MSDL and C-BML Working Together for NATO MSG-085

Dr. J. Mark Pullen and Douglas Corner
C4I Center
George Mason University
Fairfax, VA 22030, USA
+1 703 993 3682
{mpullen, dcorner}@c4i.gmu.edu

Dr. Robert Wittman
MITRE Corporation
3504 Lake Lynda Drive, Suite 410
Orlando, FL 32817
+1 321-235-7601
rwittman@mitre.org

Adam Brook
QinetiQ, Cody Technology Park
Ively Road, Farnborough, Hampshire GU14 0LX, UK
+44 1252 396427
rabrook@QinetiQ.com

Ole Martin Mevassvik and Anders Alstad
Norwegian Defence Research Establishment, FFI
PO Box 25, NO-2027 Kjeller, Norway
+47 6380 7423
{ole-martin.mevassvik, anders.alstad}@ffi.no

Keywords:
Coalition Battle Management Language, Military Scenario Definition Language,
Command and Control Simulation Interoperability

Two SISO technology threads are associated with the general area of C2-Simulation Interoperability: the Military Scenario Definition Language (MSDL) and the Coalition Battle Management Language (C-BML). The two have been developed independently since 2005, with the understanding that, for full effectiveness, they must operate together. This paper reports on an effort to achieve a basic level of compatibility between MSDL and C-BML in support of NATO MSG-085 "Standardization for C2-Simulation Interoperation." The purposes and states of development of MSDL and C-BML are described, followed by a discussion of issues that must be resolved for the two to work together harmoniously. The MSG-085 project's purposes and general approach are set forth, followed by a description of work currently underway to use MSDL and C-BML together in that project, including software needed for supporting services. The paper concludes with a projection of the path to full harmonization of these two SISO standards.

1. Overview

Two of SISO's standards developments, the Military Scenario Definition Language (MSDL) and the Coalition Battle Management Language (C-BML) are in the process of converging to yield a powerful, standardized technical specification for command and control (C2) systems to interoperate with simulation systems, a capability long sought to support a range of important military operations. This paper describes developmental work leading to a capability 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.

The sections which follow in this paper describe the areas where MSDL and C-BML must function together, followed by a summary of the technical and operational requirements of MSG-085 that drove the approach we are taking. This is followed by a description of implementations that address those needs: a server environment assembled to provide support needed by MSG-085 in experimenting with operations using MSDL and C-BML, along with a description of client implementations in both C2 and simulation systems. The paper concludes with a summary of MSG-085 plans and a proposed path forward to full harmonization of these two SISO standards.

2. Background of MSDL and C-BML

This section summarizes the history and status of the two standards.

2.1 MSDL

The Military Scenario Definition Language [24] grew out of a desire within the OneSAF Program to reduce scenario development time and cost and then be 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) that could easily and dependably be consumed by current and evolving simulations. This was prototyped within OneSAF during its early architectural development phase between 2001 and 2004 and then proposed for additional study to the larger international simulation community within the

SISO Study Group (SG) construct. The Study Group concluded that there was in fact a community-wide need for a standardized military scenario format to reduce development time and cost, and to enable sharing of valuable scenario products. A standardized scenario format was also 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 this manual process.

In 2006, the formal SISO MSDL standard Product Development Group was established with the specific intent of producing a standard Military Scenario Definition Language data model. They began by reviewing the OneSAF program provided MSDL specification and ultimately decided to refine the 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; and removing elements that were under study or standards development such as the Course of Action structure that was equivalent to the work being pursued under the SISO C-BML PDG.

The MSDL PDG culminated in November of 2008 with a formal SISO approved Version 1.0 MSDL specification and XML schema set. The PDG continues to meet regularly and plans to integrate the MSDL specification with the C-BML standard upon its completion. Currently members of both the MSDL and C-BML PDGs are experimenting with ways to reference C-BML instance documents at initialization time within MSDL documents so that plans and orders can be provided and are consistent with the task organizations and tactical graphics as defined within a MSDL document.

Use of MSDL continues to expand not only because of its promulgation within OneSAF, but also through additional US and international community involvement and investments by the US Army Modeling and Simulation Office (AMSO), Air Force, Marine Corps as well as NATO activities including Spain, France, the United Kingdom, Norway, Germany, Canada, and others.

2.2 C-BML

Battle Management Language (BML) and its various proposed extensions are intended to facilitate interoperation among command and control (C2) and modeling and simulation (M&S) systems by providing a common, agreed-to format for the exchange of information such as orders and reports. In recent implementation, this has been accomplished by providing 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 currently being adopted by the US Department of Defense and its coalition allies [1,2].

SISO's BML study group created a plan to develop a Coalition BML (abbreviated C-BML) standard in 2005 [3] and the corresponding product development group (PDG) was chartered in 2007. Progress has been slow, for reasons documented in [4]. However, the C-BML Phase 1 Draft Standard reached the point of Trial Use in 2011 and is expected to be balloted in 2012. The approach has generally followed the Lexical Grammar approach introduced by Schade and Hieb [5,6]. Informing the standardization process have been multiple projects under various US DoD sponsors [7,9,11] and an ongoing sequence of experimental BML configurations developed and demonstrated by the members of NATO MSG-048 and MSG-085 [8,10,12,13,14,15] (see next section).

2.3 Areas of Convergence

Areas where MSDL and C-BML must agree on representations are described below.

Task Organization Definition: Various ongoing projects, including SISO C-BML development, have independently derived formats for the friendly and adversary order of battle (ORBAT), also called Task Organization in military orders. The primary requirements are (1) identify the name and type of each unit (including its US MIL STD 2525C icon or NATO APP-6C) with enough detail to allow a common interpretation of the unit type by many different simulations, mission command, and C2 systems; (2) identify command relationships (parent and child). MSDL has standardized an XML document structure for this purpose, which has been used successfully by multiple national teams in MSG-085. The C-BML Phase 1 schema draft contains only composite definitions (including Task, but no Task Organization); no full Order or Report is in the normative specification. It is currently under debate by members of the development team whether the MSDL Task Organization specification, which relies on the 2525C symbol identifier to define a unit's type, is sufficient or if an alternative approach aligned with JC3IEDM is necessary. One such approach, proposed by members of the team (Ole Martin Mevassvik and Anders Alstad) is to explicitly define all unit types of interest within a JC3IEDM unit structure and then reference these unit types within the MSDL and C-BML instance documents. This would allow for both specifying entity categories and defining entity types used by members of a specific simulation and C2 federation, e.g. by defining Abrams Main Battle Tank as a specific type. As we gain more experience pairing MSDL and C-BML documents, we look forward to providing a reasoned resolution to this debate.

Tasking Definition: The definition of actions to be carried out, their interrelations, and the control measure to be employed, is the basic reason for existence of C-BML. The MSDL standard includes a placeholder for an initial tasking which has not been developed in detail; it has no provision for a continuing flow of orders, or for reports. By contrast, C-BML has a well-developed Trial Use draft, based on experience developed in NATO MSG-048 that supports both initial and subsequent orders, and it also provides for reports from simulations (and potentially also from humans), providing situational awareness information to be made available to C2 systems. Here again, the opportunity is clear for MSDL version 2 to adopt the Tasking definition as standardized under C-BML (expected to be formalized in 2012).

Tactical graphics: MSDL has adopted the tactical graphics (unit type symbols and descriptive data) of US MIL STD 2525C and NATO APP-6C (which are very similar). C-BML also needs some of this information. We conclude that both MSDL and C-BML should adhere to the existing tactical graphics standard for the environment in which they are used.

3. MSG-085 and its Requirements

Technical feasibility of coalition BML was demonstrated by NATO MSG-048 in a project conducted 2006-2009. The basic architecture for C2-simulation operations was developed by MSG-048, as described in [8,10,12,21,22]. The final experimentation event of MSG-048 involved six national C2 systems, five national simulations, and two supporting software systems, as shown in figure 3.1.

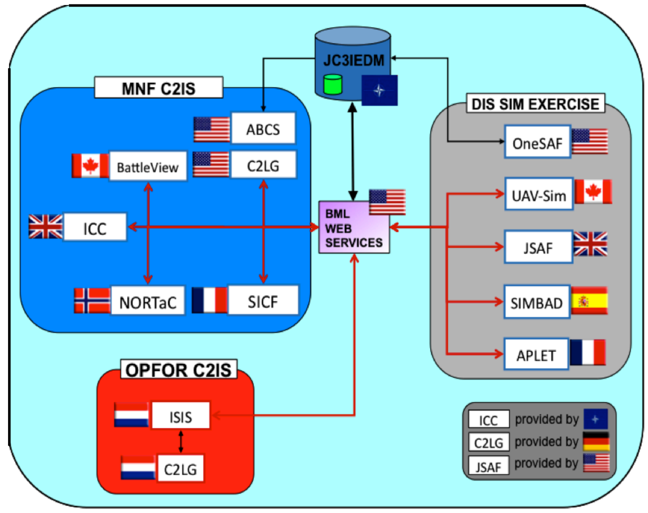

Figure 3.1 MSG-048 Experimentation Architecture

The follow-on Technical Activity, MSG-085, is chartered to demonstrate and facilitate the operational utility of MSDL and C-BML in military coalitions. 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 participated in a demonstration of the initial infrastructure, held at the Interservice/Industry training, Simulation and Education Conference (I/ITSEC) 2011. The demonstration was conducted over the Internet, with participating sites in Norway, England, Virginia, and Florida. Some of the participating systems are described below as examples of MSDL and C-BML implementation.

4. Coalition Server Implementation

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. This section describes the general scripted server capability and how it has been enhanced to provide infrastructural needs for aggregating MSDL files, storing and publishing C-BML orders and reports, and coordinating operation of multiple C2 and simulation systems. The top-level architecture of a C2-simulation coalition using these services is shown in Figure 4.1. These implementations are available at http://c4i.gmu.edu/OpenBML as open source software.

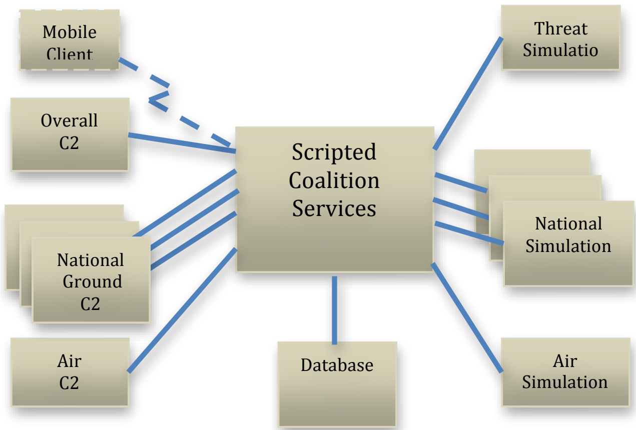

Figure 4.1 Overall Client-Server Architecture

4.1 Scripted Server

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, 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

The heart of SBMLServer is a scripting engine, introduced in [2], that implements a BML WS by converting BML data into a database representation and also retrieving from the database and generating BML as output. It could be scripted to work with any XML-based input and any relational database. Current SBMLServer scripts implement the Joint Command, Control and Consultation Information Exchange Data Model (JC3IEDM). However, any logically consistent and complete data model could replace JC3IEDM. Further, we have developed a Consolidated Scripting Language (CSL) that represents the scripting information in a concise, practical format. The architecture of the server system is shown in Figure 4.2.

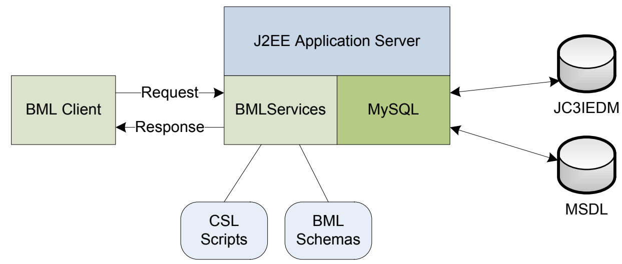

Figure 4.2 Scripted BML Server Architecture

4.2 Adapting SBMLServer to Support MSDL

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. With a simple addition to SBMLServer, we were able to implement the required logic in CSL scripts. 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 for the whole coalition assembled up to that time. Upon signal from the master controller, via the Status Monitor and Control service described section 4.4 below, 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.

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

Transactions are edited 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 signaled 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 can alternatively execute a query and retrieve the same information. This query may also be used by organizations joining the exercise after the MSDL data has been published.

All the elements submitted by clients under a single scenario are aggregated in to a single MSDL document. 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. This 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 4.3.

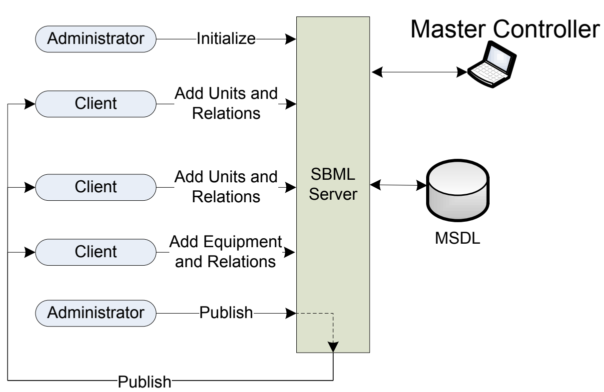

Figure 4.3 MSDL Server Operation

4.3 C-BML Application of SBMLServer

SBMLServer was used to support NATO MSG-048 experimentation in 2008 and 2009, as described in [12] and [17]. This resulted in a broader understanding of the range of requirements for effective prototyping support of C2-simulation in experimentation. A particularly important capability is publish/subscribe, which avoids the unnecessary server load created by clients polling the server, which is required otherwise.

The current SBMLServer implementation and scripts support a direct SQL interface, used with a MySQL database server, and a publish/subscribe capability, illustrated in Figure 4.4, using the Java Message Service (JMS) as implemented by the open source JBoss Application Server (see http://www.jboss.com). The server also implements the XML Path Language (XPATH) (see http://www.w3.org/ TR/xpath), wherever a relative path in the XML input is required, and a RESTful interface. The server has a range of capabilities, developed based on experience gained in using it to support MSG-048. Basic functions are to accept *push* of orders and reports, *publish* them, store the information they contain, and allow clients to *pull* copies on demand. Details are available in [18,19,22].

MSG-085 has made new demands on the SBMLServer. Because national groups have implemented various "dialects" of BML, developed at different times, there is a need to interoperate different schemas. Because the SBMLServer maintains XML documents in a common database representation (JC3IEDM), it is possible to push in orders and reports from one version and pull the information out under a different schema. This capability is already functional and is expected to provide significant flexibility to MSG-085 as it pursues its program of work.

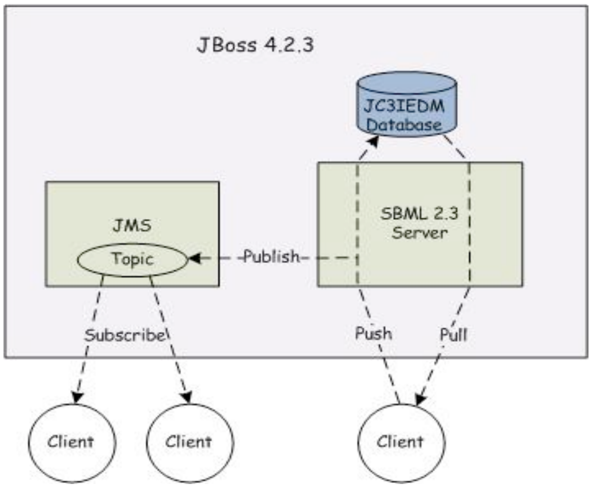

Figure 4.4 Publish/Subscribe Architecture for SBML

4.4 Status Monitoring and Control Service

Experience in MSG-048 taught us that it is impractical to coordinate multiple interoperating C2 and simulation systems using human operators with simple spoken coordination. In preparation for MSG-085 C2-simulation coalitions that are likely to be more complex than those of MSG-048, the GMU C4I Center has developed a Status Monitoring and Control System (SMCS) that provides a means of displaying to all system operators the status of each participating system, along with a "Master Controller" capability that can provide coordinated direction to the systems, either through their human operators or, through web services, interfaced directly to the software systems. An example interface webpage is shown in Figure 4.5.

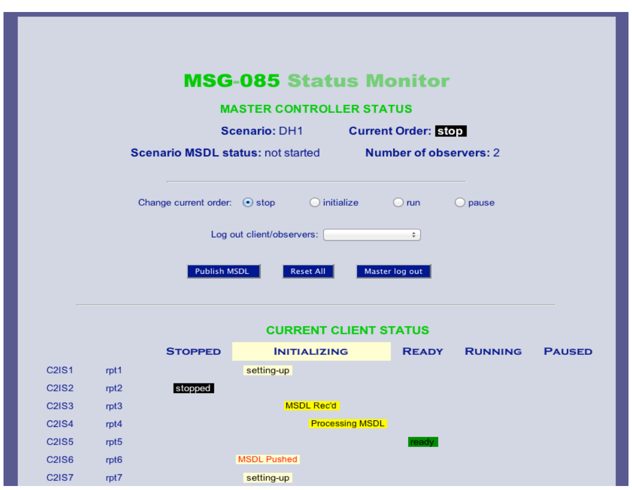

Figure 4.5 Status Monitoring and Control Webpage

5. MSDL/C-BML Client Implementations

Some of the C2 and simulation systems to be used by MSG-085 already have early BML implementations. However, only OneSAF had an MSDL implementation as of Spring 2011. This section describes how MSDL was implemented by QinetiQ in the ICC and JADOCS C2 systems and the JSAF simulation system, how an MSDL export capability in NORTaC-C2IS was developed by FFI and how MITRE updated C-BML in the OneSAF simulation system.

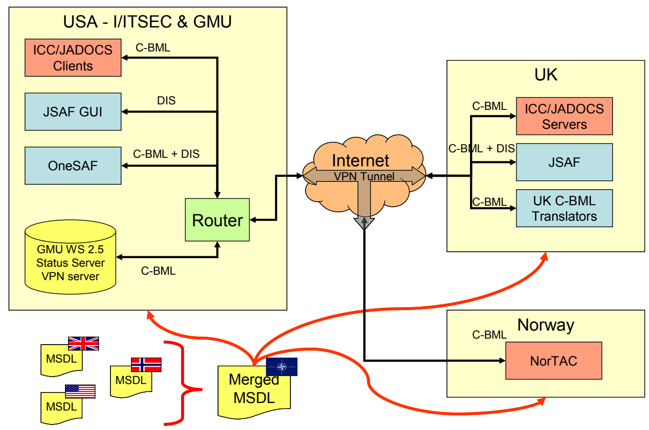

Figure 5.1 System Architecture for 2011 I/ITSEC MSDL/C-BML Demonstration Harness 1

Figure 5.1 shows the top-level system architecture implemented at I/ITSEC in 2011 for MSG-085 Demonstration Harness 1. This system, which was distributed across four sites, was coordinated using the GMU status service described above.

5.1 ICC/JADOCS

The NATO Integrated Command and Control system (ICC) and the Joint Automated Deep Operation Coordination System (JADOCS) were used as command and control applications during the demonstration. Both of these systems are networked, server/client, database applications. However, neither has a native capability for processing either C-BML or MSDL, but both had been used with previous C-BML experimentation and were able to display C-BML reports through the use of existing translator applications. Air Tasking Orders (ATOs) and Airspace Control Orders (ACOs) prepared in ICC were used to generate C-BML air tasks. The C-BML translators were updated to be compatible with the latest version of the SBML server.

Use of MSDL has been investigated to see how easily it is possible to populate and represent the internal databases used by C2 systems. This is a new use for MSDL, which has until now been used primarily to initialize simulations. It is now possible to translate both ways between MSDL TaskOrgs and the JADOCS Friendly Order of Battle (FOB) database. The corresponding translation capability for ICC has not yet been implemented but this is not considered to be a very difficult activity. Future developments will address the exchange of MSDL tactical graphics with these C2 applications.

The ICC ATO and ACO databases are interrogated to populate C-BML orders with air tasks, which are then published through the SBMLServer web services and subscribed to and consumed by the relevant simulation systems. Neither ICC nor JADOCS uses a JC3IEDM database and this leads to interesting mapping problems with C-BML as it stands, *e.g.* numerous air task definitions are not represented in JC3IEDM.

5.2 JSAF

JSAF is an entity-level constructive simulation, which has been used in MSG-085 on several occasions. C-BML orders and reports are processed using a federated interface application.

A new two-way MSDL interface has been produced, which permits correctly located TaskOrgs to be created from JSAF scenario files and *vice versa*. This therefore allows MSDL to be used for both initialization and check-pointing while remaining consistent with other systems in the C-BML environment.

The JSAF C-BML interface declares newly discovered units (found by JSAF sensor models) and publishes MSDL “snippets” to support the representation of perceived truth in the C-BML system.

5.3 NORTaC-C2IS

The Norwegian Tactical Command and Control Information System (NORTaC-C2IS) was developed by Kongberg Defence Systems (KDS) for the Norwegian Army. With support from KDS, Norwegian Defence Research Establishment FFI developed the FFI C2-gateway (a BML interface to NORTaC-C2IS) in 2008 for use in NATO MSG-048 experiments [13]. FFI C2-gateway reads orders from the NORTaC-C2IS database and translates them into a BML format and inserts BML reports into the C2IS database for presentation to the user. The FFI C2-gateway interacts with the SBMLServer by pushing BML orders and subscribing to BML reports.

For the 2011 demonstration, FFI extended the FFI C2-gateway with a MSDL export capability in order to participate in simulation initialization through the SBMLServer. This new capability extracts the task organization for the Norwegian forces from the NORTaC-C2IS database and creates an MSDL document. During the demonstration this capability was used to initialize JSAF with the Norwegian units. JSAF subsequently simulated a BML order developed in NORTaC-C2IS tasking the same units. Figure 5.2 shows a reconnaissance task in NORTaC-C2IS.

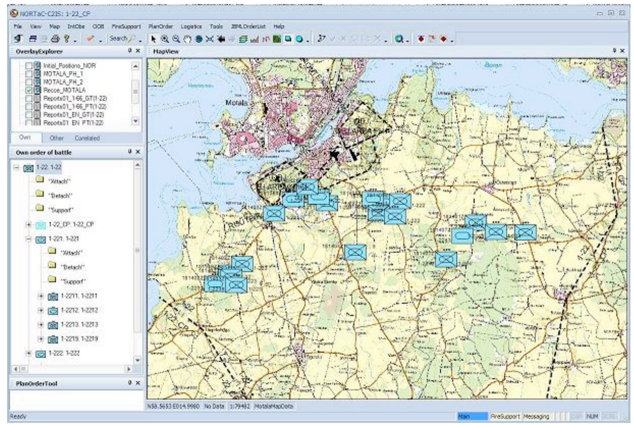

Figure 5.2 Reconnaissance task in NORTaC-C2IS

The NORTaC-C2IS database utilizes the JC3IEDM. The C2IS have functionality for defining overlays where units and tactical graphics can be positioned. The FFI C2-gateway allows the user to generate MSDL by loading and selecting such overlays (i.e. a JC3IEDM “context”). The current MSDL capability only extracts the static task organization and for each unit/equipment adds the position if it is defined in the selected overlay. The FFI C2-gateway currently does only export data for the following MSDL sections: Forces/Sides, Units and Equipment.

Experience has shown that various systems differ regarding units in the task organization for which they require positions. Some systems cannot generate aggregated positions for e.g. troops and companies, and/or cannot generate positions for subordinate units. FFI therefore found it necessary to use the C2IS to define positions for every unit in the task organization.

While extracting the task organization from JC3IEDM was relatively straight-ahead, the mapping from the JC3IEDM type-structure to MSDL compliant symbol IDs was complicated. The mapping was partly implemented by utilizing existing NORTaC-C2IS mapping files and partly by using mapping rules found in [25].

5.4 OneSAF

The MITRE OneSAF part of this demonstration was funded by the Army Modeling and Simulation Office to effectively integrate MSDL and C-BML data models into a working OneSAF solution. The MITRE effort fits nicely as part of the long-term OneSAF MSDL/C-BML capability and MITRE appropriately reused and continues to extend the OneSAF Version 5.1 MSDL capability and plans to step up to OneSAf Version 5.1.1 when available and will ultimately hand-over the code for integration into Version 6.0.

5.3.1 Baseline OneSAF Capability

Although OneSAF Version 5.0 had a basic local-file-based MSDL import and export capability there were a number of challenges that required a "crawl, walk, run" strategy to overcome and meet the cost and schedule timelines associated with this and the longer-term MSDL/C-BML integration effort. These challenges included:

1) Existing OneSAF specific MSDL schema required modifications;

2) The use of specific OneSAF MSDL tags,

3) No support for access (post or receive) to/from MSDL/C-BML web-server,

4) No support to relate C-BML orders/reports to MSDL provided units and platforms during initialization and runtime, and

5) No support to appropriately task OneSAF units and platforms with C-BML provided orders.

These challenges and their resolution for this demonstration are discussed in the following sections.

First, OneSAF Version 5.0 modifies the MSDL standard schema to support the "any" XSL element. The "any" element was added as a final element to all complex MSDL element types to directly support extensions to the MSDL schema while still allowing W3C compliant XML document validation. The impact to the MSDL Version 1.0 standard is that the inclusion of the "any" element precludes the use of the "all" compositor, which the MSDL schema employed within several complex elements. The solution employed within OneSAF was to replace the "all" with "sequence" compositors. These compositors are largely identical with one important exception. "All" compositors allow sub-elements to be provided in any order while the "sequence" compositor requires all sub-element to be in the defined sequential order. Both compositors allow zero to many of each sub-element, but in a sequence if they are present they must be ordered. So a perfectly valid MSDL Version 1.0 schema employing an unordered sub-element set may not validate within the OneSAF's current MSDL importer. While this is not an insurmountable complication, there were enough issues with OneSAF Version 5.1 import capability that schema-based validation was turned off for this part of the demonstration effort. As part of the longer-term effort, MITRE is improving the MSDL validation capability to report and fix validation issues when and where appropriate. The "any"-based design to support extensibility has been proposed to the PDG for inclusion in the standard but has not yet been formally accepted.

Secondly, OneSAF uses the modified MSDL schema to support a OneSAF namespace using the <onesaf:> prefix to specifically identify OneSAF representations of units and platforms to be imported into the OneSAF. For demonstration purposes, it was planned that the mappings for the unit and platform representations provided by other systems would be mapped to a OneSAF unit or platform by a knowledgeable OneSAF scenario developer and then included through an editor into the MSDL document. This again was seen as a significant shortcoming and MITRE is improving the ability to match OneSAF units and platforms (including life forms) to appropriate 2525B symbol identifiers provided within a MSDL Version 1.0 standard compliant document in a more automated fashion.

Thirdly, OneSAF Version 5.0 had no capability to access an MSDL payload from a webserver and very limited capability to access a C-BML payload from a webserver. For the demonstration MITRE provided a basic application to access the GMU Scripted BML server, store the MSDL document within a local file, and then import that MSDL file through the existing OneSAF Version 5.0 import capability. MITRE is improving this capability to allow a web-based import from the Management and Control Tool (MCT) patterned after the file-based MSDL import. An MSDL Version 1.0 compliant export capability also will be provided.

Fourthly, OneSAF Version 5.0 provided no support to relate units and platforms within OneSAF to the C-BML orders and reports during initialization and runtime. The agreement within the MSG-085 demonstration team was to associate the MSDL provided unique identifier element (available on every complex element) with the appropriate C-BML payload (task, order, report, etc.). Additionally, per earlier demonstration agreements the group also agreed to use the MSDL element <mid:reference> element to hold the file name of the C-BML document to use as part of the import process. Where the sub-element <mid:type> holds "C-BML reference" and the <mid:reference> sub-element holds the file name such as "./cbml-demo-routewhere-42011.xml". For this demo MITRE created a hard-coded example of how C-BML files could be read, referenced to an existing OneSAF unit or platform, and used to populate the

Mission Editor within the MCT. MITRE is now working to populate the Mission Editor relating local-file-based C-BML taskings to appropriate unit and platforms.

Finally, there was no runtime capability in OneSAF Version 5.0 to relate C-BML tasks to OneSAF units and platforms based on a unique identifier. MITRE developers did not complete a demonstrable capability in this area but are actively engaged and expect to complete this capability for integration into OneSAF version 6.0.

Although there is more work to be done for OneSAF, this demonstration showed the value of being able to share standardized scenario and order-based in a web-served environment between different and multi-national simulations and C2 devices.

6. The Way Forward

We conclude with summaries of the expected way forward for MSG-085 and for SISO MSDL and C-BML.

6.1 MSG-085 Planned Activities

MSG-085 is actively engaged in developing both scenarios and use cases for military operations (training and mission support) using MSDL/C-BML C2-simulation interoperation. In addition to the systems described in this paper, other national systems from France, Spain, and the USA were demonstrated at I/ITSEC'11. Further, a set of French-German experiments to interoperate national C2 systems and simulations, using MSDL and the version of BML used by MSG-048, are reported in [23].

MSG-085 is planning an experimentation event in 2012, leading up to participation in a NATO exercise in 2013 that can validate the conclusions of the OSG with regard to the expected operational utility of C2-simulation interoperation using MSDL and C-BML.

6.2 MSDL/C-BML Convergence

The ongoing work of the MSDL and C-BML Product Development groups affords ample opportunity for these standards to achieve full compatibility, in compliance with both the SISO Standards Activity Council (SAC) guidance and common sense. C-BML should adopt the MSDL Task Organization (ORBAT) while MSDL should adopt the C-BML Task definition and both should employ the existing NATO APP-6A standard for tactical graphics.

The use of MSDL has gone a long way to solve the initialization problems which were identified in the earlier work of MSG-048. It is now possible to ensure that there is consistency of representation across all systems. However, the processes which are required to use this information now need to be understood and developed. For example, JSAF requires entity (or equipment) information which may not be required, and hence not be present, in the C2 system which is being used to prepare the C-BML orders. This means that processes are needed to manage this information, perhaps inferring it using a rule system or creating it manually through an MSDL editor such as MSDE. Similarly, JSAF does not require all the higher unit echelons, which are present in the MSDL TaskOrg so these too have to be subject to a filtering process.

6.3 Conclusions

Work in C2-simulation interoperation, using emerging SISO standards, continues to make progress as described in this and companion papers. 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. 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.

References

[1] Carey, S., M. Kleiner, M. Hieb, and R. Brown, "Standardizing Battle Management Language – A Vital Move Towards the Army Transformation," IEEE Fall Simulation Interoperability Workshop, Orlando, FL, 2001

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

[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] Abbott, J., J. Pullen and S. Levine, "Answering the Question: Why a BML Standard Has Taken So Long to Be Established?" IEEE Fall Simulation Interoperability Workshop, Orlando FL, 2011

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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

[10] 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

[11] 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

[12] Pullen, J. *et al.*, NATO MSG-048 Coalition Battle Management Initial Demonstration Lessons Learned and Way Forward, IEEE Spring Simulation Interoperability Workshop, Providence, RI, 2008

[13] 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

[14] 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

[15] 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

[16] 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

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

[18] Corner, D. J. Pullen, S. Singapogu, and B. Bulusu, "Adding Publish/Subscribe to the Scripted Battle Management Language Web Service," IEEE Spring 2010 Simulation Interoperability Workshop, Orlando, FL, 2010

[19] Pullen, J., D. Corner, S. Singapogu, B. Bulusu, and M. Ababneh, "Implementing a Condensed Scripting Language in the Scripted Battle Management Language Web Service," SCS/SISO Euro-Simulation Interoperability Workshop, Ottawa, Canada, 2010

[20] Pullen, J. *et al.*, "An Expanded C2-Simulation Experimental Environment Based on BML," IEEE Spring Simulation Interoperability Workshop, Orlando, FL, 2010

[21] Heffner, K. *et al.*, "NATO MSG-048 C-BML Final Report Summary," IEEE Fall 2010 Simulation Interoperability Workshop, Orlando, FL, 2010

[22] Nicklas, L., J. Pullen, and D. Corner, Dynamic Publish/Subscribe Topics in the Scripted BML Server, IEEE Spring Simulation Interoperability Workshop, Orlando, FL, 2011

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

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

[25] IBM ILOG JViews Maps for Defence V8.7, *http://publib.boulder.ibm.com/infocenter/jviewmap/v8r7/topic/com.ibm.ilog.jviews.defense.doc/Content/Visualization/Documentation/JViews/JViews_Defense/_pubskel/ps_usrprgdef792.html*

Author Biographies

ADAM BROOK is a member of the Training and Simulation Systems Group at QinetiQ. He has worked in the fields of distributed training systems and simulation-C2 interoperability since 1996 and has been a member of MSG-048 since April 2007 and MSG-085 since its inception. He is advising the UK MoD Defence Science and Technology Laboratory on the use of MSDL and C-BML within the UK.

DR. J. MARK PULLEN is Professor of Computer Science at George Mason University, where he serves as Director of the C4I Center and also heads the Center's Networking and Simulation Laboratory. He is a Fellow of the IEEE, Fellow of the ACM and licensed Professional Engineer. He has served as Principal Investigator of the XBML, and JBML, IBML and SBML projects.

DOUGLAS CORNER is a member of the staff of the George Mason University C4I Center. He is the lead software developer on the SBML scripting engine.

DR. ROBERT WITTMAN is a Principal Modeling and Simulation for the MITRE Corporation in Orlando, Florida and is the Chief Architect for the Army's OneSAF Simulation. He holds a PhD in Industrial Engineering/Interactive Simulation from the University of Central Florida, an M.S. in Software Engineering from the University of West Florida, and a B.S. in Computer Science from Washington State University.

ANDERS ALSTAD is a Research Scientist at the Norwegian Defence Research Establishment (FFI). His work is in the field of modeling and simulation. His interests are in design and development of distributed systems.

OLE MARTIN MEVASSVIK is a Principal Scientist at the Norwegian Defence Research Establishment (FFI). His research interest is within modeling and simulation, with application to training and experimentation.