A Web-Based Coordination System for MSDL/C-BML Coalitions

Priscilla McAndrews, Lisa Nicklas, and Dr. J. Mark Pullen
C4I Center
George Mason University
Fairfax, VA 22030, USA
+1 703 993 3682
{pmcandre,lnicklas,mpullen}@c4i.gmu.edu

Experience has demonstrated that practical deployment of SISO Military Scenario Definition Language (MSDL) and Coalition Battle Management Language (C-BML) at a scale beyond one C2 system connected to one simulation system requires a significant coordination effort. The NATO MSG-048 Technical Activity deployed, in its final experimentation, six national C2 systems and five national simulations, all them capable of simultaneous interoperation. In this environment, it became clear that synchronized operation of multiple systems under the control of human operators is extremely difficult if not impossible. Yet one of the greatest potential benefits of MSDL/C-BML systems is likely to come from interoperation of multiple C2 and simulation systems, for example for training and rehearsal within a multinational brigade. To address this problem, the GMU C4I Center has developed Status Monitor and Control, a Web-based coordination system that can give phased cues to all operators in the coalition and provide them with visibility regarding the state of all participating systems. With the addition of a central coordinating Web service and automated interfaces on the client systems, synchronization can be partly or fully automated. This paper describes the human and software interfaces of Status Monitor and Control and the logic of its internal coordination processes, as well as initial experience with its use.

1. Introduction

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 format 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].

The BML study group for the Simulation Interoperability Standards Organization (SISO) 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 that were developed and demonstrated by the members of NATO MSG-048 and MSG-085 [8,10,12,13,14,15]

References [16] and [17] describe the C2-Simulation environment developed for NATO Technical Activity MSG-048, *Coalition Battle Management Language*, which deployed the system of systems shown in Figure 1. This activity included six national C2 systems and 5 national simulations, a scale of interoperation not previously attempted. Experience in MSG-048 revealed 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 (SMC) 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.

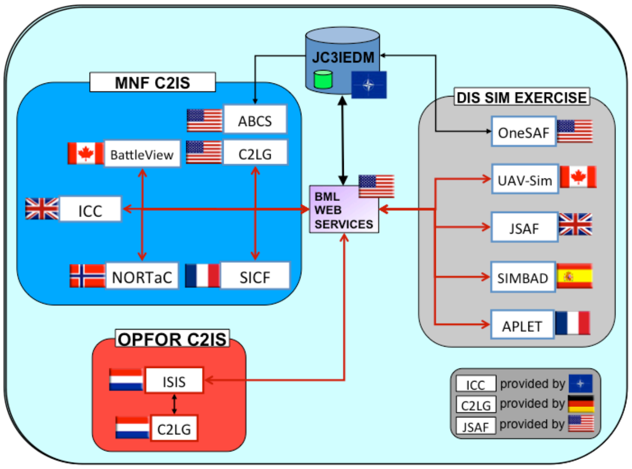

Figure 1: MSG-048 System Architecture

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. 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 2). This paper provides a detailed description of the implementation of such a service in the context of MSDL and C-BML coalitions as described above.

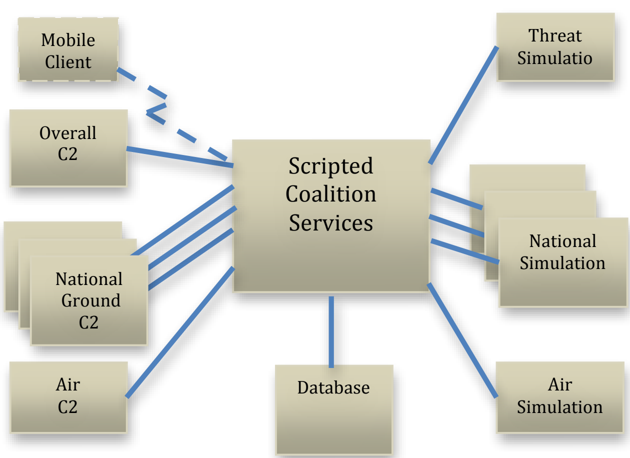

Figure 2: Coalition Services Architecture

2. Status Monitor and Control Functions

Functions of the SMC are summarized below, in three dimensions: coalition structure, C2/simulation asymmetry, and phases of the coalition event being monitored.

2.1 Coalition Structure

The basic problems addressed by SMC are:

- how to start, pause, and stop all coalition systems in a synchronized way
- how to keep operators and software for all coalition systems informed as to state of all participating systems

The process of start/pause/stop for all systems is under control of a *Master Controller*, who has the unique role to enter directions for participating systems.

Other participants in the coalition system of systems are:

- The *Coalition Server*, which accepts MSDL and C-BML XML documents from each participating client system, stores those documents according to logic specified by MSDL and C-BML, and forwards a copy to all clients that have subscribed for documents of that specific type.
- The *SMC Server* is part of the coalition server complex. It interacts with human users via webpages and also may initiate XML interactions with clients of the coalition server.
- The *C2 System Clients* are command and control systems that have been interfaced to C-BML and possibly also to MSDL and/or SMC.
- The *Simulation System Clients* are simulations that have been interfaced to C-BML, MSDL, and possibly also to SMC.
- *Client System Controllers* who operate the human control interfaces of C2 and simulation systems.
- *Observers* who can monitor all status but do not have any control role and thus do not provide status input.

2.2 Differences between (and among) C2 and Simulation Systems

C2 systems do not always display complete symmetry with simulation systems. This is because C2 systems are designed to exchange data about military operations with human users, while simulation systems are designed to generate such data under control of a model that is tailored to the particular need for simulation (e.g. training at various echelons, analysis of operational concepts, or mission rehearsal). The ways in which this asymmetry affects the SMC are:

- C2 systems may not require initialization from MSDL (although some of them do so).
- C2 systems may not have a concept of "stopped" or "paused" since real military operations normally take place continuously.
- Simulation systems may perform "checkpoint/restart," allowing a simulated activity to start running again using values captured at a checkpoint.

2.3 Master Controller Inputs

The Master Controller can set the desired coalition state providing direction to:

- *Stop*
- *Initialize* (i.e. start coalition MSDL generation and loading)
- *Run*
- *Pause*

2.4 Client Status Phases

SMC assumes the operation will take place in a sequence of well-understood phases in response to the Master Controller inputs. As noted above, not all systems will necessarily display each of these phases, but SMC is capable of displaying each phase for every client system. The phases are:

- *Stopped*
- *Initializing* (i.e., is participating in process to generate and ingest aggregated coalition MSDL)
- *Ready* (i.e., has completed generating and ingesting coalition MSDL)
- *Running*
- *Paused*

Client System Controllers indicate the current state of their C2 or simulation system by choosing a phase name from a dropdown menu. The names of phases are designed to be easy to change in SMC, by entering them in a file. In addition to the specified phases, the Master Controller and Client System Controllers have a free-text field available where they can enter a "comment" to indicate a status that does not exactly match a defined phase, for example "my system has crashed".

2.5 Login Issues

By design, html does not provide for a continuous connection between webpage and webserver; instead, each transaction takes place over a new TCP connection. As a result, the server has no way to know if the user has accidentally closed the browser. This can result in lockout of new sessions for the Client Controller of a particular C2 or simulation client. SMC includes two features to overcome this:

- A new SMC session may be started for any client, superseding its existing session.
- The Master Controller has the option of logging out any Client Controller or observer.

3. Status Monitor & Control Layout

The webpage layout for Master Controller and Client System Controllers are shown in figures 3 and 4. The general layout is:

- Heading, showing system which client controller is represented, or Master Controller
- Current order from Master Controller
- MDSL scenario name and status
- Number of observers logged in

Input areas:

- For Master Controller only, directions to "publish MSDL" "restart all" or "master logout" plus radio buttons to provide directions to *Stop, Initialize, Run, or Pause.*
- For Client Controllers, dropdown to provide a state input.
- Both Master and Client Controllers have text box for comments, as explained above.

Output area:

- For every Client Controller, the last status input provided (initially, *Stopped*)

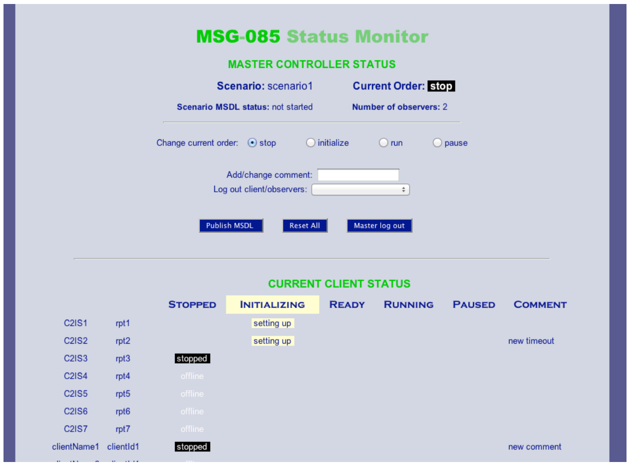

Figure 3: Example SMC Master Controller webpage

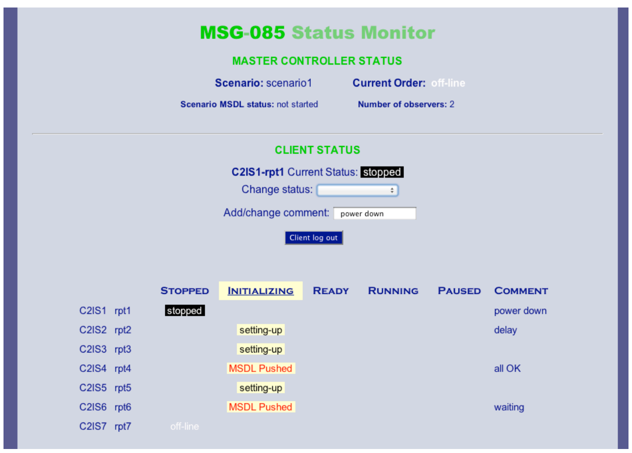

Figure 4: Example SMC Client Controller webpage

4. Automated interface using Web service

Initial experience using SMC indicates that human Client Controllers are very poor providers of status input. They have many distractions and therefore forget to post updated status for the system they are controlling. They may also miss the direction to change the status of their system, resulting in an unrealistic lag in simulated operations. A far better way for SMC to interact with client systems is to exchange directions via a Web service interface similar to the ones used for MSDL and C-BML. In fact, the same publish/subscribe capability can be applied. A prototype has been developed for an SMC Web service client that is able to do this, but as yet it has not been tested with actual systems. It should be noted that SMC can function with a mixture of human and Web service clients at the same time, which will facilitate transition to the Web service model.

We have created a RESTful web service that will allow software clients to manage their status in the status database as an alternative to using the client browser interface. To inform the master controller of a new client and its changing status, the following methods are available to clients:

1. c2SimAdd: adds a new client with current status
2. c2SimModClientStatus: modifies the status of an existing client
3. c2SimModComment: modifies the comment on an existing client
4. c2SimModSessionKey: modifies the session key on an existing client
5. c2SimModAutomated: modifies the automated flag on an existing client

In addition to updating client status with the web service, clients also can subscribe to a JMS topic to get status on the entire scenario. Using this 'master' status for the scenario, clients can determine when it is time to initialize (start their software, push any MSDL specific to their C2 or simulation system, receive MSDL from all other clients, process MSDL received), run, pause, and stop. Clients can receive the current status for a particular scenario by subscribing to a pre-defined topic and using the scenario identification string ('scenarioId'), as a message selector.

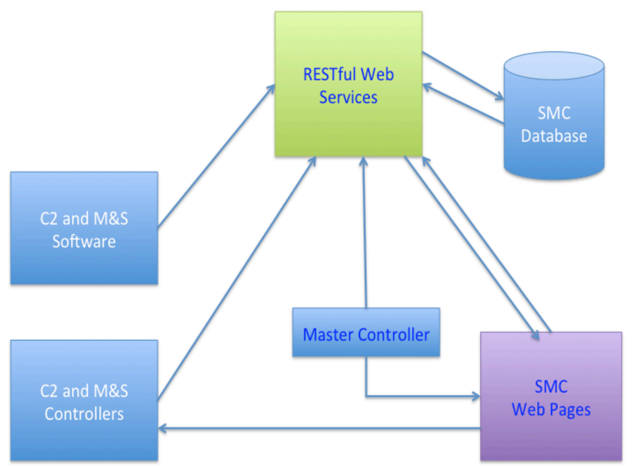

Figure 5: interaction control flow

5. SMC Experience

We have experience from one actual event, in addition to considerable testing experience.

5.1 I/ITSEC 2011

The actual event is the MSG-085 demonstration in the NATO booth at the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011 in Orlando, Florida. The experience of the "Demo Harness One" group, described in [18], was that SMC is useful and necessary. However, that experience also underscored the need for fully automated interfaces, based on the fact that all human controllers occasionally either missed an input (go to state) cue or missed providing a state change update to SMC.

The system interconnections involved in the I/ITSEC 2011 demonstration are shown in Figure 6. Note that Demo Harness One was physically distributed across the Internet with systems in Norway, United Kingdom, Virginia, and Florida. As a result the need for and benefits of SMC were strongly driven home. Even with the small number of systems demonstrated, coordination among controllers by voice became very difficult.

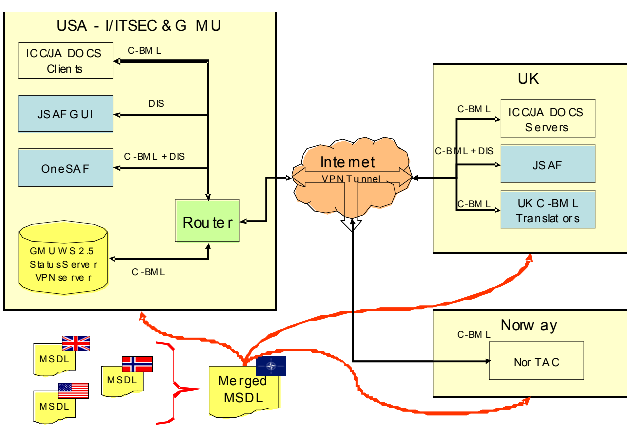

Figure 6: NATO MSG-085 I/ITSEC 2011 MSDL/C-BML Demonstration

5.2 Coalition Testing

Continued development of SMC has been the occasion of several system tests. Experience continues to indicate that SMC is useful and necessary. An automated interface is being developed for the US Army OneSAF system, to show the benefits of "human not in the loop" for status monitoring and control.

6. Conclusions

This paper has presented the nature of C2 and simulation interoperation using MSDL and C-BML for coalitions, and the experienced need for a coordinating mechanism, particularly when operations are distributed over networks. We have described here the design of a webpage-based coordinating mechanism, with optional automatic interface to client systems using a Web service interface. The system called Status Monitor and Control has been used in one actual event and considerable development testing. It is designed for flexible adaptation and is available as open source software from http://c4i.gmu.edu/OpenBML.

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

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

[18] Pullen, J., D. Corner, A. Brook, R. Wittman, O. Mevassvik, and Anders Alstad, "MSDL and C-BML Working Together for NATO MSG-085," IEEE Spring Simulation Interoperability Workshop, Orlando, FL, 2012

Author Biographies

PRISCILLA MCANDREWS is Laboratory Manager of the Networking and Simulation Laboratory, Webmaster of the C4I Center, and a lead developer in the team that developed MIST/C. She holds the MS degree in Computer Science and has many years experience in software development. She is the developer of the Status and Control Monitor described in this paper.

LISA NICKLAS is a member of the staff of the George Mason University C4I Center. She is the lead implementer for Scripted BML Server script execution engine. She developed the Web service interface for clients described in this paper.

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, and currently is co-lead of the Integrated Standard-Based Initialization Service Set (ISBSS) MSDL/C-BML convergence project.