APPROVED FOR PUBLIC RELEASE

Multi-Schema and Multi-Server Advances for C2-Simulation Interoperation in MSG-085

Dr. J. Mark Pullen and Douglas Corner
Center of Excellence in C4I
George Mason University
4400 University Drive
Fairfax, VA 22030
USA

{mpullen,dcorner}@c4i.gmu.edu

Dr. Robert Wittman
MITRE Corporation
3504 Lake Lynda Drive, Suite 410
Orlando, FL 32817
USA

rwittman@mitre.org

Adam Brook
QinetiQ, Cody Technology Park
Ively Road, Farnborough, Hampshire GU14 0LX
UK

rabrook@QinetiQ.com

Dr. Per M Gustavsson
Saab AB
Security and Defence Solutions
Skövde
SWEDEN

per.m.gustavsson@saabgroup.com

Dr. Ulrich Schade and Thomas Remmersmann
Fraunhofer-FKIE
Fraunhofer Straße 20
53340 Wachtberg-Werthhoven
GERMANY

{ulrich.schade,thomas.remmersmann}@fkie.fraunhofer.de

ABSTRACT

The conference theme, M&S Support to Transitioning Forces, speaks to the need to grow better capabilities for the future while sustaining the lessons learned from past deployments. One such better capability now growing under the auspices of the NMSG is MSG-085 "Standardization for C2-Simulation Interoperation" which is leading the way to new capability where coalition partners can simply "plug in" their command and control (C2) and simulation systems on a common network and thereby exchange the information needed for simulation-based training, mission rehearsal, and course of action analysis. Using such a capability, every group would "train as you fight," using their own national C2 system and would be represented by their own simulation.

This paper reports on significant steps forward in C2-simulation interoperation that have been developed and implemented by the authors in support of MSG-085: a coalition of simulation, C2, and web-enabled infrastructure systems that are capable of interoperating over a variety of syntactically different schemata, to the extent they are semantically compatible. Further, the system provides support for distributed, interoperating servers. This system enables interoperation among coalition partners who may be working with simulation and C2 systems developed with different schemata generations or versions. It also supports geographic and organizational scalability by enabling support from distributed servers.

The key to multi-schema operations lies in the ability to parse every BML document (including C-BML, MSDL, and related schemata) and map it into a common data model. Doing this in a usable way requires a server that is capable of parsing XML documents and republishing them at high rates (at least one hundred per second) to service both C2 and simulation clients. The existing open source Scripted BML (SBML) Server has the needed translating and C-BML/MSDL capabilities, but not the required performance. Therefore the SBML server has been re-implemented using Saab's WISE infrastructure, which not only provides high performance but also features a graphical editor that enables configuring a new schema even more rapidly than the previous scripting approach. This design has been extended and refined, working with client systems developed by other authors of this paper. The process includes appropriate simulation and C2 system initialization and information-based interactions to support necessary user and C2-simulation interactions.

The key to multi-server operations is a server-to-server protocol, used with a schema that is common to both servers. Such a protocol has been implemented between the existing Fraunhofer-FKIE server and the new WISE-SBML server. The approach could be generalized, so long as each pair of servers implements some schema that supports the required semantics.

The paper concludes with a summary of new achievements in C2-simulation interoperability conceived and implemented by the authors in support of MSG-085, and a projection of future developments that can reach the ultimate goal of "plug and play" C2-simulation interoperability among coalition partners.

1.0 INTRODUCTION

Battle Management Language (BML) and its various planned and proposed extensions are intended to facilitate interoperation among command and control (C2) and simulation systems or "C2-Sim" by providing a common, agreed-to format for the exchange of information such as orders and reports. In recent implementation, this capability has been accomplished through a repository service that the participating systems can use to post and retrieve messages expressed in BML. The service is implemented as middleware that is essential to the operation of BML and can be either centralized or distributed. Recent implementations have focused on use of Extensible Markup Language (XML) along with Web service (WS) technology, a choice that is consistent with the Network Centric Operations strategy adopted by NATO [1].

Continuing progress of a Coalition Battle Management Language (C-BML) and Military Scenarios Definition Language (MSDL), both in NATO MSG-085 and in the standards process of the Simulation Interoperability Standards Organization (SISO), has produced a situation where various clients have implemented different generations of schemata, which, while semantically equivalent or nearly so, cannot interoperate directly because of schema mismatch. Retrofitting these clients would be a drain on scarce resources. In our work to implement a robust C-BML/MSDL server using the WISE platform, we have incorporated the ability, previously demonstrated in the SBMLServer, to translate among semantically equivalent schemata. The resulting capability is planned for use by NATO MSG-085 to enable interoperation of all participating groups in the MSG-085 final demonstration. The same continued progress in C-BML within NATO MSG-085 *Standards for Command and Control-Simulation Interoperation* has resulted in a need to extend the scope possible by supporting C2-Sim with a single server. The extension could be either for purposes of geographic distribution or for extended performance. In preparing for the MSG-085 final demonstration, we have developed an example of such a distributed server, achieved by interoperation between the WISE-SBML server and the FKIE BML server.

Section 2 of this paper provides a synopsis of the organizational context, NATO MSG-085 and SISO MSDL/C-BML, which provided a stimulus for the work. Section 3 explains how the schema translation works and issues encountered in its implementation, which includes an MSDL aggregation server. The discussion includes the history of SBMLServer, a description of WISE, a discussion of how WISE is used to enable an operationally-focused server supporting schema translation (WISE-SBML), and a description of how MSDL also is supported in this environment. Section 4 describes server interoperation of WISE-SBML and the FKIE server, the mechanisms used, and issues encountered in implementation, and the resulting capability for distributed operation combined with schema translation.

2.0 ORGANIZATIONAL CONTEXT: NATO MSG and SISO

The work described below was undertaken by participants in NATO Modelling and Simulation Technical Activities and has included developing and validating SISO standards. We summarize this important context here; further details can be found in references [10] and [20].

2.1 NATO MSG-085

The NATO Modelling and Simulation Group (MSG) charters technical activities conducted by groups from nations that are members or NATO or Partners for Peace to improve understanding and utility of technology involving modelling and simulation. Technical feasibility of coalition BML was demonstrated by NATO MSG-048 in a Technical Activity conducted 2006-2009. References [6 - 13] describe the C2-Sim environment developed for NATO Technical Activity MSG-048, *Coalition Battle Management Language*. This activity included six national C2 systems, five national simulations, and two supporting software systems, a scale of interoperation not previously attempted. Reference [18] describes developmental work for NATO Technical Activity MSG-085, *Standardization for C2-Simulation Interoperation,* leading to an experimental operational environment where multiple national C2 and Simulation systems can interoperate using MSDL and C-BML (see Figure 1).

The follow-on Technical Activity, MSG-085, is chartered to demonstrate and facilitate the operational utility of MSDL and C-BML in military coalitions, and also to assess the operational relevance of MSDL and C-BML while enhancing the technical readiness level of their available implementations. MSG-085 had its initial organizing meetings in 2010, resulting in an Operational Subgroup (OSG) that is defining validation experiments and a Technical Subgroup (TSG) that is assembling required C2 and simulation systems and necessary infrastructure. The authors are participants in the TSG and have participated several demonstrations of that infrastructure.

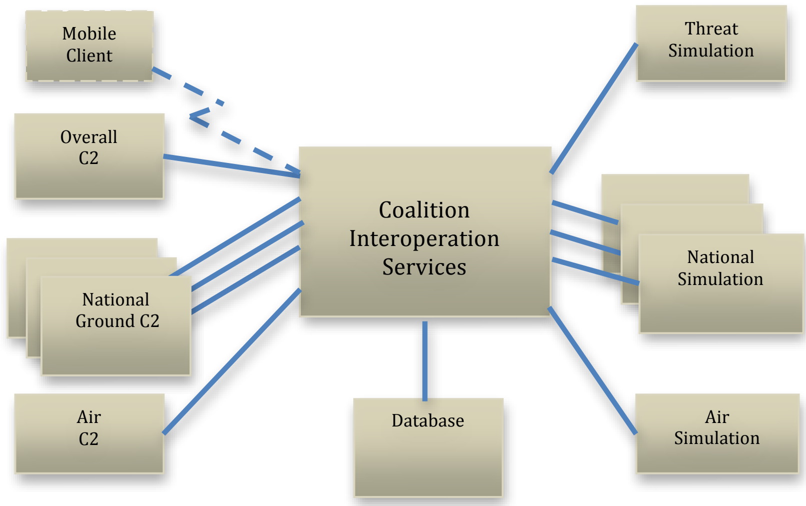

Figure 1: C2-Sim Coalition Services Architecture.

2.2 SISO MSDL and C-BML

SISO has a two-part standards effort supporting BML. The Military Scenario Definition Language (MSDL) standard [2] was approved in 2008. It is intended to reduce scenario development time and cost by enabling creation of a separable simulation independent military scenario format, focusing on real-world military scenario aspects, using the industry standard data model definition eXtensible Markup Language (XML) as input to initialize C2 and simulation systems. The Coalition BML (C-BML) standard [3] provides the tasking and reporting aspects of C2-Sim. It was balloted in 2012 and is expected to be approved as a SISO standard in 2013. Several recent studies and implementations have addressed effective combination of MSDL and C-BML [6-9]. Informing the standardization process have been multiple projects under various US DoD sponsors and an ongoing sequence of experimental BML configurations that were developed and demonstrated by the members of NATO MSG-048 and MSG-085 [14-17]. The experience gained in these activities has proved critical to shaping the MSDL and C-BML standards and implementing infrastructure, such as the translation service described in this paper, and also in demonstrating the potential applicability of BML.

MSDL grew out of a need within the OneSAF Program to reduce scenario development time and cost, with the additional goal of being able to use the resulting scenario across multiple simulations, running within a federated environment or as independent simulations. The original concept was to create a separable, simulation-independent military scenario format, focusing on real-world military scenario aspects, using the industry standard data model definition eXtensible Markup Language (XML). Such a scenario could easily and dependably be consumed by current and evolving simulations. The concept was prototyped during the early development of OneSAF. A subsequent SISO study group identified a community-wide need for a standardized military scenario format in order to reduce development time and cost and to enable sharing of re-usable scenario products. The result was seen as a way to automate the largely manual reproduction of a scenario into multiple simulation scenario formats and reduce the number of errors introduced during manual scenario development.

A formal SISO MSDL standard Product Development Group (PDG) was established in 2006 to produce a standard Military Scenario Definition Language data model. The PDG built on the MSDL specification provided by the OneSAF program and then refined its data model by aligning it with the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM), adding some elements such as weather information, and a scenario identification section leveraging the Base Object Model Identification schema. They postponed work on a Course of Action structure that was equivalent to the work being pursued under the SISO C-BML PDG. A formal SISO Standard Version 1.0 MSDL was approved in 2008 and has seen additional investment and use by the US Army Modeling and Simulation Office (AMSO), Air Force, Marine Corps as well as MSG-085 activities involving several national development groups.

SISO also created a study group to develop a Coalition BML (C-BML) standard in 2005 [3] and a corresponding PDG was chartered in 2007. The C-BML Phase 1 Draft Standard reached the point of Trial Use in 2011 and is was balloted in 2012. The approach has generally followed the Lexical Grammar approach introduced by Schade and Hieb [4,5]. Informing the standardization process have been multiple projects under various US DoD sponsors and an ongoing sequence of experimental BML configurations developed and demonstrated by the members of NATO MSG-048 and MSG-085 as described in the references.

3.0 TECHNOLOGY CONTEXT: PREVIOUS SERVER SYSTEMS

The work reported here builds on a rich legacy of effort in the context of MSG-048, MSG-085 and SISO C-BML. This section describes those technical components assembled in that context that have provided a basis for the advances reported in section 4.

3.1 Scripted BML Server

The George Mason University C4I Center, under management of US Army PM OneSAF and in close cooperation with MITRE and QinetiQ personnel, has developed a set of services that provide infrastructure to support implementation of MSDL/C-BML in MSG-085 C2 and simulation systems. The top-level architecture of a C2-simulation coalition using these services is shown in Figure 1. These implementations are available at http://c4i.gmu.edu/OpenBML as open source software.

Experience to date in development of BML indicates that the language will continue to grow and change. This is likely to be true of both the BML itself and of the underlying database representation used to implement the scripted server capability. However, it also has become clear that some aspects of BML middleware are likely to remain the same for a considerable time: namely, the XML input structure and the need for a repository server to store a representation of BML in a well-structured relational database, accessed via the Structured Query Language (SQL). This implies an opportunity for a re-usable system component: a scripted server that can convert between a relational database and XML documents based on a set of mapping files and XML Schema files. The scripted server introduced in [19] and now named "SBMLServer," accepts *push* and *pull* transactions (BML/MSDL XML documents) and processes them according to a script (or mapping file, also written in XML). While the scripted approach may have lower performance when compared to hard-coded implementations, it has several advantages:

- new BML constructs can be implemented and tested rapidly
- changes to the data model that underlies the database can be implemented and tested rapidly

- the ability to change the service rapidly reduces cost and facilitates prototyping
- the script provides a concise definition of BML-to-data model mappings that facilitates review and interchange needed for collaboration and standardization

Since its initial use in NATO MSG-048 [18], SBMLServer has been enhanced considerably by:

- Supporting XML namespaces: XML tagnames can be qualified by addition of a “namespace” prefix. This allows tagnames from different sources to work together safely.
- Schema validation: the server confirms that each document received conforms to the schema, which identifies a likely source of incompatibilities, at the cost of slowing the service
- Filtering data to restrict delivery based on user-defined criteria: SBML supports dynamic definition of Publish/Subscribe Topics
- Logging/replay: the server writes a file showing every transaction it receives, with time stamps. The server is then capable of replaying this file to recreate the original sequence of Orders and Reports at original time intervals.
- Multithreading: server throughput can be improved by processing multiple messages in parallel.
- RESTful Web service interface: initially, SBMLServer supported only the traditional Web service protocol SOAP, which is intended to support remote procedure calls. However, the need in BML is for transfer of documents, which can be achieved more efficiently via Representational State Transfer (REST) protocol, reducing overhead and facilitating C++ implementation. MSG-085 has adopted the RESTful approach.
- Aggregating MSDL inputs: see section 5 below.
- Schema translation: because SBMLServer parses the BML input documents and stores their XML elements in a database, it is possible to generate a version of the document translated to comply with a different, semantically similar schema.

3.2 FKIE BML Server

The FKIE Server was developed as an alternative to the SBMLServer with the focus on allowing a simple, fast exchange of BML while keeping flexibility to change the schema. The FKIE Server uses the same web interfaces as the SBML Server to be able to exchange the servers while using the same clients. Incoming messages are distributed over the messaging service and are stored in the file system for later requests. The XML of the messages is neither changed nor validated by the server. However, for some messages types the server does a search for predefined strings to extract order, report or request IDs. This makes the server fast and allows changing the schema from one experiment to another without also changing the server. This allowed fast development of new schemata and schema extensions.

Currently, the FKIE Server supports IBML, SISO Phase 1 and the CIG Land Ops schema [15]. Since there is no converting done by the server, all clients must agree upon one of the mentioned schemata at the start of a session or experiment. In addition to BML message exchange, the FKIE server supports MSDL for scenario initialization. Messaging services JMS and STOMP currently are supported, while SOAP and a RESTful interface are offered for Web services.

For the CIG Land Ops experiment, the SOAP and the JMS interface of the FKIE Server were used together with the CIG Land Ops schema. For the experiment, we used the following systems C2-Systems: SITAWARE

(Denmark), SIR (FRANCE), TALOS (SPAIN) and C2LG GUI (Surrogate, Germany) and simulation systems: SWORD (France/Denmark), APLET (France) and VR-Forces (Netherlands/Spain). More Information can be found in [23].

3.3 Ground Operations: OneSAF

The USA Army Modeling and Simulation Office funded a MITRE effort to integrate MSDL and C-BML data models into a working OneSAF solution. OneSAF is an entity-level simulation developed by the Program Executive Office for Simulation Training and Instrumentation (PEO STRI) and is used across the US Army for analysis, experimentation, testing, and training. It is under active evolutionary DoD and government open-source development (available under USA Foreign Military Sales) and is delivered as a simulation toolkit that can be tailored by end-users for their specific purposes.

To support an integrated MSDL and C-BML OneSAF capability, a number of enhancements have been included within OneSAF version 5.5. These enhancements provide an implementation that fully complies with the MSDL standard, while allowing for local extensions, and also support the C-BML draft standard now in balloting. The effort also provided the ability for OneSAF to import and export a limited set of the Full and Light data elements associated with the C-BML standard schema. Enhancements included:

- Improved MSDL document validation and 2525B symbol code use for unit/platform type and echelon;
- Ability to persist and reference entity mapped unit/platform in subsequent MSDL imports;
- Capability to import and export C-BML orders (move, attack, etc.) using both Full and Light elements;
- Capability to connect to the web-based coalition-monitor tool described in [21];
- Capability to send and receive MSDL and C-BML documents to/from BML servers;
- Capability to reference one or more C-BML files within an MSDL file for use during OneSAF initialization to populate the OneSAF mission editor; and
- Capability to cross-reference units and equipment tasking between MSDL and C-BML documents using unique identifiers.

The ability to identify and reference files external to the MSDL instance document without modification to the MSDL standard is one of the most significant design contributions provided by this effort. This design, which is further described in [17], allows for one or more C-BML files to be used during initialization time and also allows for other types of information to be referenced and used by a simulation importing an MSDL file. These referenced files may contain additional detail and/or simulation specific information not included within the MSDL file. Figure 2 provides a graphical depiction of the MSDL reference design. With the implemented reference design, the files containing additional data can be defined and created using XML or other data modeling technologies and referenced within the MSDL file. The importing simulation can then use the referenced data to augment the information in an MSDL file.

The reference design pattern has the additional benefit of allowing local extension files to be independently defined, used, matured, and shared between simulation and Mission Command federations when appropriate. Over time, as these referenced files and associated data models are shared and gain community acceptance, they can be proposed for integration into the core MSDL standard. The standardization process itself is quite lengthy and with community inputs these data models are likely to change as they are integrated into the standard. Although the changes are a necessary and valuable part of the process they mean software and import level changes to existing users and producers of data. The advantage of the "reference" design pattern is that it allows users to manage when and how to step up to the new MSDL standard. Users can either step up to the new MSDL schema and associated data elements, or continue to reference the files within the new MSDL standard schema, or use a legacy MSDL schema with the legacy reference files.

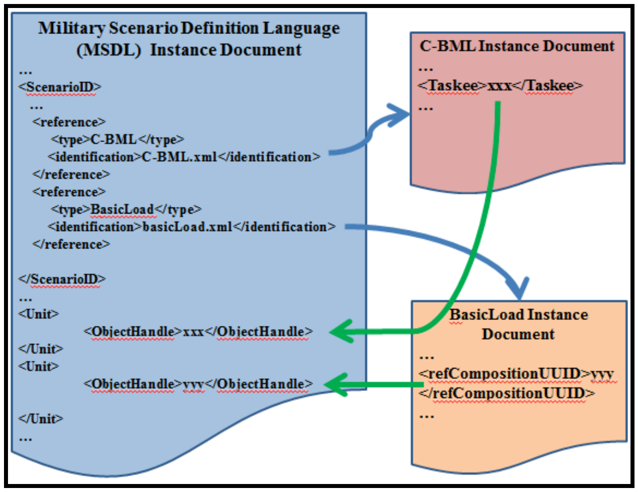

Figure 2. MSDL Reference Design Pattern

3.4 Air Operations: ICC, JADOCS, and JSAF

Existing capabilities for air operations have been centred on the use of the NATO Integrated Command and Control (ICC) tool and the Joint Automated Deep Operations Coordination System (JADOCS), both operational C2 systems and the Joint Semi-Automated Force (JSAF) for simulation.

ICC is used to generate Airspace Control Orders and Air Tasking Orders (ACOs and ATOs). These provide the operational framework for an air component commander. The ACO defines the operational areas, air routes, navigation and timing information, etc. The ATO defines the aircraft allocation and their missions, including communication, load, refuelling and coordination, referring to the ACO. ACOs and ATOs are seldom modified during the course of an exercise serial. JADOCS is used in conjunction with ICC to coordinate, de-conflict and prioritise targeting across domains. JADOCS may also be used to issue targeting instructions, e.g. for Close Air Support (CAS). Both ICC and JADOCS may be used to integrate C4ISR information and display a Common Operational Picture (COP) through the receipt of a variety standard military message formats, e.g. OTH-Gold and NFFI.

JSAF is a constructive simulation and has been used here for the simulation of air operations. Through the use of a translator system, JSAF is able to execute ACOs and ATO missions and generate reports expressed in C-BML.

Experience gained from earlier experimentation has led to the development under Canadian auspices of a revised architectural design to support the use of these and other applications. The design provides modules capable of being adapted for use with a number of different systems.

Figure 3 shows an overview of the Air Operations technical architecture, there are system interfaces to ICC, JADOCS and JSAF. For other purposes a number of other specific interfaces have been or are being developed, including for the UK BCIP army C2 system and an XMPP chat interface. The modular middleware interface has permitted the system to be upgraded easily from using the old SBML JMS-based system to the new STOMP-based WISE-SBML system. The C-BML middleware permits MSDL and C-BML to be shared amongst other C2 and simulation domains, e.g. for joint mission planning purposes and to permit joint COPs to be generated. Further detailed information relating to the use of this generic C-BML architecture is in [24].

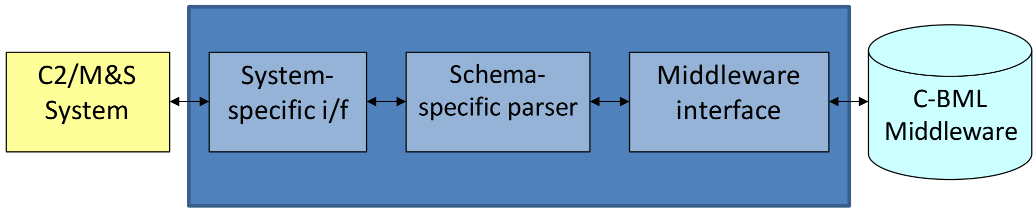

Figure 3: Generic C-BML Architecture Used with Air Operations.

MSDL is used in the air operations applications for initialisation purposes. Currently MSDL is used to initialise and to record JSAF scenario states for overlays, tactical graphics and unit and equipment or entity definition, i.e. JSAF may be initialised using MSDL or it may be used as an MSDL generator. Work continues with the use of MSDL and the C2 systems, e.g. it has been used to initialise and record JADOCS Friendly Order of Battle data tables.

Experience with the use of MSDL has shown that where it is being shared across a number of applications, for example after being merged or combined from a number of different national sources, that the elements described in the data are also apportioned to client applications correctly, this is particularly important for units and equipment. In general, distributed simulations will only require information about the units and equipment they are each simulating – the public or private simulation protocol will complete the picture. However, this is not necessarily the case for C2 systems since the operator may need information relating to force elements beyond his own immediate control, e.g. orders of battle for coalition units. This merging function is described in section 4.2, below. Currently data apportioning is considered a client function and should be defined by reference to pre-defined system design agreements. In the future, however, this too may be developed as a web service.

The systems used for air operations were developed to use the Integrated BML (IBML09) prototype schema originally used by NMSG-048 for their experimentation in 2009. For a number of reasons: simplicity and adequacy, other development priorities; this schema is still in use. The new air operations architecture design has however also been used by Canada with both “light” and “full” variants of the SISO Phase 1 C-BML schema. At the time of writing only the IBML09 schema has been used with the WISE-SBML system.

Specific expressions for air operations will be included in SISO C-BML Phases 2 & 3 at which point the requirement to use IBML09 will have been superseded. It is expected that these developments will include air operations-specific data forms and initialisation structures. Until that time, the use of translating BML web services is seen as a necessary expedient.

3.5 The WISE Integration Environment

Saab Corporation is in the business of providing software for military command and control. They have been active in the Swedish delegation to NATO MSG-085 and have offered use of their Widely Integrated Systems Environment (WISE) for experimentation support. In 2012, discussions between the GMU C4I Center and Saab concluded that the general approach used in SBML could be productively re-implemented in WISE. WISE supports a robust, high-performance information switching capability with a graphic setup editor that provides and improves upon the advantages associated with the scripted approach of SBMLserver. This capability enables fundamental research at GMU, prototyping a new generation server that is expected ultimately to transition to operational military use as described in [22].

Saab also is providing to MSG-085, through GMU, access to its 9LandBMS command and control system. 9LandBMS is a battalion/brigade level ground C2 system that is intended for use with touch-sensitive tablet computers. It had an existing interface to WISE; thus it is now capable of interoperation with C-BML capable systems using the WISE-SBML server, without going through the usual process of building a C-BML interface.

Figure 4 shows the architecture of the WISE-SBML server. The “BMS” system in Figure 4 represents the 9LandBMS or other interfaced C2 system. WISE appears to SBML as an in-memory, non-persistent database. This approach enables a great improvement in performance over the existing SBMLServer (improvement measured at over 10X in early experiments) and is suitable for deployment in the high-performance cloud computing environment.

Integrating a new capability into WISE requires creating a software interface element and then using the WISE graphic editor to configure information mappings between that interface and the WISE internal database. These configuration elements must be maintained as changes to the schema occur. It is noteworthy that the second step in particular can be achieved more quickly than developing an SBML Version 2 script. It also is noteworthy that the WISE architecture is well suited to operation in a cloud.

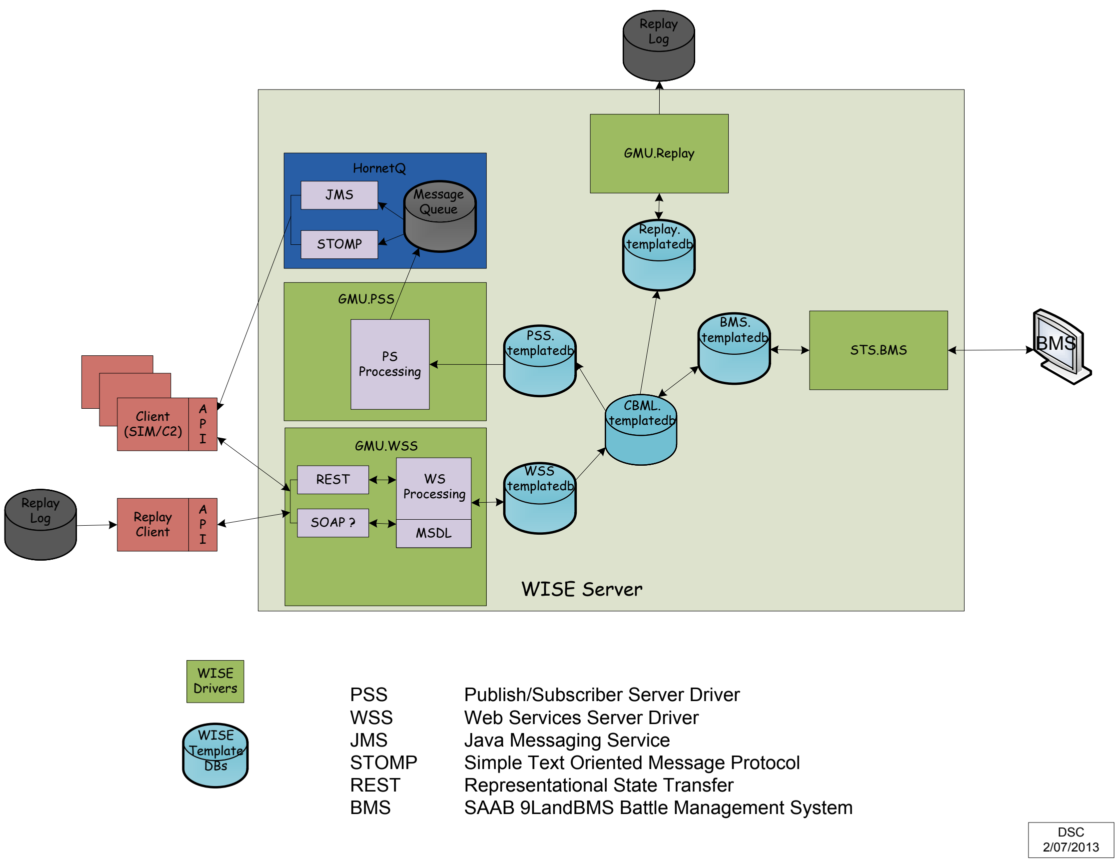

Figure 4. WISE/SBML Architecture.

4.0 MULTI-SCHEMA AND MULTI-SERVER SYSTEMS

The advances described by this paper fall in two general areas: the ability to support practical levels of message flow in a production-oriented server while translating among BML schemata and the ability to interoperate linked server systems for distributed operation. Specifics are provided in this section.

4.1 Rebuilding SBML on the WISE Platform

As shown in figure 4, the WISE-based Web service accepts XML inputs through a REST interface and publishes one or more XML documents (the original plus translations) through a STOMP interface.

Therefore, to build a server based on WISE, the GMU team had to complete two important steps:

- Build a WISE driver, shown in green on the figure, for each major information flow to be interfaced: C-BML/MSDL Web service (one for each schema version); publish-subscribe service; persistent recording interface; and the 9LandBMS WISE interface, adapted for C-BML/MSDL.
- Use the WISE graphic editor to specify all information flows between the WISE data repository and these drivers.

The second of these steps represents an added power from the WISE approach and requires only drag-and-drop in the WISE Connectivity Designer. However, there remains a sequence of steps to be programmed in conjunction with the specifics of the application (in the case at hand, C-BML and MSDL) for XML documents, both incoming from the REST interface and outgoing to the STOMP interface for publication. When the document is published to a client that is subscribed to the topic using the schema under which the document was written, it is simply retransmitted. However, when the client is subscribed to a topic using a different schema, a translation must occur. This takes place as follows:

Accepting incoming XML from REST:

Parsing: using the open source DOM parser, the interface extracts each data element from the input XML file to an internal data structure.

Building: the internal data structure is pushed into the WISE database.

Producing outgoing XML through STOMP:

Receiving: a matching internal data structure is extracted from the database.

Generating: XML output is generated from this data, in accordance with the appropriate schema.

The WISE driver software generated for this purpose, written in C++, is available as open source at http://c4i.gmu.edu/OpenBML. Also available are the client software, replay logger, and replay client.

WISE-SBML preserves all features of SBMLServer with the exception of the obsolete SOAP interface. It does not feature a persistent database; however, the trace of all inputs is captured in the replay log and can be used to restore the internal database to any desired checkpoint for restart. Reference [22] provides more information on WISE-SBML. It has received extensive testing, using the clients described in section 3 above.

4.2 MDSL Implementation in WISE-SBML

Reference [17] identifies a path to combined use of MSDL and C-BML, where:

- A common set of unit identifiers is used in applying both standards
- Standards-based XML documents are cross-coupled by references: the MSDL Scenario File references the C-BML tasking document, while the C-BML Orders and Reports refer to the Scenario File
- All of the XML documents (Scenario, Order, Report, etc.) use the MILSTD 2525C or NATO APP6-C tactical graphics identifiers as established in the Scenario File.

When multiple systems participate in a coalition, it is necessary to merge their MSDL files. Some parts of the merge process consist simply of concatenation, but other parts require functions such as the largest of a group or the total count. The MSDL scenario is the element that binds together the components to be used for a particular exercise. Once the scenario has been initialized and the signal given by the master controller, participating organizations may add additional components to the scenario. These include:

- Geographic Region of Interest
- Force/Sides
- Units
- Equipment
- Installations
- Overlays
- Graphics

With a simple addition to SBMLServer, it became possible to implement in CSL scripts the required logic for MSDL aggregation. The various clients push their MSDL documents into the SBMLServer and the XML structure is validated during this process. At any time, any client can pull an aggregated MSDL document, as assembled up to that time, for the whole coalition. Upon signal from the master controller, via the Status Monitor and Control service described in [21], the SBMLServer publishes the aggregated MSDL document to all participating C2 and simulation systems. Information from the aggregated MDSL file also is used to initialize the units and control features in the SBMLServer database. If the MSDL documents of the client systems are extracted automatically, this assures that all participating systems have available globally correct initial information. Transactions are validated as they are received to insure correct format, unique unit and equipment names and object handles, and valid references between components.

Once all organizations have submitted their data and signalled their status to the master controller, the master controller will submit a publish transaction for the scenario being used. This will cause the transmission of the full MSDL XML data to all subscribers to the MSDL Topic. Clients not using the publish/subscribe service alternatively can execute a query and retrieve the same information. The same query also may be used by organizations joining the exercise after the MSDL data has been published, in order to obtain initialization information.

All of the elements submitted by clients under a single scenario are aggregated into a single MSDL document by the server. It is assumed that clients have submitted complete components: Units, Equipment Items, Installations, Overlays and Graphics. The aggregated MSDL document will then consist of the data entered during initialization and the complete components entered by the individual transactions submitted by the clients. New units and equipment may be discovered after the exercise has started (these generally will be enemy units or equipment). In this case an update will be published on the MSDL topic detailing the newly discovered unit or equipment item. An overview of MSDL aggregation is shown in Figure 5. This approach has been re-implemented in WISE-SBML, using the WISE database to hold the Scenario File elements as they become available.

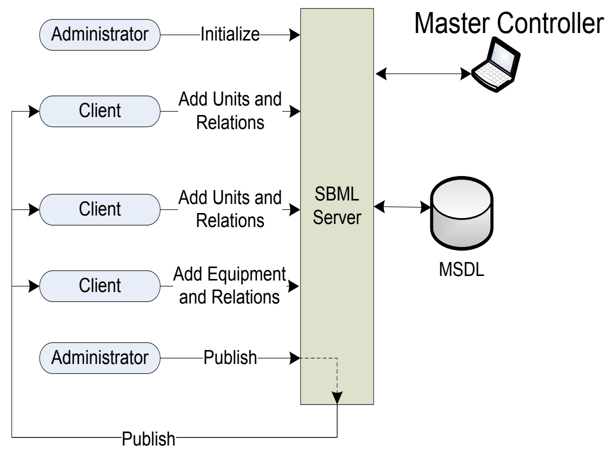

Figure 5. MSDL Server Operation for C2-Sim Clients.

4.3 Linked Interoperating Servers for MSG-085

Technical activity MSG-085 is nearing the end of its chartered mission to assess the operational relevance of MSDL and C-BML while enhancing the technical readiness level of their available implementations. MSG-085 is planning its final demonstration in December 2013, involving eleven C2 and simulation systems operated by eight different nations.

The scope of the planned demonstration is such that MSG-085 plans to combine the FKIE server described in section 3.2 above with the WISE-SBML server described in sections 4.1 and 4.2. The FKIE server will support a coalition of systems previously demonstrated by France and Germany [15], operating under the Integrated BML (IBML) schema as used in 2009 by MSG-048 and enhanced for logistics, which we will refer to here as the CIG Land Ops schema. The WISE-SBML server will support the remaining national C2 and simulation systems, and will translate among four schemata to do so: the two variants defined by the finalized C-BML Phase 1 standard, the original IBML schema, and the CIG Land Ops schema (for details on schema translation see [20]). The general architecture of the linked servers is shown in figure 6.

We have found a straightforward way to link the two servers. Since each server implements the publish/subscribe protocol STOMP for its outputs to clients, the outputs can be captured by the other server, which listens to the STOMP publication. Each server therefore implements a back-to-back (B2B) client, which contains both a STOMP subscriber and a REST output, and shown in figure 7. The B2B client starts after both clients are running, subscribes to the source server, and forwards all received BML documents to the sink server, subject to a filter if so configured.

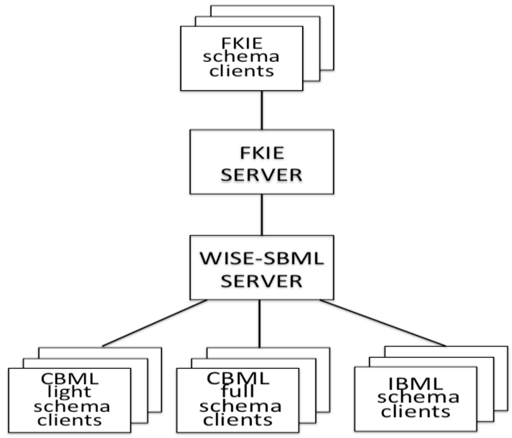

Figure 6: MSG-85 Linked Server Architecture.

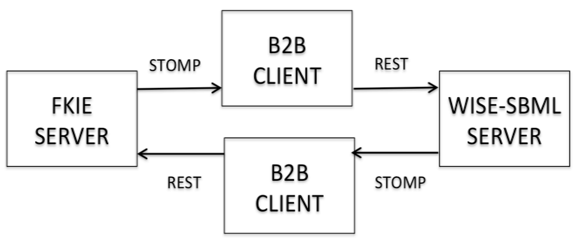

Figure 7: Server Linking Architecture.

One potential problem was detected when using this arrangement: a BML document could circulate between the two servers indefinitely unless some mechanism is provided to break the loop. We achieved breaking the loop by adding a parameter to the REST and STOMP interfaces, intended for use by the B2B clients but not by regular C2-Sim clients. This parameter *firstforwarder* is set by the first server to publish each document, to identify itself by a unique code (we use the server's IP address for this). The B2B client checks this code for each received document and does not forward any document to the server that was the first forwarder.

We have considered the question of whether the architecture shown in figure 5 is scalable to multiple servers. The answer appears to be "no", because a loop can exist that does not include the first forwarder. In the general case, it would be necessary to devise a mechanism that can break every possible loop. We will leave the creation of this mechanism to future work.

5.0 CONCLUSIONS

Work in C2-simulation interoperation, using emerging SISO standards MSDL and C-BML, continues to make progress as described above. Practical implementation by MSG-085 team members is leading to understanding of how military operations can be supported effectively by this technology. National implementations in both C2 and simulation systems, coupled with supporting open source server software, make the feasibility of this approach clear. However, a natural aspect of this flowering of development is the existence of multiple, related schemata among the various C2-Sim clients. Availability of translating services and distributed, linked servers supported by a robust, high-performance platform can be an enabler to gain critical experience through expanded application and interoperation of various national systems, as exemplified in MSG-085. This work has two commendable results: the interoperating systems will support operational experimentation now being planned by MSG-085, and also will continue to provide the experience needed for SISO MSDL and C-BML product development groups to produce effective standards, based on technical approaches that have been demonstrated to be effective.

The multi-schema and multi-server architectures described in this paper are the next step toward larger BML systems, based on distributed servers. We anticipate that future coalition partners will incorporate simulation into their C2 systems routinely, with every coalition partner providing a simulation that represents their own forces. Such a future could require complex architectures consisting of C2 and simulation systems, all of which are clients of a distributed system of servers supporting multiple schemata. There is considerable room for further research on these topics.

REFERENCES

[1] Sudnikovich, W., J. Pullen, M. Kleiner, and S. Carey, "Extensible Battle Management Language as a Transformation Enabler," in SIMULATION, 80:669-680, 2004

[2] Simulation Interoperability Standards Organization, Standard for: Military Scenario Definition Language (MSDL) http://www.sisostds.org/DigitalLibrary.aspx? Command=Core_Download&EntryId =30830

[3] Blais, C., K. Galvin and M. Hieb, "Coalition Battle Management Language (C-BML) Study Group Report," IEEE Fall Simulation Interoperability Workshop, Orlando FL, 2005

[4] Schade, U. and Hieb, M., "Formalizing Battle Management Language: A Grammar for Specifying Orders," 2006 Spring Simulation Interoperability Workshop, IEEE Spring Simulation Interoperability Workshop, Huntsville, AL, 2006

[5] Hieb, M. and U. Schade, "Formalizing Command Intent Through Development of a Command and Control Grammar," 12th International Command and Control Research and Technology Symposium, Newport, RI, 2007

[6] Perme, D., M. Hieb, J. Pullen, W. Sudnikovich, and A. Tolk, "Integrating Air and Ground Operations within a Common Battle Management Language," IEEE Fall Simulation Interoperability Workshop, Orlando FL, 2005

[7] Sudnikovich, W., A. Ritchie, P. de Champs, M. Hieb, and J. Pullen, "NATO Exploratory Team – 016 Integration Lessons Learned for C2IEDM and C-BML," IEEE Spring Simulation Interoperability Workshop, San Diego CA, 2006

[8] Hieb, M., S. Mackay, M. Powers, M. Kleiner, and J. Pullen, "The Environment in Network Centric Operations: A Framework for Command and Control," 12th International Command and Control Research and Technology Symposium, Newport, RI, 2007

[9] Galvin, K., W. Sudnikovich, P. deChamps, M. Hieb, J. Pullen, and L. Khimeche, "Delivering C2 to M&S Interoperability for NATO - Demonstrating Coalition Battle Management Language (C-BML) and the Way Ahead," IEEE Fall Simulation Interoperability Workshop, September 2006

[10] Pullen, J., M. Hieb, S. Levine, A. Tolk, and C. Blais, "Joint Battle Management Language (JBML) - US Contribution to the C-BML PDG and NATO MSG-048 TA," IEEE European Simulation Interoperability Workshop, June 2007

[11] de Reus, N., R. de Krom, O. Mevassvik, A. Alstad, U. Schade and M. Frey, "BML-enabling national C2 systems for coupling to Simulation,", IEEE Spring Simulation Interoperability Workshop, Newport, RI, 2008

[12] Gustavsson, P., M.R. Hieb, M. Groenkvist, V. Kamath, Jakob Blomberg, and Joakim Wemmergard. "BLACK-CACTUS – Towards an Agile Joint/Coalition Embedded C2 Training Environment," IEEE Spring Simulation Interoperability Workshop, Providence, RI, 2008

[13] Levine, S., L. Topor, T. Troccola, and J. Pullen, "A Practical Example of the Integration of Simulations, Battle Command, and Modern Technology," IEEE European Simulation Interoperability Workshop, Istanbul, Turkey, 2009

[14] Pullen, J., D. Corner, R. Wittman, A. Brook, O. Mevassvik, and A. Alstad, "Technical and Operational Issues in Combining MSDL and C-BML Standards for C2-Simulation Interoperation in MSG-085," NATO Modelling and Simulation Symposium, Stockholm, Sweden, October 2012

[15] Remmersmann, T., U. Schade, L. Khimeche, and B. Gautrau, "Lessons Recognized: How to Combine BML and MSDL," IEEE Spring Simulation Interoperability Workshop, Orlando, FL, 2012

[16] Heffner, K. C. Blais and K. Gupton, "Strategies for Alignment and Convergence of C-BML and MSDL," IEEE Fall 2012 Simulation Interoperability Workshop, Orlando, FY, 2012

[17] Pullen, J., D. Corner and R. Wittman, "Next Steps in MSDL and C-BML Alignment for Convergence, IEEE Spring 2013 Simulation Interoperability Workshop, San Diego, CA, 2013

[18] Pullen, J., D. Corner, P. Gustavsson, and M. Grönkvist, "Incorporating C2---Simulation Interoperability Services into an Operational C2 System," International Command and Control Research and Technology Symposium 2013, Alexandria, VA

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

[20] Pullen, J. and K. Heffner, "Supporting Coalition Battle Management Language Experiments with Scripted Web Services", NATO 2009 Modeling and Simulation Symposium, Brussels, Belgium, 2009

[21] McAndrews, P., L. Nicklas and J. Pullen, "A Web-Based Coordination System for MSDL/C-BML Coalitions," IEEE Spring Simulation Interoperability Workshop, San Diego, CA, 2012

[22] Pullen, J., D. Corner, P. Gustavsson, and M. Grönkvist, "Incorporating C2---Simulation Interoperability Services Into an Operational C2 System," International Command and Control Research and Technology Symposium 2013, Alexandria, VA

[23] Gautreau, B. *et al.*, "Lessons Learned from NMSG-085 CIG Land Operation Demonstration," IEEE Spring Simulation Interoperability Workshop, San Diego, CA, 2012

[24] Brook, A., Patel, B., Heffner, K. and Hassaine, F., "NATO MSG-085 Standardisation for C2-Simulation Interoperation: Autonomous Air Operations Experiments", 13S-SIW-009, IEEE Spring 2013 Simulation Interoperability Workshop, San Diego, CA, 2013