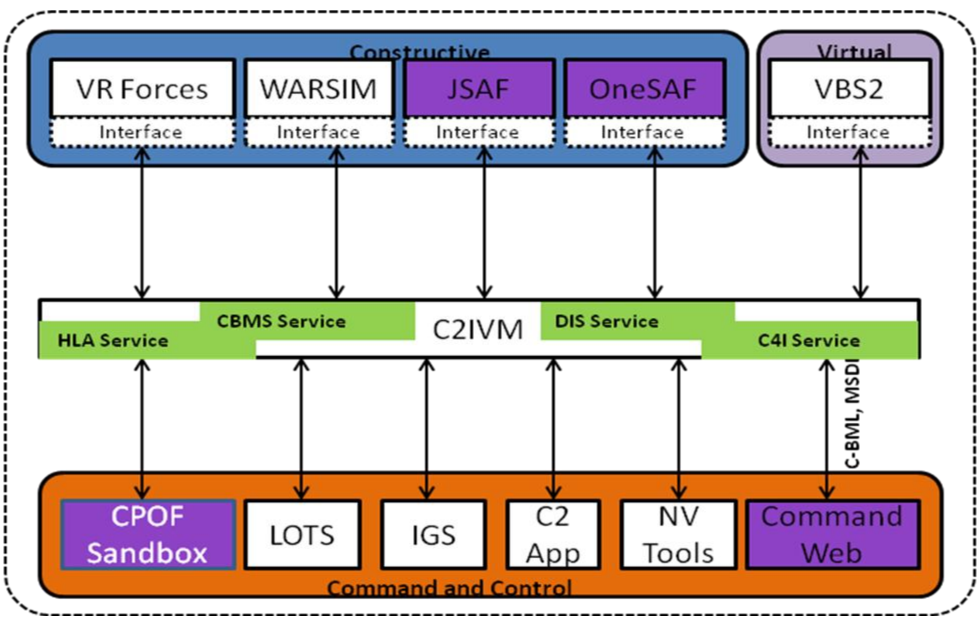

Figure 1: C2-M&S System of Systems Architecture

The architecture design will support the Military user(s) sitting anywhere in the world using an interface to generate C-BML tasks, orders, requests, and receiving reports in the context of mission rehearsal. The C-BML messages are executed by a simulation; the simulation will generate reports and requests to be fed back to the user. The user should be agnostic of the executing simulation and therefore, the C-BML messages should be complete enough to be universally executable. Similarly, the user interface should be decoupled from the simulation. In the next section, we provide a description of the architecture components required to implement this system of systems.

Architecture Components

Several web-based approaches support the exchange of information between C2 and M&S (Pullen, Corner & Singapogu, 2009). CBMS is designed specifically to support SoS interoperability and the Command and Control Infrastructure Virtual Machine (C2IVM) is designed to support operational Joint and Coalition Mission Command interoperability.

CBMS is a collection of composable web services that can be orchestrated to support the needs of a particular federation (Diallo, Wood, & Bizub, 2013). CBMS is currently implemented as a Service Oriented Architecture (SOA) with an interrupt mechanism, a filtering mechanism and a data distribution mechanism that can be used to support the validation, storage, search, and exchange of XML-based languages. These languages include, but are not limited to, C-BML and MSDL. CBMS is accessible via any commercially available web browser, and uses only next generation XML-based technologies in its implementation.

C2IVM is an interoperability solution used in the United States Army that provides an extensible set of services. We use C2IVM because it unifies all of the middleware into a single virtual machine as opposed to many distinct individual solutions and supports the exchange of tactical and operational messages between operational systems in Joint and Coalition environment. C2IVM uses a service oriented architecture that allows systems to define and use services, as well as publish and subscribe to messages. In order to further advance the interoperability between C2 systems and simulations, we added services to support the exchange of data between systems that utilize CBMS and HLA-compliant systems. The routing of data between CBMS and HLA consists of three C2IVM services: HLA Service, CBMS Service, and HLA-CBMS Mediator Service. The route is divided into separate services because the endpoints may potentially be reused if there is a future requirement to map HLA or CBMS to a new message format.

The CBMS Service acts as a client to the CBMS server. This service is capable of subscribing to XML files it is interested in receiving, as well as posting XML files to the server. This service is initialized with a configuration file that contains the server Uniform Resource Locator (URL) and subscription strings, much the same way any CBMS client would be configured. On start of this service, a connection is opened with the server. This connection is left open, allowing the server to push documents to the client as they are received.

The HLA Service acts as a federate in an HLA federation in much the same way as a simulation such as OneSAF does. Once the service has joined a federation, it initializes the handles for the classes to which it wishes to publish and subscribe to based on the federation FOM. At this point it can begin exchanging messages with other simulations in the federation through the Run-Time Infrastructure (RTI). Data is never sent directly from the service to an HLA-compliant simulation, it always communicates through an RTI.

In addition, we created an HLA to CBMS mediator in order to allow the exchange of data between the XML schemas commonly used in CBMS, such as MSDL and C-BML, and the HLA classes specified by a FOM. The routing of data through this service is bidirectional, meaning data can be routed from CBMS to HLA or from HLA to CBMS. When routing from CBMS to HLA, the Mediator Service parses the XML data it receives from the CBMS service into binding classes, which are classes generated from a XML schema. Data in binding classes are then mapped to the HLA FOM classes supported by the HLA service. The success of this step is dependent on the shared semantics of data contained in the XML schemas and the FOM. Once the mapping is complete, the data is forwarded along to the HLA service to be published to the RTI for distribution to simulations.

When routing from HLA to CBMS, the HLA service receives data from a simulation through RTI callback functions and forwards the data to the HLA-CBMS Mediator Service. Mapping functions translate that data from the FOM classes to XML binding classes. Once in the form of the binding classes, XML files are generated and routed to the CBMS service. The components of this architecture are reusable and independently developed so they can be upgraded and changed with minimal impact. In the next section, we describe an implementation of the architecture.

Application Use Case

In this section we describe a federation of CPOF sandbox (C2 system), OneSAF and JSAF (simulation systems). The systems receive initialization data from MSDL and exchange C-BML orders and reports. The use of C-BML and MSDL as supporting languages is consistent with the approach outlined in Pullen, Corner, & Wittman (2012).

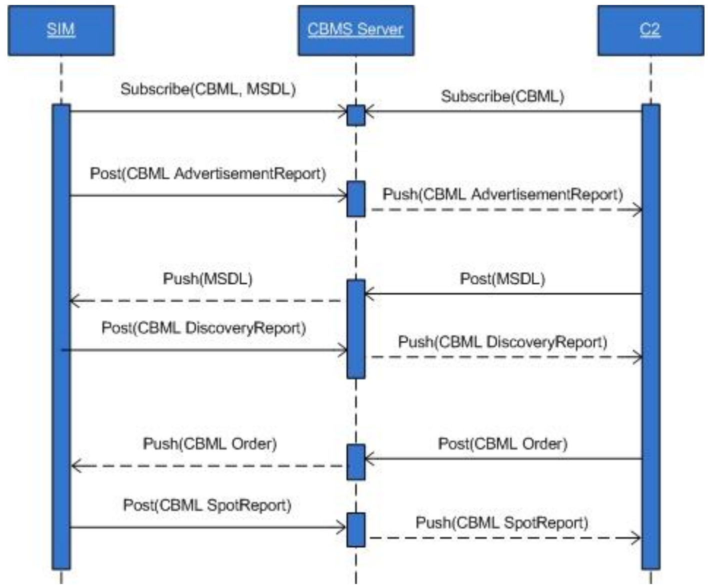

Figure 2: C2-M&S Initialization and Execution Sequence

Figure 2 shows a typical sequence in which C-BML and MSDL messages are exchanged between a simulation and C2 system using web services. The client systems start by subscribing to the type of XML files in which they are interested. When these files are posted to the server, they are pushed out to the interested client based on their subscription. To further advance the interoperability between C2 systems and legacy simulations, services were added to the C2 Infrastructure Virtual Machine (C2IVM) to support the exchange of data between systems which utilize CBMS services and HLA compliant systems such as JSAF. C2IVM uses an Enterprise Service Bus (ESB) as a messaging framework to integrate various loosely coupled services; therefore services can be added to act as a gateway which maps data between systems using CBMS and HLA. The routing of data between CBMS and HLA consists of three C2IVM services: HLA Service, CBMS Service and HLA-CBMS Mediator Service. These separate services are provided so that the endpoints can be reused for future requirements to map HLA or CBMS to a new message format. This federation was tested with a simple scenario to demonstrate the capability. In the next sections, we examine each system and show how they were integrated.

CPOF Sandbox

CPOF Sandbox is a website that interfaces with a CPOF (Command Post of the Future) server to provide a portable, remote interface to activity in CPOF. Additional plug-ins can be added to the CPOF Sandbox website simply by dropping the XAP into the CPOF Sandbox directory. The CBMS plug-in for CPOF Sandbox acts as an interface between the CBMS server and the CPOF Sandbox API. On initialization of CPOF Sandbox, the user can connect the CBMS plug-in with the CBMS subscription service. The CPOF service supports the XML schema MSDL for initialization and C-BML for tasking and reporting. CPOF Sandbox currently supports Location and Spot reports. Figure 3 shows a scenario initialized in CPOF sandbox using MSDL and the corresponding orders generated in C-BML.

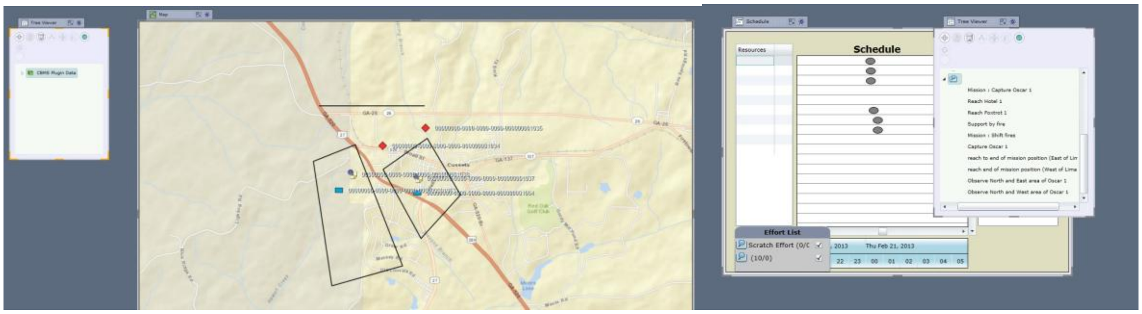

Figure 3: CPOF Sandbox Initialization and Order Generation

On receipt of an MSDL file, the CBMS plug-in uses the CPOF Sandbox API to add the specified entities and graphics, and publishes those entries in the CBMS plug-in Data tab of the TreeViewer. When the user drags the top node of the CBMS plug-in Data tab in the TreeViewer onto the map, the entities and units appear. CPOF Sandbox also supports C-BML files, but will only process orders and tasks after initialization from an MSDL file. Similarly to MSDL files, the CBMS plug-in uses the CPOF Sandbox API to add the specified orders and tasks to the CBMS plug-in Data tab of the TreeViewer.

OneSAF

The CBMS client for OneSAF was developed as a OneSAF extension. An extension to OneSAF allows for the addition of a component without having to rebuild the core classes. The CBMS client for OneSAF acts as an interface between the CBMS server and the OneSAF Application Programming Interface (API). On initialization, the CBMS client connects with the CBMS subscription service to notify it of the types of XML files it is interested in receiving. Currently, OneSAF supports the XML schema MSDL for initialization and C-BML for tasking and reporting. On receipt of an MSDL file, the CBMS client parses the data from the file and creates the specified entities and tactical graphics using the OneSAF entity creation API. MSDL file processing is supported at any time before or after the start of the simulation; multiple MSDL files may be used. The CBMS client for OneSAF does not generate any outgoing MSDL files. Along with MSDL, the CBMS client for OneSAF also supports C-BML files. Figure 4 shows CPOF Sandbox and OneSAF as initialized by through MSDL. A C-BML file will only be processed once the simulation has be initialized and started. Orders and reports are the types of C-BML files currently supported by the client.

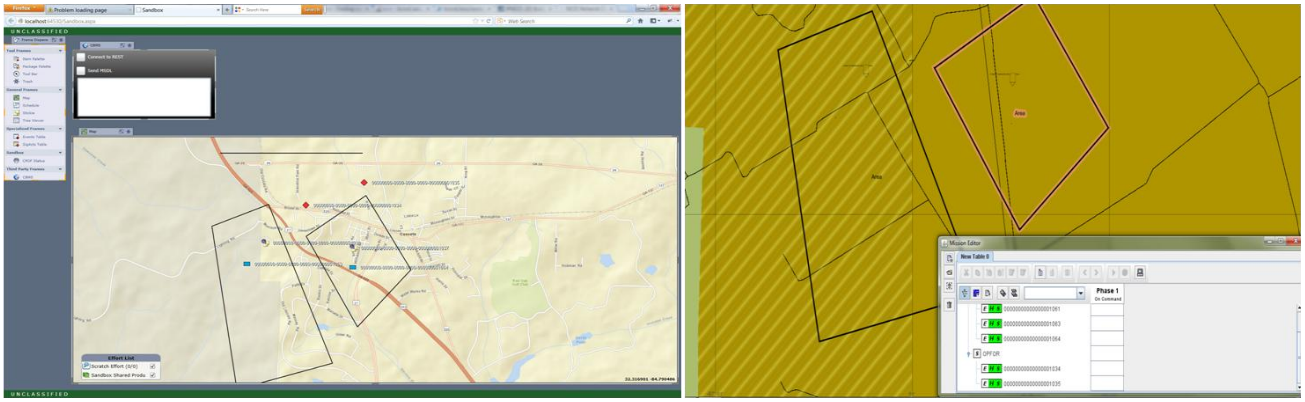

Figure 4: CPOF and OneSAF Initialization using MSDL

A C-BML order contains a list of tasks to be added as OneSAF missions. A task is only processed if the referenced entity to be tasked has already been created by a previously received MSDL file, otherwise the task is discarded. If an order has the same identifier as a previously received order, this is treated as a Fragmentary Order (FRAGO). The FRAGO will replace the old version of the order, and the entities are re-tasked. The CBMS client for OneSAF does not generate any outgoing orders.

Figure 5: Location and SPOT Reports From OneSAF in CPOF Sandbox

The CBMS client for OneSAF supports C-BML reports, which can be incoming or outgoing. At regular intervals, the client generates status and location reports for all of the entities it controls and posts those reports to the server. The OneSAF client also receives the location reports posted by other simulations. An incoming location report is parsed, and if the report references a remotely controlled entity, OneSAF will update the entity's location in the simulation. Spot reports are also posted to the server when an entity gets within range of an opposing force. Figure 5 shows a location and SPOT report generated by OneSAF and displayed in CPOF Sandbox.

Along with status, location, and spot reports, the CBMS client for OneSAF also supports a group of reports for managing and troubleshooting the simulation. These reports are defined as an extension to the C-BML schema and include advertisement, discovery, and alerts. An advertisement report contains a list of entity types that can be created in the simulation. It is posted to the server when OneSAF is initialized. A discovery report lists the valid activity codes that may be assigned to each entity and is posted in response to the receipt of an MSDL file. An alert report is a list of warnings and errors raised by the simulation, such as a fire task failure because an entity is out of range or out of ammunition.

JSAF

JSAF does not provide an API for the creation and tasking of entities so interactions are achieved through a RTI using HLA 3.1. The HLA Service added to the C2IVM service bus joins a federation along with a JSAF instance and subscribes to Platform objects. As JSAF executes tasks and updates the position of its entities as shown in the JSAF screen shot, it publishes a Platform object to the RTI which contains entity location information. C2IVM then routes this object to the HLA-CBMS Mediator Service. The HLA-CBMS Mediator Service parses the data from the HLA object and uses that data to generate a CBML location report. This report is routed to the CBMS Service which pushes the document to OneSAF. The location of the entities is updated in OneSAF to reflect the information in the report, as shown by the highlighted entities in Figure 6.

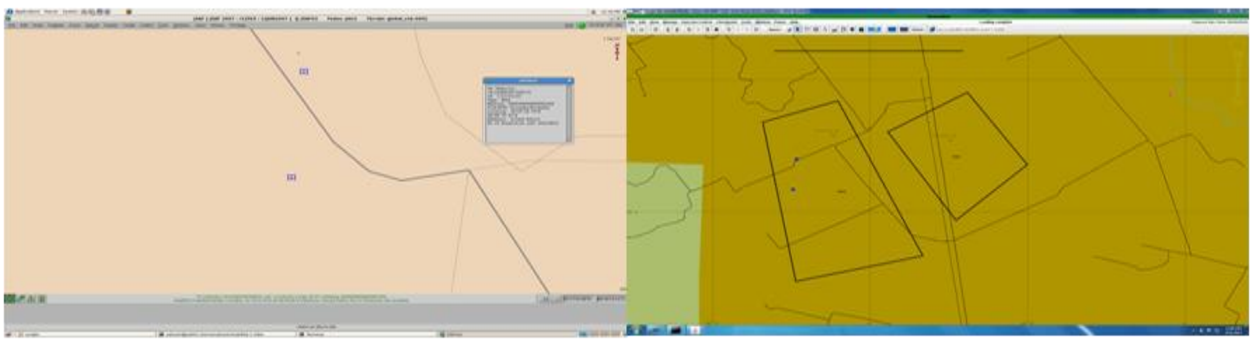

Figure 6: JSAF and OneSAF federation through C2IVM

Conclusion

The proposed architecture increases interoperability between multiple ongoing efforts through the use of common middleware (C2IVM). The services developed using this approach can be directly reused by future projects, and the architecture is flexible enough to support additional languages. As C-BML and MSDL evolve and new languages emerge, the architecture can support new services and plug-ins in the form of additional modules within C2IVM. Finally, by using the C2IVM interoperability language, it is possible to collapse all of the standard languages into a CRM that can be used as the generic language for both C2 and simulations.

References

Blais, C., K. Galvin and M. Hieb (2005). Coalition Battle Management Language (C-BML) study group report.” *Proceedings of the IEEE Fall Simulation Interoperability Workshop. IEEE CS Press.*

Blais, C., Brutzman, D., Drake, D., Moen, D., Morse, K., Pullen, J., and Tolk, A., Extensible Modeling and Simulation Framework (XMSF) (2004) *Project Summary Report, NPS-MV-05-002, Naval Postgraduate School MOVES Institute, Monterey, CA*

Diallo, S., Wood, W.D., Bizub, W. Coalition Battle Management Services (CBMS) (2012), *Proceedings of The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*

Diallo, S., Tolk, A., Graff, J., and Barraco, T. (2011). Using the levels of conceptual interoperability model and model-Based data engineering to develop a modular interoperability framework, *Proceedings of the Winter Simulation Conference, pp. 2576-2586.*

IEEE, (1998). Standard for Distributed Interactive Simulation, IEEE Std 1278-1998.

IEEE, (2000). Standard for modeling and simulation (M&S) high level architecture (HLA)– framework and rules, IEEE Std 1516-2000.

Pullen, J., Corner, D., Singapogu, S., (2009) Scripted Battle Management Language Web Service Version 1.0: Operation and Mapping Description Language, *IEEE Spring Simulation Interoperability Workshop, San Diego CA*

Pullen, J., Corner, D., Wittman, R., (2012) MSDL and C-BML Working Together for NATO MSG-085, *IEEE Spring Simulation Interoperability Workshop, Orlando, FL.*

Schade, U., (2004) Towards a higher level of interoperability: Ontology components for command and control systems, *Proceedings of the NATO R.T.O. IST-Panel Symposium on Coalition C4ISR Architectures and Information Exchange Capabilities. Den Haag*