17th ICCRTS "Operationalizing C2 Agility"

Services to Support Experimentation for Operational Use of Simulations in Coalition Command and Control

Topics Experimentation, Metrics, and Analysis Modeling and Simulation Military and Civil-Military Operations

Names of Authors

J. Mark Pullen, Douglas Corner, Lisa Nicklas, and Priscilla McAndrews George Mason University C4I Center

Point of Contact

J. Mark Pullen 4400 University Drive, Fairfax, VA 22030 703-993-1538 {mpullen,dcorner,lnicklas,pmcandre}@c4i.gmu.edu

ABSTRACT

Incorporating simulation into operational military command and control is a long-sought goal. Recent developmental work in the NATO Modeling and Simulation Group (MSG) has demonstrated the potential of SOA-based systems to support this capability, but much remains to be accomplished before it can support operational military systems. Toward this end, the NATO Technical Activity MSG-085 is planning experimentation to establish capabilities and procedures for C2-simulation interoperation. A recognized technology gap for support of such experimentation is the ability to combine standards-based initialization of C2 and simulation systems with standards-based, unambiguous communication of orders, requests, and reports. This paper describes development of a capability achieving that combination, using the SISO standards MSDL and C-BML. It concludes with lessons learned during initial application of the capability.

1. Introduction

The potential for improved command and control (C2) by incorporating simulations has long been recognized [1,2]. Simulation as a built-in capability can enable training "as-youfight," realistic mission rehearsal, and automated course-of-action analysis. Accordingly, standardization of Battle Management Language (BML) has been underway for several years [3], although progress has been slow [4] despite development of a solid technical basis in machine grammar [5]. Moreover, recent development and experimentation by members of the NATO Modeling and Simulation Group (MSG) indicates that coalitions who are able to interoperate their C2 and simulation systems could have a significant operational advantage [6, 7, 8, 9, 10, 11]. We envision a day when the members of a coalition interconnect their networks, C2 systems, and simulations simply by turning everything on and authenticating, in a standards-based environment. This will be a major step forward in C2 for coalition agility.

The general paradigm for all of this work has been the service-oriented architecture of the Web service, where client C2 and simulation systems interchange orders, requests and reports in a common representation through a store-and-forward server. Figure 1 shows the basic architecture.



Figure 1. Basic C2-Simulation Architecture with SOA

Our work in BML has followed and extended this paradigm, as described in [12, 13, 14, 15]. We have created the Scripted BML server (SBMLServer), which is intended to be rapidly reconfigurable while providing a robust publish/subscribe infrastructure for experimentation in BML. NATO MSG-048 used this capability to support its 2009 experimentation, interoperating six national C2 systems with five national simulations as shown in Figure 2. We have progressed SBMLServer from a simple, eXtensible Markup Language (XML) based database service to an expanded capability that can support dynamic publish/subscribe topics and translate among multiple data schemas, configured under a concise scripting language, as open source software [16].



Figure 2. NATO MSG-048 Experimentation Architecture Using SBMLServer

One of the significant lessons learned in MSG-048 experimentation, as documented in that group's final report [17], is that a system-of-systems with the complexity of Figure 2

requires additional services to be operationally effective. Specifically, services are needed to:

- Provide for consistent initialization of the entire coalition of C2 and simulation systems.
- Provide coordination of execution state that is visible to the operators of the various systems and can interact automatically with the various software systems, to maintain a consistent state.

The remainder of this paper describes our work to expand the Scripted BML server in support of initialization and coordination. Section 2 describes the architecture of SBMLServer which served as a basis; section 3 describes convergence of developing standards for C2-simulation: the Military Scenario Description Language (MSDL) and Coalition Battle Management Language (C-BML). Section 4 describes how SBMLServer has been expanded to support this convergence; section 5 describes a companion synchronization service. The resulting software is available as open source on our website and is expected to prove useful in both US and NATO experimentation.

2. SBMLServer

The scripted approach employed in SBMLServer is widely used in software systems. For BML, it has these characteristics:

- While the details of BML electronic documents continue to grow and evolve, the basic functions of the server remain as described at the end of section 1 above.
- The script is capable only of the limited functionality needed to express mappings to and from BML and the relational data model used; for MSG-048 this was the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM).
- Skills needed to create the script are narrower than those needed to create a generalpurpose Web Service (WS) since scripts are written in the simpler special purpose scripting language.
- Development of the scripting engine can be a focus separate from the data mappings, resulting in improved performance and robustness.
- Ability to change the service rapidly, by modifying the script, reduces cost and facilitates prototyping.

Figure 3 shows the architecture of SBMLServer. The BML Input may be a *push* containing data (e.g. an Order) or may be a *pull* request for data. If successful, a push returns a response indicating success; a pull returns the requested data, formatted in BML per the script. If unsuccessful, either push or pull will return an error message. The SBMLServer operation is driven by elements of the BML that are individually processed by the script. These elements are XML aggregates, known as *BusinessObjects* (BO). (Alternately, they could be described by their grammatical role; they are *constituents* of the BML grammar [5]



Figure 3. SBMLServer Operating Configuration

As described further in [16], the SBML service runs under the JBoss J2EE Web service environment. Methods available provide for push and pull of a collection of Orders, Reports, and supporting services (such as NewUnitType and NewUnit, for database initialization). SBMLServer is capable of persisting the supporting information, using either an SQL-based relational database or Java Objects exchanged with the Reference Implementation (RI) JC3IEDM persistence service [12]. This dual capability enabled MSG-048 to combine US Army systems based on the RI with other NATO national systems that used the SQL database.

Two files control the BML/JC3IEDM conversion. The *BML schema* is an XML schema document (XSD) that specifies the structure and contents of the input document, while the *mapping script* contains scripting to process each BO. The BO is treated as an XML subtree rooted at a particular XML tag in the BML input. The BO script contains all the variable definitions and processing instructions needed for that subtree; it may be thought of as a subroutine, with parameters passed in and return variables passed back. The first phase of BML operation identifies the tags and the BO names with which they are associated. A BML transaction input may cause the invocation of multiple BOs. The root of the BML input document is also the name of the root BO; all other BOs are invoked by calls in the script. The script itself is coded in XML; it is derived at runtime from a more human-friendly version that is coded in the Condensed Scripting Language (CSL) as described in [16].

3. Convergence of MSDL and C-BML

MSDL [18] grew out of a desire within the US Army OneSAF program to reduce scenario development time and cost based on the ability to re-use a 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 standard XML data description, that could easily and dependably be consumed by current and evolving simulations. After prototyping within OneSAF, MSDL was proposed for use by the international military simulation community in 2004. A SISO Study Group (SG), formed to consider its potential, concluded that there was 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. Meanwhile, Coalition BML (C-BML) standardization progressed along a parallel track as described in [4].

It is clear that a convergence of MSDL and C-BML is needed to support standardized military operational use of BML. Working with MSDL developers, we determined that three areas of convergence are needed:

Task Organization Definition: Several independently derived formats exist for the friendly and adversary order of battle (ORBAT), also called Task Organization in military orders. The primary requirements for ORBAT 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 and thus no ORBAT. As a starting point for ORBAT, our work uses the MSDL ORBAT format. Given the proliferation of ORBAT formats, we anticipate that there may be a need to accept other formats containing the same basic information.

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. The opportunity is clear for MSDL version 2 to adopt the Tasking definition as standardized under C-BML (expected to be formalized in 2012); our work has proceeded on the assumption that this will happen.

Tactical graphics: MSDL has adopted the tactical graphics (unit type symbols and descriptive data) from standards 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.

4. Expanding SBMLServer to Support MSDL

Figure 4 shows the environment, which the expanded Scripted Coalition Services will support. (The Mobile Client does not yet exist, but is planned as a future capability.) The supporting server configuration is shown in Figure 5.



Figure 4. Expanded C2-Simulation Coalition Environment



Figure 5. Scripted BML Server Architecture With 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 in section 5 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 and synchronized state.

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 (except Forces/Sides, which applies to the entire scenario). These include:

- Force/Sides
- Units
- Equipment
- Installations
- Overlays
- Graphics

Transactions are validated as they are received, insuring correct format, unique unit and equipment names and object handles, and valid references between components. When 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 document 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 also may 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 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 6.



Figure 6 MSDL Server Operation

5. Status Monitor and Control Synchronization

Experience in MSG-048 taught us that it is impractical to coordinate multiple interoperating C2 and simulation systems through 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 7.

Based on experience gained in the demonstration described below, the SMCS has been refined to include a more straightforward graphic presentation, user configuration using a file of system states, a comment field to convey information not included in the configured states, and a graceful disconnect/restart strategy. We believe the resulting service will prove highly valuable for any distributed C2-simulation coalition, and even more valuable as the number of participating systems increases.



Figure 7. Status Monitoring and Control Webpage

6. Early Application

The system described here has seen some early use; responses of some MSG-085 participants are included in [19] and [20], which report on work in NATO MSG-085 that made experimental use of the MSDL/C-BML capability. Although formalized experimentation is planned as a part of MSG-085 activities, the initial applications have been focused on "getting it to work" in preparation for formal experimentation. Nevertheless, we can report good success in those endeavors. One of the most positive outcomes has been a determination that at least one C2

system, the Norwegian NORTaC-C2IS, readily adapted to MSDL initialization, as reported in [20].

A prototype of the Coalition Services as described above was demonstrated during the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011. With participating systems operating over the Internet from Norway, England, Virginia, and Florida (see Figure 8), the service worked well. The following steps were demonstrated:

- The overall process was coordinated using SMCS. The distributed nature of the demonstration made this essential.
- The Master Controller started the MSDL service from GMU in Virginia.
- The C2 system in Norway entered MSDL for the basic scenario.
- Other participating systems in England added their MSDL inputs.
- The SBML server published consolidated MSDL initialization to all systems.
- C2 systems in Norway (NorTAC) and England (ICC/JADOCS) submitted BML orders.
- The JSAF simulation carried out the orders.
- Tactical Reports were generated by JSAF and returned to the C2 systems via the SBML server.



Figure 8. System Architecture for 2011 I/ITSEC MSDL/C-BML Demonstration [20]

7. Conclusions

While considerably more experience and convergence will be necessary on the road to operational experimentation and eventual operational use of simulations in coalition C2, it is clear that there is much to be gained and also that there are no major technical barriers to success. We look forward to continued progress that will bring about our vision: when a military coalition is formed, its components simply interconnect their networks, C2 systems, and simulations, authenticate. and start operations. Our open source software. available at http://c4i.gmu.edu/OpenBML, is intended to support progress toward reaching this goal.

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] 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] 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
- [10] de Reus, N., R. de Krom, O. Mevassvik, A. Alstad, U. Schade and M. Frey, "BMLenabling national C2 systems for coupling to Simulation,", IEEE Spring Simulation Interoperability Workshop, Newport, RI, 2008
- [11] 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
- [12] 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
- [13] Pullen, J. and K. Heffner, Supporting Coalition Battle Management Language Experiments with Scripted Web Services, NATO 2009 Modeling and Simulation Symposium, Brussels, Belgium, 2009
- [14] 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
- [15] Pullen, J. et al., "An Expanded C2-Simulation Experimental Environment Based on BML," IEEE Spring Simulation Interoperability Workshop, Orlando, FL, 2010
- [16] Pullen J., D. Corner, and L. Nicklas, "Supporting NATO C2-Simulation Experimentation with Scripted Web Services," 16th International Command and Control Research and Technology Symposium, Montreal, Canada, 2011
- [17] Heffner, K. et al., "NATO MSG-048 C-BML Final Report Summary," IEEE Fall 2010 Simulation Interoperability Workshop, Orlando, FL, 2010

- [18] Simulation Interoperability Standards Organization, Standard for: Military Scenario Definition Language (MSDL) http://www.sisostds.org/DigitalLibrary.aspx? Command=Core Download&EntryId=30830
- [19] 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
- [20] Pullen, J., D. Corner, R. Wittman, A. Barooks, O. Mevassvik, and A. Alstad, "MSDL and C-BML Working Together for MSG-085," IEEE Spring Simulation Interoperability Workshop, Orlando, FL, 2012