C-BML Infrastructure

Dr. J. Mark Pullen
George Mason University C4I Center
Fairfax, VA 22030
USA

mpullen@c4i.gmu.edu

ABSTRACT

This paper is one of a coordinated set prepared for a NATO Modelling and Simulation Group Lecture Series in Command and Control – Simulation Interoperability (C2SIM). This paper describes the design of messaging and control infrastructure that has been used effectively for C2SIM.

1.0 INTRODUCTION

This paper was prepared to support a session in the NATO Modelling and Simulation Group Lecture Series in Command and Control – Simulation Interoperability (C2SIM). The session describes the design of messaging infrastructure that has been used effectively for working C2SIM systems. It consists of describes the various supporting software used with C2SIM, and in particular, the server. We'll start with basics: how the system works – then go on to details about the role of each system component.

2.0 BML ARCHITECTURE

We started work on BML in 2003 with the goal of making it possible for C2 and simulation systems to exchange information according to an open standard [38]. This requires that the information is presented with no ambiguity, because the systems involved must have precise inputs or you get "garbage in, garbage out." The ultimate goal is that the semantics (meaning) of the information is interpreted the same by every system participating in the C2SIM system of systems, which we call a "coalition" in the same sense that a HLA system of simulation systems is called a "federation." The analogy to a military coalition is a good one, since the systems are cooperating because they have agreed to do so, not because they are organized as part of a larger national force.

2.1 Generalized BML Architecture

Figure 1 gives an overview of how C2SIM coalitions work. There must be at least one C2 system; possibly, many; and the C2 systems can communicate via mechanisms such as the JC3IEDM or simply via BML. Similarly, there must be at least one simulation system; possible, many; and they may communicate among themselves using techniques such as DIS or HLA; or only through BML [23].

Though it is possible for C2 and simulation systems to communicate with each other directly via BML messages, it is customary to pass the messages through a server. The server implements communication protocols (web services in every case we're aware of) and provides access to a data repository, which stores copies of tasking and reporting transactions as well as providing for initialization and synchronization [30].

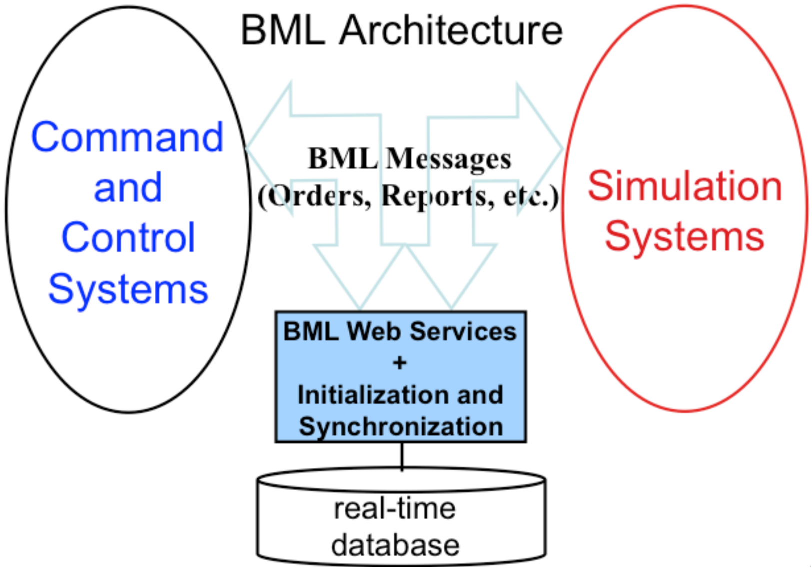

Figure 1. BML Architecture

2.2 MSG-048 BML Architecture

In 2009, MSG-048 assembled the first major BML coalition of systems, as shown in Figure 2. It had 6 national C2 systems, 5 national simulations, a live opposing force, and several pieces of software to support communications, including the server that my center provided.

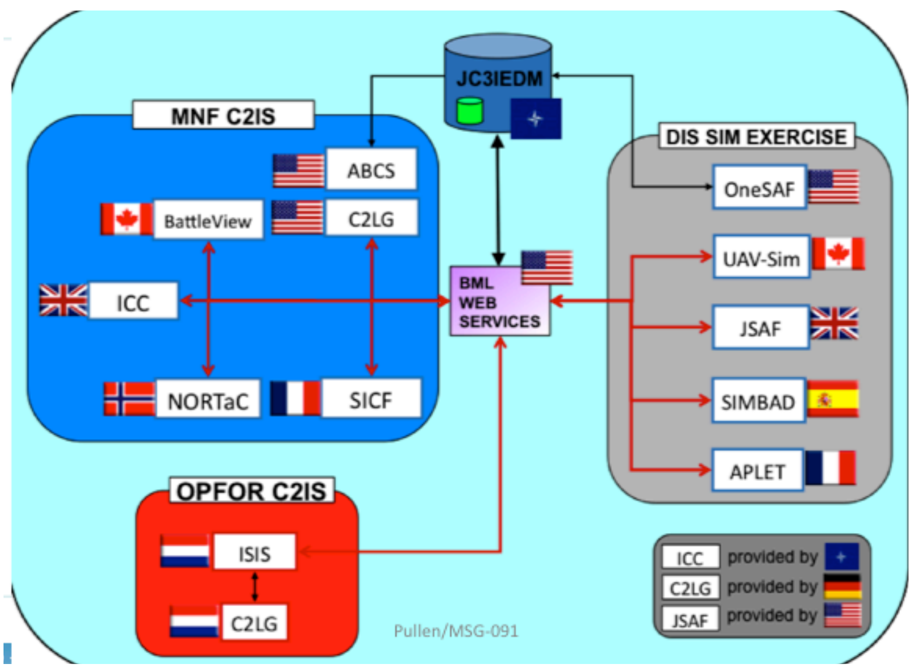

Figure 2. MSG-048 2009 Architecture

2.2 MSG-085 BML Architecture

By 2013 the BML system had gotten more robust – supported by industry and distributed over the Internet. And it was used to demonstrate the operational effectiveness of distributed, BML-based use of simulation to support a collaborative planning process as shown in Figure 3.

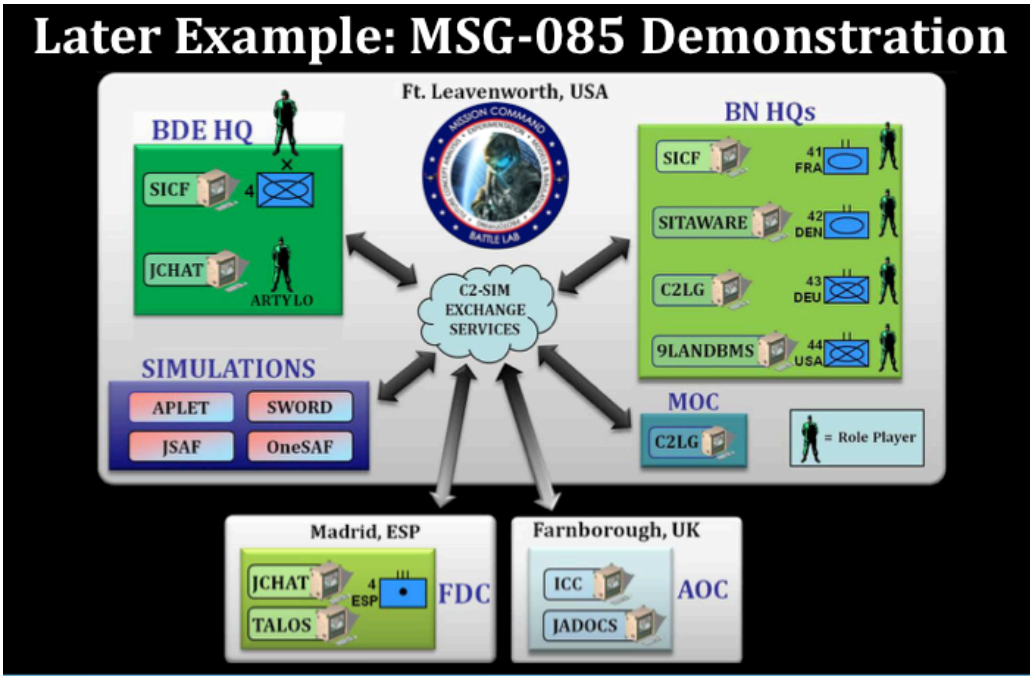

Figure 3. MSG-085 Final Demonstration Architecture

3.0 SERVER OPERATIONS

The servers we use are connected by "web service" technologies. This works a lot like using the web browser on your own computer, but the operations are started by C2 and simulation software, not by human users. The service allows information to be "pushed" in and "pulled" out. The original protocol for this was called SOAP but now we use a more efficient protocol called REST that avoids unnecessary steps.

The server also has to send a copy of each BML message to those systems that need to it. This is done with a protocol called STOMP, the Streaming Text Oriented Messaging Protocol. Systems "subscribe" for Topics of interest. The BML server uses STOMP for forward a copy to each subscribing system [27].

3.1 BML Message Representation

The data is coded in a forward called Extensible Markup Language (XML). XML uses "tags" to describe the data it contains. A "schema" provides a definition of all possible tags and the order and groupings in which they are allowed. There may also be a description of the schema in the Web Service Description Language (WSDL). The pattern of the data is called a "tree" because, starting at the root tag, it always branches out – never connects back into itself.

Figure 4 shows a simple BML report document, coded in XML. You can see that it can be read by a human if necessary, but it's not a very pleasant process. For example, the unit reporting is "1-22" and it is reporting something about a Friendly unit (in fact, about itself, because it also is the “executer”). XML is often called “verbose” because the tags take up so much space in the message. Figure 5 shows the remainder of the example report. . If you look closely you can see that no tag refers back to an earlier part of the tree – they all close off as separate “branches”.


```
•<?xml version="1.0" encoding="UTF-8"?>
•<BMLReport
• xmlns:jc3iedm="urn:int:nato:standard:mip:jc3iedm:3.1a:oo:2.0"
• xmlns:bml="http://netlab.gmu.edu/IBML"
• xmlns:msdl="http://netlab.gmu.edu/JBML/MSDL">
•  <Report>
•    <CategoryOfReport>StatusReport</CategoryOfReport>
•    <TypeOfReport>GeneralStatusReport</TypeOfReport>
•    <StatusReport>
•      <GeneralStatusReport>
•        <ReporterWho>
•          <bml:UnitID>1-22</bml:UnitID>
•        </ReporterWho>
•        <Hostility>FR</Hostility>
•        <Executer>
•          <bml:Taskee>
•            <bml:UnitID>1-22</bml:UnitID>
•          </bml:Taskee>
•        </Executer>
```


11

Figure 4. Example XML Report First Part

3.2 XML Schema

Figure 6 shows how the content and structure of an XML document is dictated by its schema. The schema tells what data elements are possible, whether they are mandatory, and how they are grouped. Each “Type” specifies data order and format for a “chunk” of the XML. The schema itself can be expressed as an XML document of type “XSD”. A very important part of the C-BML standard is the collection of schema files which define the all the possible elements and groups of elements in Orders and Report.

```
<OpStatus>MOPS</OpStatus>
        <WhereLocation>
              <bml:GDC>
                <bml:Latitude>33.424079</bml:Latitude>
                <bml:Longitude>44.682716</bml:Longitude>
                <bml:ElevationAGL>2.054</bml:ElevationAGL>
              </bml:GDC>
        </WhereLocation>
        <When>20070101000000.000</When>
        <ReportID>506</ReportID>
        <Credibility>
          <bml:Source>HUMINT</bml:Source>
          <bml:Reliability>A</bml:Reliability>
          <bml:Certainty>RPTFCT</bml:Certainty>
        </Credibility>
      </GeneralStatusReport>
    </StatusReport>
  </Report>
</BMLReport>
```

12

Figure 5. Remainder of Example XML Report

Schema is a representation of the organization and format that is allowable for a given XML file, for example this slice:

```
<xs:complexType name="ReportHeaderLightType">
 <xs:sequence>
  <xs:element name="ReporterWho" type="cbml:ReporterWhoType"/>
  <xs:element name="ReportingData" type="cbml:ReportingDataType">
   <xs:choice>
    <xs:element name="AbsoluteReportedWhen"
      type="cbml:AbsoluteReportedWhenLightType"/>
    <xs:element name="RelativeReportedWhen"
      type="cbml:RelativeReportedWhenLightType"/>
   </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Figure 6. XML Schema

4.0 C2SIM COALITION SYSTEM COMPONENTS

Here are the types of systems that can be part of a C2SIM coalition system-of-systems [28]. The next section of the slides describes each of these in more detail:

- Command and Control systems
- Simulation systems
- Servers

- Graphic User Interfaces
- Status monitoring and control

Figure 7 shows how all those type of systems can be interconnected. Not shown is the monitoring/control station that can be used to coordinate the whole coalition:

- C2 Systems
- Simulation systems
- Servers (the central block could be comprised of more than one server)
- Graphical user interfaces (GUIs)

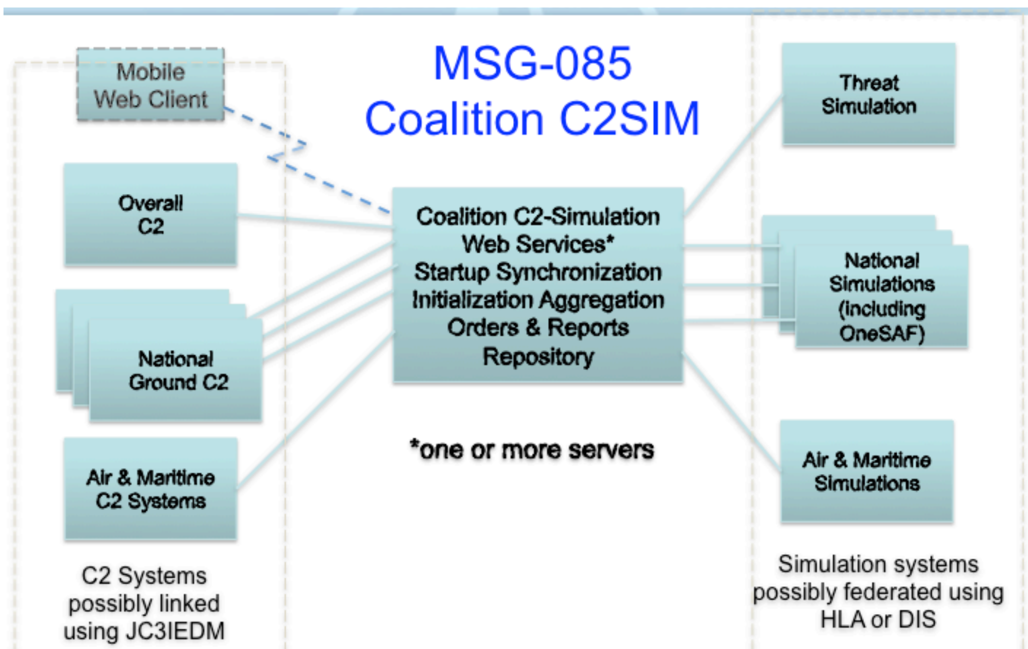

Figure 7. MSG-085 Coalition C2SIM

4.1 C2SIM Client Functions

Another paper provides details of interfacing clients. From the server's perspective, the client's role is straightforward. Client C2 systems provide orders for the coalition and in return they get back status in the form of reports. When we interface a C2 system to the C2SIM server, what we do is:

- Send orders to the server as they are generated
- Subscribe to and receive reports as they come in from the simulations

We want to use operational C2 systems to remove the need for an artificial simulation user interface and "train as you fight" – but we must be careful to:

- Make it very clear when the C2 system is displaying simulated status
- Provide an easy way for the users to start, stop and pause the simulation

Client simulation systems receive orders from the coalition and produce reports showing status of simulated objects. Depending on the simulation, the reports could be for individual platforms (like tanks and airplanes) or aggregated objects (like platoons). What the simulation has to do is:

- Subscribe to appropriate source of orders, receive them and execute them
- Produce report per object at designated time interval and send to server
- Might also produce report when some event happen, such as start of attack
- Be controllable for start/stop/pause

4.2 C2SIM Server Functions

Now let's consider what servers do. The BML server has three main functions [27]:

- Accept and store BML and MSDL documents incoming from client C2 and Simulation systems
- Accept and remember subscriptions
- Publish the incoming documents to clients that have subscribed to Topic they contain; also respond to requests for specific documents by ID

There are other functions that servers may perform:

- Handle "namespace" that can be used in XML to distinguish among multiple sources
- Verify that an input document conforms to the schema being used (if it doesn't the server may get wrong results – but checking sows down the server)
- Filtering data to avoid unwanted outputs

And more functions:

- Logging inputs, with time stamps, for review
- Replaying the log to recreate the effect of the input stream
- Working with one or more other servers to distribute load (this can, for example, reduce overall network traffic)

4.2.1 Distributed Server System

Figure 8 shows an example of two servers cooperating in MSG-085 final demonstration. The FKIE server supported French and German C2 and simulation clients; the WISE-SBML server supported US and UK clients and translated among the 3 schemas used by different clients in the demonstration [30].

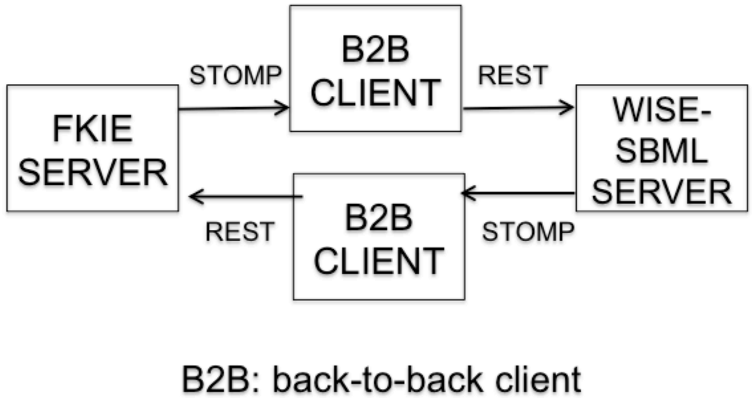

Figure 8. Distributed servers connected using back-to-back clients

4.2.2 Server Schema Translation

This brings us to the topic of translation. As C-BML evolved, various C2 and simulation clients were interfaced under different schema definitions. Even though they used essentially the same information, it was not all in the same format. To enable them to interoperate, we had to build a server that could translate among the schemas used. Of course we couldn't do that if they used different information – it was possible when only the format was different. Using a translating server allowed us to assemble a much larger system of systems in MSG-085 [27].

4.2.3 Server Support for MSDL

A server can also be useful to support MSDL. MSDL scenario files are used at the beginning of simulation, to initialize the system based on the location and state of units .If multiple simulations are brought together on short notice, it is necessary to combine their scenario files so they can all deal with each other's simulated objects. This could be done manually, using a tool; or, the server can combine the scenario files and publish the result back to all participating simulations. As we move toward plug-and-play C2SIM, doing it that way will eliminate a manual step [24].

Figure 9 shows how aggregation worked where we implemented it for testing. The final scenario file is not produced until the last simulation submits its MSDL file. One way to detect this is that human operator confirms that all participating systems have connected.

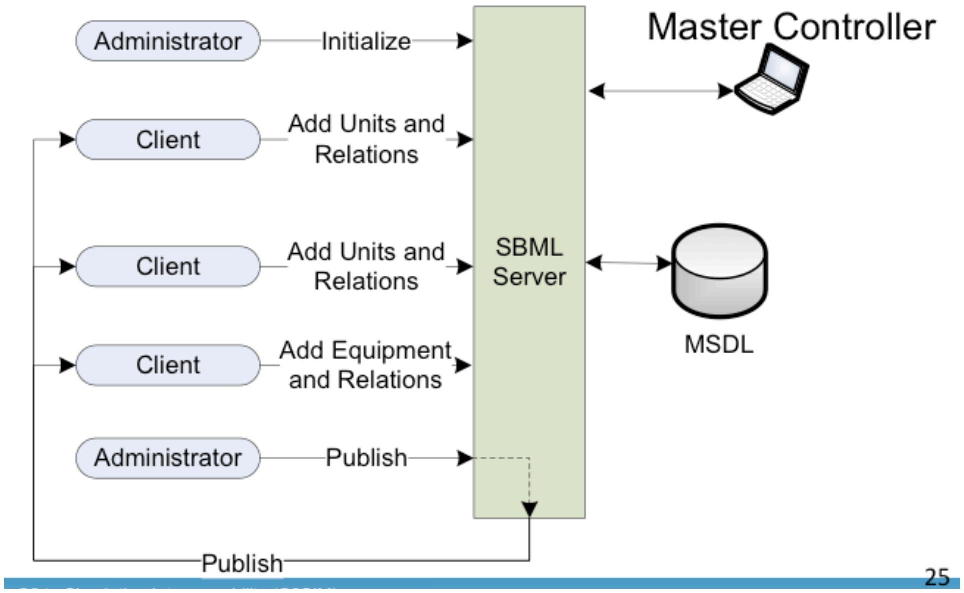

Figure 9. MSDL Server Operation

4.3 Other Software Supporting C2SIM

Some other software systems also are useful in assembling C2SIM coalitions.

4.3.1 General-Purpose Graphical User Interface

For system testing, it is very useful to have a graphical user interface (GUI) that can assemble and submit order and reports, and also receive and display orders and reports from a server. This is a very flexible system component because it can receive a document, edit it, and resubmit to the server. It also can stand in for a C2 system as a "surrogate" when you don't have the C2 system you need.

Such a GUI is capable of many functions:

- Editing a C-BML or MSDL document
- Merging MSDL documents
- Serialization of document
- Grammar validation of document
- Schema validation of XML document
- Auto-configuration to schemaPulling a document
- Pushing a document
- Subscription to server Topics
- Retrieving latest reports
- C2 capability

- Displaying maps with overlays
- Geolocation entry from maps

The first such GUI was built by FGAN (now Fraunhofer FKIE) and has seen a lot of use, starting with MSG-048 [12]. Because the FKIE GUI could not be released without restrictions, GMU C4I Center created an open source version, advised by FKIE. It is available from our webpage. The two versions have diverged somewhat – FKIE's GUI now has added functionality for dealing with robotic systems, where the GMU version had features added to work with SBML. Figure 10 shows a screenshot from the FKIE GUI. Figure 11 is a screen shot from the GMU GUI. Note the use of open-source OpenMap software.

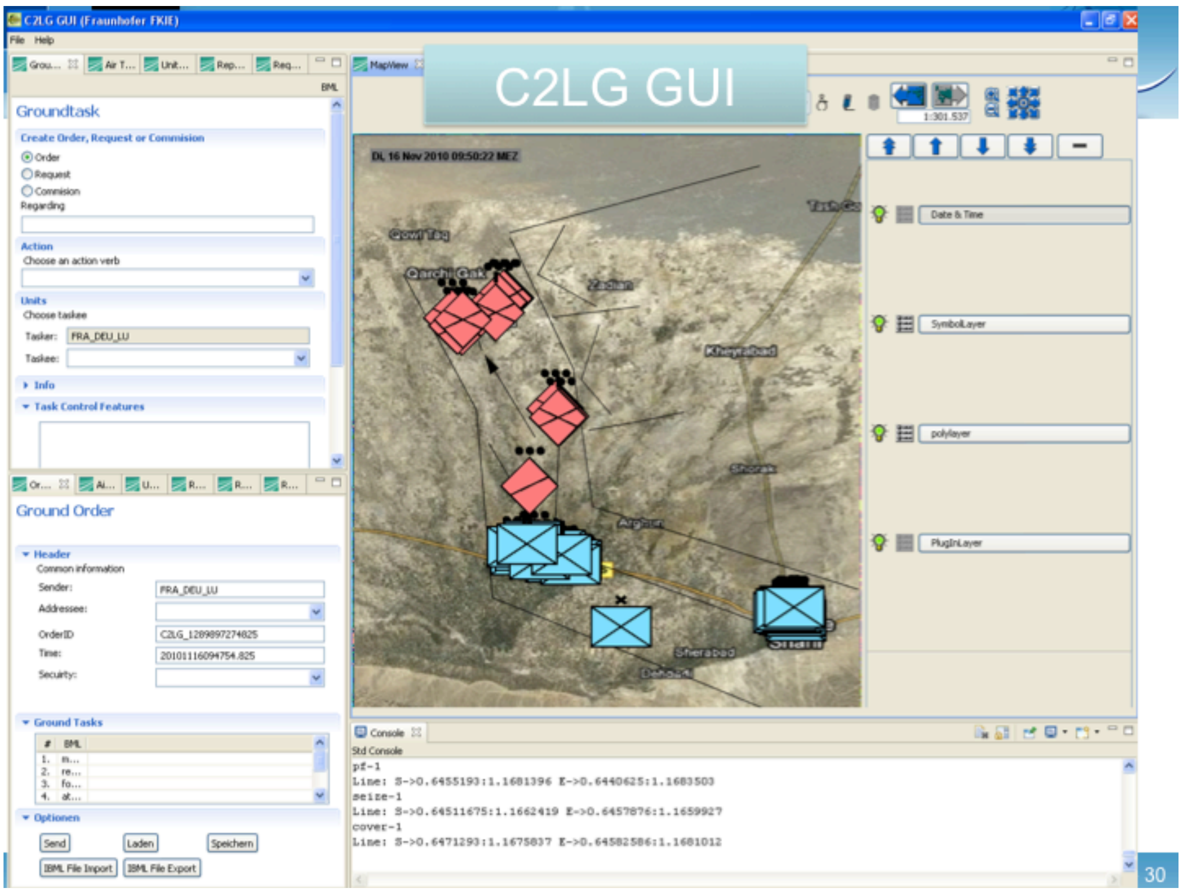

Figure 10. FKIE GUI

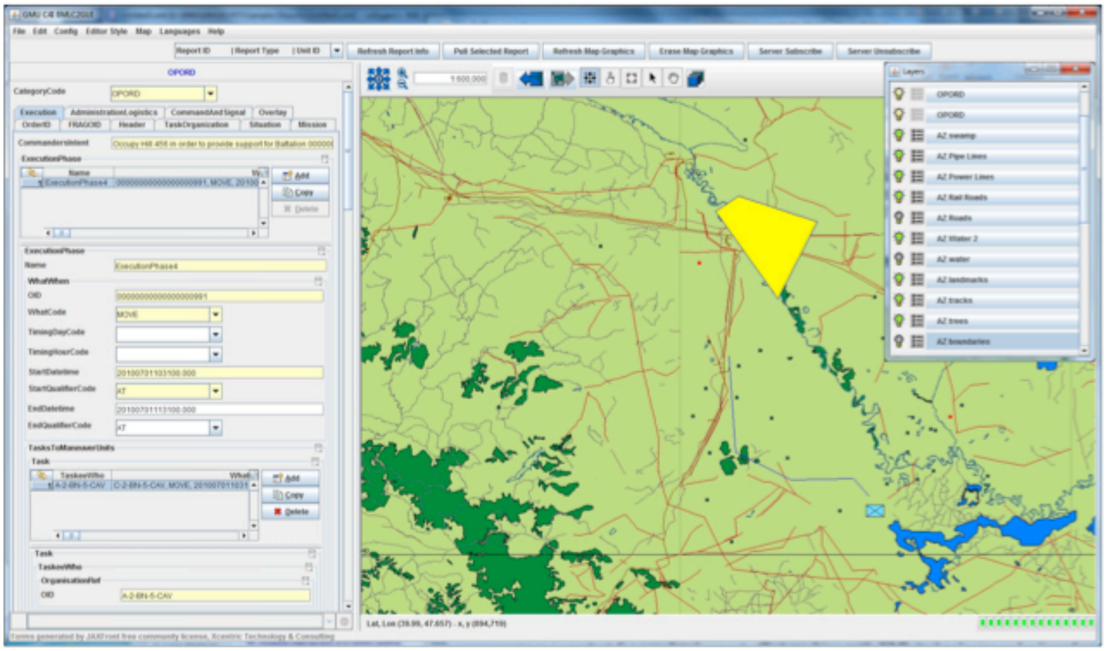

Figure 11. GMU BML C2 GUI

4.3.2 Virtual Private Network Software

One other supporting software component we have found very useful is Virtual Private Network (VPN), consisting of a central server and a client on each participating computer. The VPN sits on an underlying network, typically one that uses the Internet Protocol (IP), which might be a private network like the US DoD SIPRNET or it might be the public Internet.

Programs that connect to the VPN don't have access to the underlying network. They see a "virtual" network that exists among the computers where the VPN client is deployed and configured. Information sent among VPN clients is encrypted, although typically not at a level required for classified information. Thus the information the participating machines exchange is "private".

MSG-085 set up a VPN, using Open VPN free software. This gave us a way to work together without worrying about Internet hackers. We did a lot of testing this way and ultimately used two different VPN segments over cellular Internet to connect UK and Spain systems during the Final Demonstration. The MSG-85 VPN is still running 24x7 today and has a C2SIM server available. Ultimately, we expect to set up "simulation as a service" to support testing of C2 systems in a C2SIM environment.

5.0 CONCLUSION

Following the C2SIM architecture developed for MSG-048 and MSG-085, a functional messaging infrastructure is essential to effective operation. The whole C2SIM coalition must function smoothly as a system of systems. The key element here is the server (or distributed system of servers), which must implement, and be coupled to, a schema that meets the needs of the C2 and simulation clients for information exchange, including MSDL scenarios files. In cases where the clients have been interfaced with different schemata that are semantically compatible, a translating server can provide for coalition interoperation. Other useful components for the system-of-systems include general-purpose GUIs and VPN software.

6.0 BIBLIOGRAPHY

[1] Abbott, J., C. Blaise, T. Chase, J. Covelli, M. Fraka, F. Gagnon, K. Gupton, P. Gustavsson, K. Peplow, D. Prochnow, R. Wittman, "MSDL the Road to Balloting," IEEE Fall Simulation Interoperability Workshop, Orlando FL, 2007

[2] 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

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

[5] Burland, B., J. Hyndøy, and J. Ruth, "Incorporating C2--Simulation Interoperability Services Into an Operational Command Post," International Command And Control Research and Technology Symposium 2014, Alexandria, VA

[6] 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 2001, Orlando, FL

[7] 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

[8] Diallo, S., R. Gore and A. Barraco, "Integrating CPOF, JSAF and ONESAF through CBMS," International Command and Control Research and Technology Symposium 2013, Alexandria, VA

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

[10] Gautreau, B., L. Khimeche, J. Martinet, E. Pedersen, J. Lillesoe, D. liberg, T. Remmersmann, D. Muniz, T. Serrano, N. Dereus, H. Henderson., "Lessons Learned from NMSG-085 CIG Land Operation Demonstration," IEEE Spring Simulation Interoperability Workshop, San Diego, CA, 2013

[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] Heffner, K., L. Khimeche and J. Pullen, "MSG-048 Technical Activity Experimentation to Evaluate the Applicability of a Coalition Battle Management Language in NATO," NATO Modelling and Symposium 2010, Utrecht, Netherlands

[13] 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

[14] Hieb, M., W. Sudnikovich, A. Tolk and J. Pullen, "Developing Battle Management Language into a Web

Service," IEEE Spring Simulation Interoperability Workshop, Orlando FL, 2004

[15] 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

[16] 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

[17] Lacy, L., R. Byrd, "Supporting Multiple Equipment Enumeration Approaches in the Military Scenario Description Language (MSDL)," IEEE Spring Simulation Interoperability Workshop, Orlando FL, 2009

[18] 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

[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] Moffat, James: "Adapting Modeling & Simulation for Network Enabled Operations", DoD Command and Control Research Program (CCRP), March 2011

[21] NATO Science and Technology Organization, Modelling & Simulation Group 085: Standardization for C2-Simulation Interoperation NMSG-085 Final Report, ISBN 978-92-837-2017-1, STO-TR-MSG-085, June 2015.

[22] 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

[23] 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

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

[25] 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

[26] 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

[27] 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 Modelling and Simulation Symposium 2013, Sydney, Australia

[28] Pullen, J. and L. Khimeche, "Advances in Systems and Technologies Toward Interoperating Operational Military C2 and Simulation Systems," International Command And Control Research and Technology

Symposium 2014, Alexandria, VA

[29] Pullen, J., L. Khimeche, R. Wittman, B. Burland, J. Ruth, J. Hyndoy, "Coalition C2-Simulation History and Status," NATO Modelling and Simulation Symposium 2014, Washington DC.

[30] Pullen, J., L. Khimeche, X. Cuneo, U. Schade, and T. Remmersmann, Linking C2-Simulation Interoperation Servers to Form Distributed Server Systems, International Command and Control Research and Technology Symposium 2015

[31] Pullen, L., S. Levine, K. Heffner, L. Khimeche, U. Schade, M. Frey, N. de Reus., N. Le Grand, P. de Krom, O. Mevassvik, A. Alstad, R. Gomez-Veiga, S. Galan Cubero, A. Brook, "Integrating National C2 and Simulation Systems for BML Experimentation," 2010 Euro Simulation Interoperability Workshop, IEEE Euro Simulation Interoperability Workshop, Ottawa, Canada, 2006

[32] 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

[33] 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

[34] Simulation Interoperability Standards Organization, Product Nomination for Command and Control – Simulation Interoperation, 2014

[35] Simulation Interoperability Standards Organization, Standard for: Coalition Battle Management Language (C-BML), 2014

[36] Simulation Interoperability Standards Organization, Standard for: Military Scenario Definition Language (MSDL), 2008

[37] Savasan, H., A. Caglayan, F. Hildiz, U. Schade, B. Haarmann, O. Mevassvik, G. Sletten, K. Heffner "Towards a Maritime Domain Extension to Coalition Battle Management Language: Initial Findings and Way Forward", IEEE Spring 2013 Simulation Interoperability Workshop, San Diego, CA, 2013

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

[39] 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

[40] Surdu, J., K. Galvin, W. Lam, D. Perme, J. Montgomery, "Simulation Interoperability Standards Organization (SISO) Military Scenario Definition Language (MSDL) Progress Report," June 2005

[41] Tolk, A., R. Wittman, R., "Engineering Principles of Combat Modeling and Distributed Simulation, Chapter 24," John Wiley & Sons, Inc., Hoboken, NJ, 2012

[42] US Army Doctrine Reference Publication (ADRP) 5-0, The Operations Process, 17 May 2012

[43] US Army Field Manual (FM) 6-0, Command and Staff Organization and Operations, May 2014

[44] Wittman, R., "OneSAF as an In-Stride Mission Command Asset," International Command And Control Research and Technology Symposium 2014, Alexandria, VA