Order and Report Schema Translation in WISE-SBML Server

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

Dr. Per Gustavsson
Saab AB
Skövde Sweden
+46 734 375 277
per.m.gustavsson@saabgroup.com

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

Keywords:
Schema Translation, Command and Control - Simulation Interoperability

The Coalition Battle Management Language (C-BML) as currently implemented requires that interoperating Command and Control (C2) and Simulation systems implement an interface to a Web service, using a clearly defined schema, to propagate Orders and Reports to participating systems. Continuing progress of C-BML, both in NATO MSG-085 and in the SISO standards process, 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, first enabled in the Scripted BML Server (SBML), 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. This paper explains how the schema translation works and issues encountered in its implementation, which includes an MSDL aggregation server.

1. Overview

Battle Management Language (BML) and its various 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 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].

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. The approach has generally followed the Lexical Grammar approach introduced by Schade and Hieb [4,5]. 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 [10-16]. 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.

References [15] and [16] describe the C2-Sim environment developed for NATO Technical Activity MSG-048, *Coalition Battle Management Language*. This activity included six national C2 systems and 5 national simulations, a scale of interoperation not previously attempted. Reference [17] 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). This paper provides a description of the implementation of such a service, in the context of MSDL and C-BML coalitions as described above, using the Widely Integrated Systems Environment (WISE) made available by Saab AB to MSG-085 through the GMU C4I Center as an extension of the Scripted BML Server (SBMLServer) [17-20].

Continuing progress of C-BML, both in NATO MSG-085 and in the SISO standards process, 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, first enabled in the Scripted BML Server (SBML), 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. This paper explains how the schema translation works and issues encountered in its implementation, which includes an MSDL aggregation server.

The remainder of this paper provides the history of SBMLServer, a description of WISE, a discussion of how WISE is used to enable an operationally-focused server (WISE-SBML), a description of how MSDL also is supported in this environment, and a conclusion that projects future use of BML based on MSDL, C-BML, and WISE-SBML. Much of this paper was originally produced by the same authors in [21].

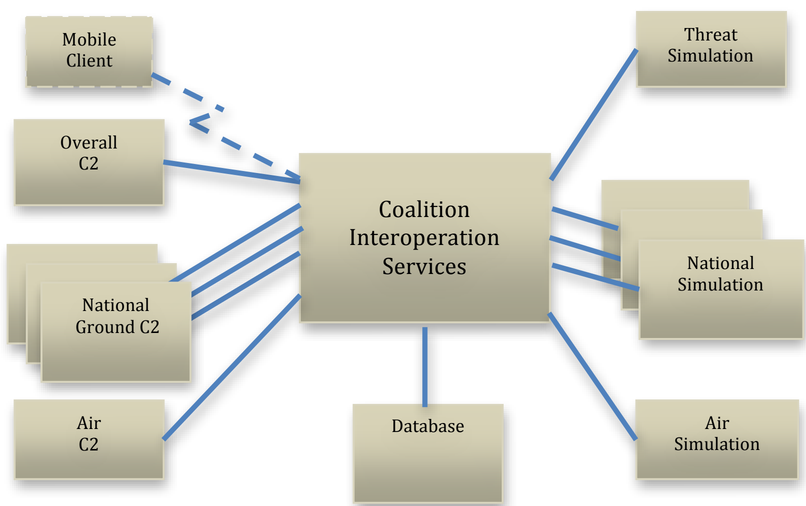

Figure 1: C2-SIm Coalition Services Architecture

2. Background: 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 [17] 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 for performance: 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. 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 [20].

Figure 3 shows the architecture of the WISE-SBML server. The "BMS" system in Figure 3 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 (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.

4. Rebuilding SBML using WISE

As shown in figure 3, 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 the power of the WISE approach and requires only drag-and-drop in the WISE Connectivity Designer. However, this leaves a sequence of steps to be programmed in conjunction with the specifics of the application (in the case at hand, C-

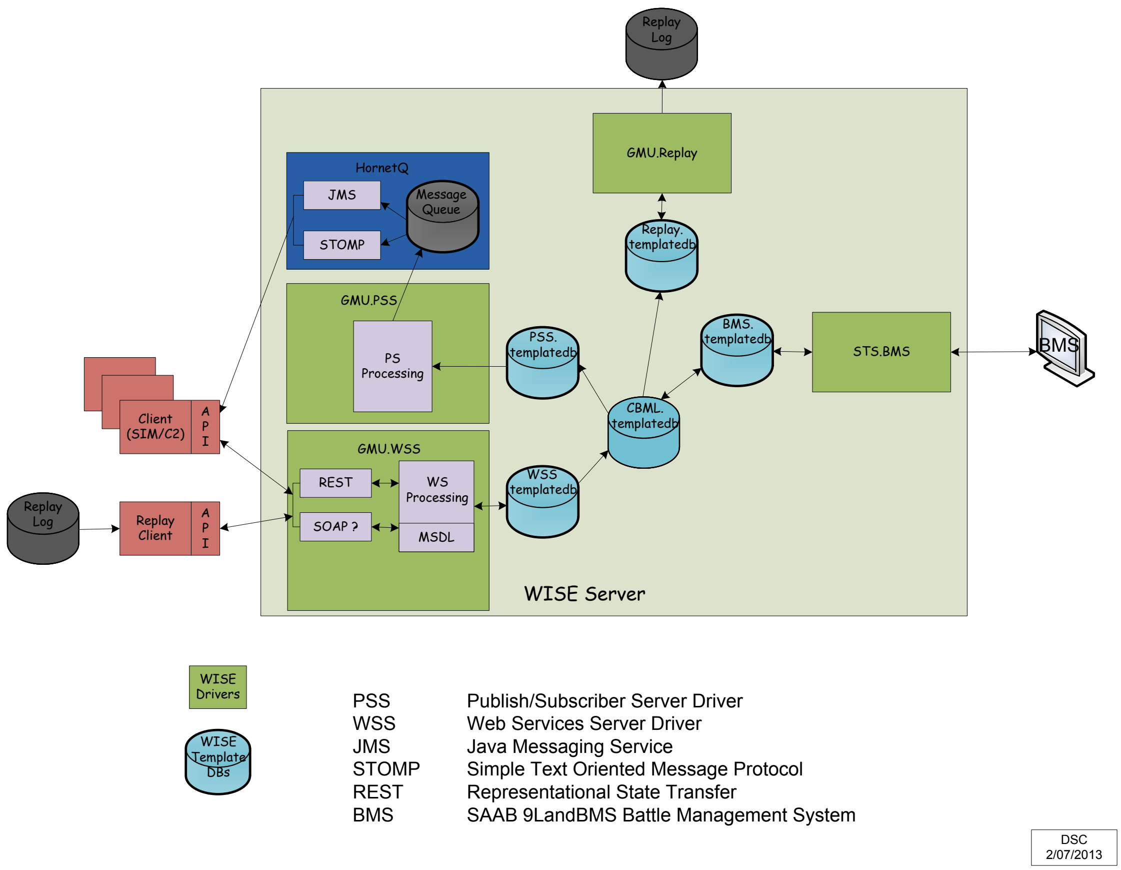

Figure 3. WISE/SBML Architecture

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 in accordance with the appropriate schema, using the internal data structure.

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 replace client. Initial tests indicate the resulting server has at least 10X the throughput of SBML, in translating mode.

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. References [21] and [22] provide more information on WISE-SBML, including its use in a distributed server system for MSG-085.

5. MSDL in WISE-SBML

Reference [23] 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 from 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 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 [19], 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 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 2. This approach has been re-implemented in WISE-SBML, using the WISE database to hold the Scenario File elements as they become available.

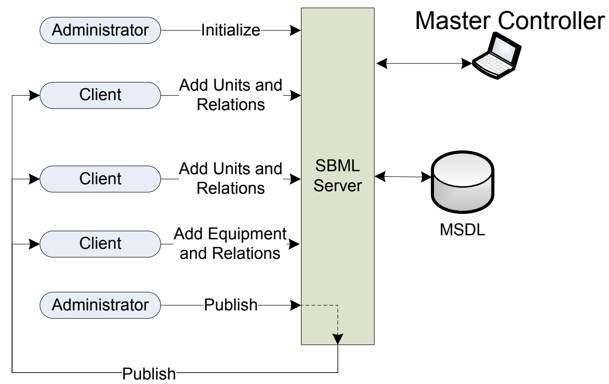

Figure 2. MSDL Server Operation for C2-Sim Clients

The MSG-085 activity provided a unique opportunity to explore the benefits and challenges of using MSDL as a basis for developing and sharing scenario information across multi-national simulation and C4I federations. Taking full advantage of this opportunity, a number of scenario development tools were used across the MSG-085 participants to construct slices of the overall scenario. Scenario slices were developed during the federation concept development phase and were allocated based on each participant's expertise and data production capabilities.

Scenario development tools included standalone MSDL production tools as well as C4I systems augmented for MSDL production. One individual MSDL slices were completed they were sent to the server for validation and merging. As a final step prior to federation execution The MSDL scenario slices were merged and proliferated to interested clients using the SBMLServer capabilities.

The final scenario, complete for execution, contained all of the required MSDL major elements and some optional elements to include:

- Scenario metadata;
- Playbox name and geographic extents;
- Side and force relationship data;
- Friendly and opposing force task organization data; and
- Tactical graphic and planning information.

Upon receipt each MSDL client then imported the scenario information, populated its internal data structures, and prepared for event execution.

As the MSG-085 coalition continues to explore the scenario development, dissemination, and loading process important lessons are being captured for use in guiding the evolution of the MSDL standard. Important lessons to date include:

- A common industry standards-based scenario data format (MSDL & XML) enable quick adoption and development of compliant import, export, and merge tools across the simulation and C4I system domains;
- A common format (MSDL) encourages the identification and tracking of scenario development agreements between scenario slice providers;
- A common industry standards-based format (MSDL & XML) enables use of widely available spreadsheets and XML tools for viewing scenario data;
- Data element extensions are critical for most MSDL uses. During MSG-085 it was shown that MSDL provides a reference element that can be used as a link to additional simulation specific data files such as a reference to one or more C-BML document while remaining compliant with the MSDL schema. It is expected that depending on the simulation this referenced data may also include unit and/or equipment-based supplies; equipment crew information; NATO stock number information, etc. and that these referenced files may become standalone standards or fully integrated into the MSDL schema over time.

6. Conclusions

Work in C2-simulation interoperation, using emerging SISO standards MSDL and C-BML, 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. However, a natural aspect of this flowering development is the existence of multiple, related schemata among the various C2-Sim clients. Availability of translating services supported by a robust, high-performance platform can serve as an enabler to gain critical experience through expanded application and interoperation of various national systems, as exemplified in MSG-085.

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] 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 Modeling and Simulation Symposium, Stockholm, Sweden, October 2012

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

[8] 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

[9] 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

[10] 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

[11] 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

[12] 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
[13] 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
[14] 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
[15] 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
[16] 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
[17] 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
[18] Pullen, J. and K. Heffner, "Supporting Coalition Battle Management Language Experiments with Scripted Web Services", NATO 2009 Modeling and Simulation Symposium, Brussels, Belgium, 2009
[19] 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
[20] 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
[21] Pullen, J., D. Corner, R. Wittman, A. Brook, P. Gustavsson, U. Schade and T. Remmersmann, "Multi-Schema and Multi-Server Advances for C2-Simulation Interoperation in MSG-085," NATO Modeling and Simulation Symposium 2013 (submitted)
[22] Pullen, J., D. Corner, T. Remmersmann and I. Trautwein, "Linked Heterogenous BML Servers in NATO MSG-085," IEEE Fall 2013 Simulation Interoperability Workshop, Orlando, FL, 2013

Author Biographies

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. PER GUSTAVSSON is Principle Research Scientist in Command and Control Science at Swedish National Defence College and holds a similar position at Saab. He is an affiliate faculty member of the GMU C4I Center. He holds a PhD in Engineering and simulation from De Montfort University Leicester, UK. Gustavsson is acting co-chair for C-BML and MSDL standardization within SISO and is the Principal Investigator for C2-SIM Interoperation within Saab.

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.