Creating a BML Software Infrastructure

Dr. J. Mark Pullen
Director, C4I Center
George Mason University
mpullen@c4i.gmu.edu

http://c4i.gmu.edu/BML
Overview of Presentation

• Essential elements for BML
• The IBML Schema
• Developing the Scripted BML Server
• Implementing publish/subscribe
• Other software essential for distributed development
Essential Elements for BML

• C2 Systems
• Simulation systems
• BML schema
• BML server
• Network infrastructure
Terminology

• What do you call the system of systems that interoperate via BML?
• HLA calls their assembly a “federation”
  – Implies the systems are like member states
• For BML I propose “coalition”
  – Reflects the military application
  – And the relationship among systems
  – They come together voluntarily, for a specific purpose, under mutually agreed leadership
• I will use that terminology in this talk
* database could be distributed via replication,
allowing a coalition to employ multiple such clusters
MSG-048 Architecture

[Diagram showing the relationships between various components including MNF C2IS, ABBCS, C2LG, BattleView, ICC, NORTaC, SICF, OPFOR C2IS, ISIS, C2LG, BML WEB SERVICES, JC3IEDM, DIS SIM EXERCISE, OneSAF, UAV-Sim, JSAF, SIMBAD, APLET, ICC, C2LG, JSAF provided by flags from different countries.]
C2 and Simulation Systems

• These are the main focus of activity
  – Others will speak about them at length

• Important characteristics from infrastructure view
  – Must be possible to interface them to comply with BML schema and selected network protocols
  – Must be possible to control them in coalition context
  – C2 systems must be able to accept reports at system designated rate
  – Simulation systems but be able to throttle back to system designated reporting rate
    • A particular problem if faster than real time
BML Schema

• Essential to define information that can be exchanged
  – Expressed as XML Schema Definition (XSD)
  – Establishes structure of information
    • Basically, a tree – like any XML
  – And “namespace” of tags that identify it

• Very important to have this defined first
  – The server and all C2/simulation interfaces must implement it
  – Late binding of schema nearly derailed MSG-048
Layers in BML

BML as implemented in MSG-048 has three layers:

Top: Language layer: primary exposed service which implements tagset for operational functions (tasking, reporting, etc.)
  • JBML showed how to structure this using C2 Lexical Grammar for unambiguous parsing

Middle: Business object layer: internal building blocks at the level of “who/what/when/where/why”
  • SISO is working on a standard for this

Bottom: Database layer: JC3IEDM
  • Standardized by MIP
BML Layers

- Orders/Reports
- Language
- Business Object
- Five Ws
- JC3IEDM
- Database
BML Schema for Task
(top layer)

<xs:complexType name="GroundTaskType">
  <xs:sequence>
    <xs:element name="TaskeeWho" type="WhoType"/>
    <xs:element name="What" type="WhatType"/>
    <xs:element name="Where" type="WhereType"/>
    <xs:element name="StartWhen" type="WhenType"/>
    <xs:element name="EndWhen" type="WhenType" minOccurs="0"/>
    <xs:element name="Affected" type="AffectedType" minOccurs="0"/>
    <xs:element name="Why" type="WhyType" minOccurs="0"/>
    <xs:element name="TaskControlMeasures"
      type="TaskControlMeasuresType" minOccurs="0"/>
    <xs:element name="TaskLabel" type="LabelType"/>
  </xs:sequence>
</xsd:complexType>
BML Schema for *WhoType*
(middle layer)

```xml
<xs:complexType name="WhoType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="UnitID" type="jc3iedm:OIDType"/>
      <xs:element name="NameText"
        type="jc3iedm:Text100XmlType"/>
    </xs:choice>
    <xs:element name="Equipment"
      type="jc3iedm:Text100XmlType"
      maxOccurs="unbounded" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```
The IBML Schema

• By 2008 there were multiple flavors of BML
• US Army Geospatial Center (AGC) sponsored project produce a consolidated schema
  – Combined best of JBML, geoBML, and Army OPORD
  – MSG-048 used and refined IBML
• US Army G6 sponsored an architecture effort that further refined IBML (with MSG-048 inputs)
  – Results documented in detail
  – Available on GMU C4I Center website
  – Includes a mechanism for schema maintenance
  – When SISO standardizes a Business Object layer, the Army architecture could be harmonized with it
  – SISO plans call for standardizing top layer later
  – IBML is available for use now
• IBML OPORD gives a good start on NATO OPORD schema
Role of the BML Server

• Conceptually the server plays a very simple role
  – Accept BML Orders and Reports as XML documents pushed by clients
  – Reproduce those documents on demand, based on OrderID or ReportID
  – Provide supporting services such as initialization and summarizing status
    • For example, all ReportID in a given time window

• Issues for server
  – Performance
  – Polling model versus publish/subscribe
  – Auxiliary functions such as logging
About BML Server Performance

• BML Clients will always be able to overwhelm even the most powerful server
  – There are multiple clients and they can work from memory rather than database

• The appropriate question is whether the server is able to keep up with a realistic rate of orders and reports

• MSG-048 SME advised that a one minute interval would be realistic
  – After all clients implemented this worked well
  – Server used never had observable backlog

• Larger simulations will stress servers more
  – Replicating server could help but in the end the database is the bottleneck
  – Might need to invest in more powerful database
  – Some limitation on Report rate always will be necessary
Scripted BML Server
Why Scripted BML Server

• BML continues to grow and evolve
• JC3IEDM changes too, if more slowly
• GMU C4I Center provided BML server for JBML which was used by MSG-048 2007
• Hard coding the BML made server development a source of delay
  – And the server itself a source of bugs
• Needed a way to adapt to change faster
  – With fewer errors and easier to fix
  – SIMCI Combined Project 2008 supported a solution
BML in SIMCI Combined Project 2008

24 Feb 2010
Dr. Mark Pullen GMU C4I Center
BML Software Infrastructure
BML System Used By NATO MSG-048 for 2009 Experimentation

24 Feb 2010

Dr. Mark Pullen GMU C4I Center

BML Software Infrastructure
Technical Advantages of Scripted Approach

• Middleware functions don’t change
  – Mapping BML to JC3IEDM and push/pull to database
  – Program these once and get them right

• Interpreted WS offers flexibility
  – Rapid implementation of new BML constructs
  – Easy to modify underlying data model
    • JC3IEDM also continues to change
  – Reduces time and cost for prototyping
  – Scripting language provides a concise definition of BML-to-data model mappings
  – Although bugs still happen, the number of possible mistakes is far smaller

• Scripted operation may, however, be slower
  – We’re multithreading the server which will help this
Scripting Example: TaskeeWho

BML Input

<!-- Fragment of <OrderPush> -->

<Task>
  <GroundTask>
    <TaskeeWho>
      <UnitID>UIE9 FA</UnitID>
    </TaskeeWho>
  </GroundTask>
  ...

24 Feb 2010

Dr. Mark Pullen GMU C4I Center
BML Software Infrastructure
Previous Approach: IDEF1x Mapping Definition

JBML mapping to JC3IEDM

Schema field <What>

<table>
<thead>
<tr>
<th>Reference:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;xsd:complexType name=&quot;CommandType&quot;&gt;</td>
</tr>
<tr>
<td><a href="">xsd:sequence</a></td>
</tr>
<tr>
<td>&lt;xsd:element name=&quot;What&quot; type=&quot;GroundBMLWhatType&quot;/&gt;</td>
</tr>
</tbody>
</table>

Used as the <OrderId>

The category-code value will be set to 'ACTION-TASK' ('ACTTA')

Not machine readable though highly structured
Script is a concise XML coding of this
The category-code value will be set to 'ORDER' ('ORD')

The name of the OBJECT-ITEM who is doing the tasking <TASKERWHO>

The category-code value will be set to 'IS directed by' ('ISDSCR')

The category-code value will be set to 'ORGANISATION' ('OR')

The category-code value will be set to 'UNIT' ('UN')

The category-code value will be set to 'ACTION-TASK' ('ACTTA')

The category-code value will be set to 'ORDER' ('ORDER')

Where content-category-code is set to 'ORDER' ('ORDER')
XML Script for SBML

<BusinessObjectTransaction>
    <transactionName>TaskeeWhoPush</transactionName>
    <parameter>task_act_id</parameter>
    <tableQuery>
        <databaseTable>unit</databaseTable>
        <queryAction>GET</queryAction>
        <resultName>unit_id</resultName>
        <columnReference>
            <columnName>formal_abbrd_name_txt</columnName>
            <businessObjectTag>UnitID</businessObjectTag>
        </columnReference>
    </tableQuery>

    <tableQuery>
        <databaseTable>act_res</databaseTable>
        <queryAction>PUT</queryAction>
        <columnReference>
            <columnName>act_id</columnName>
        </columnReference>
    </tableQuery>

. . . (2 more pages like this)
Example Condensed Scripting Language
for complete TaskeeWhoPush

Transaction TaskeeWhoPush (task_act_id)
{

Query GET table unit SET unit_id
  Column (formal_abbrd_name_txt SET UnitID);
Query PUT table act_res
  Column (act_id FROM task_act_id NEWKEY)
  Column (act_res_ix FROM act_res_index)
  Column (cat_code FROM "RI")
Column (authorising_org_id FROM unit_id);
Query PUT table act_res_item
  Column (act_id FROM task_act_id)
  Column (act_res_ix FROM act_res_index)
  Column (obj_item_id FROM unit_id);
Return tag Result FROM “OK”;
}

24 Feb 2010
Scripted BML WS Configuration

Two implementations: MySQL and SIMCI RI
Scripted BML WS Design

- Basic operations: *push* and *pull*
  - Currently, servers for SQL and RI databases
  - Scripts implement BML Orders and Reports

- Script defines implementation of Business Objects (constituents of the higher-level BML grammar) over the JC3I EDM data model
  - BO is an XML subtree rooted at a defined node in the XML file – can invoke other BO

- Interpreter uses two files plus WS input
  - Mapping file contains script
  - BML schema file provides necessary context
Scripted Interpreter

XML (or API) parametric input/output for business object (TaskerWho, TaskeeWho, What, When,...)

BML Schema

Mappings (XML encoded)

SAX/DOM

mappings interpreter

Web Service using Script Schema

JC3IEDM database

API or service
Why Publish/Subscribe

• “Pure” Web Service is always accessed by push or pull transaction from client
  – No provision for server to initiate action
• For clients to stay up to date they must pull latest status from server at rate determined by their need for up-to-date information (called polling)
  – Result: MSG-048 server in 2008 spent most of its time responding to status pulls
• Publish/subscribe gets around this by letting clients identify the categories of information they need – they subscribe to Topics
  – Server sends them a copy of every update associated with each subscribe Topic
  – More timely updates and a dramatic reduction in overhead
Publish/Subscribe Architecture
new in SBMLv2.1

BML Software Infrastructure
Publish/Subscribe in SBML

• GMU team chose JBoss publish/subscribe for compatibility with SIMCI-CP architecture
  – Based on Java Messaging Service
• No native C++ interface so we provided one based on the Java Native Interface (JNI)
  – C++ interface developers got on well with this
• Established set of simple Topics based on Order and Report types
  – More sophisticated choice might have reduced message load on clients
  – However this was our first try so we opted for simplicity
  – Worked well but we might opt for more sophisticated Topics in future
Scripted BML Summary

• Scripted BML WS has great potential as development tool
  – Enables developing services more rapidly
  – Eliminates the broad class of bugs that can’t be expressed in the script
  – Will be available as open source, offered as Reference Implementation

• Scripting language provides a highly concise definition of BML mapping
  – Paradigm for executable specification
  – Recommended that SISO C-BML use this approach

• Easy to evolve with data model
  – JC3IEDM or any other

• SBML software available open source from GMU
BML C2 GUI

• Inspired by FGAN C2 Lexical Grammar GUI
• Provides a way to inspect and modify BML orders and reports in context of schema
  – Includes ability to display BML geospatial data and control features on an open format map
  – Can be used to provide simple input/output in BML system or inspect data during development

• Open source software for public release
  – Uses “Jaxfront” commercial open source software to generate forms from schema
  – And open map graphics
BML C2 GUI: ORDER
BML C2 GUI : REPORT WITH 2525B ICON
Other Software for Distributed Development

• Collaboration software
  – Internet teleconferencing

• Version Management
  – Subversion versioning system

• Trac Management Software
  – Seamless integration with Subversion
  – Wiki with collaborative editing
  – Issue tracking and resolution

• Operation software
  – Initialization of Web service
Internet Audiographic Conferencing

• Used GMU Network EducationWare (NEW)
  – Runs over open Internet and is open source/free
  – Course delivery software also for conferencing

• Important functions:
  – Half-duplex voice with quick turnover
  – Floor control establishes who is speaking
  – Whiteboard allows document to be shared and annotated by all
NEW Client Interface

GMU C4I Center Networking and Simulation Laboratory

NEW

Network EducationWare Project

For more details, see:

http://netlab.gmu.edu/NEW
Subversion Versioning System

• An excellent way to maintain shared access to changing documents
  – Maintains history of all changes
  – Normally, latest update is downloaded
  – Role-based access control
• Can be accessed via Web browser
• Upload requires a special (free) client
Trac Wiki Built on Subversion

- A *Wiki* is a set of shared files that a group of users can access and update
- Using Trac, any Wiki input always can be recovered
- MSG-048 built its Wiki around the functions we were performing
  - Experiment planning and setup
  - Technical planning and execution
  - Schema maintenance

24 Feb 2010

Dr. Mark Pullen GMU C4I Center
BML Software Infrastructure
<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Rev</th>
<th>Age</th>
<th>Last Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>../</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6Dec2009_VIP_Briefing</td>
<td>458</td>
<td>11</td>
<td>days</td>
<td>hkevin:</td>
</tr>
<tr>
<td>10S-SIW-049</td>
<td>460</td>
<td>8</td>
<td>days</td>
<td>mpullen:</td>
</tr>
<tr>
<td>2010_CBML_Workshop</td>
<td>456</td>
<td>11</td>
<td>days</td>
<td>hkevin:</td>
</tr>
<tr>
<td>ExperimentGroup</td>
<td>391</td>
<td>3</td>
<td>months</td>
<td>hkevin:</td>
</tr>
<tr>
<td>Meeting minutes</td>
<td>441</td>
<td>2</td>
<td>weeks</td>
<td>hkevin:</td>
</tr>
<tr>
<td>NEW TAP</td>
<td>133</td>
<td>8</td>
<td>months</td>
<td>lionel.khin</td>
</tr>
<tr>
<td>References</td>
<td>41</td>
<td>9</td>
<td>months</td>
<td>marc.st-or</td>
</tr>
<tr>
<td>Reports</td>
<td>442</td>
<td>2</td>
<td>weeks</td>
<td>hkevin:</td>
</tr>
<tr>
<td>Scenario team</td>
<td>388</td>
<td>3</td>
<td>months</td>
<td>ole-martin</td>
</tr>
<tr>
<td>Screenshots</td>
<td>373</td>
<td>3</td>
<td>months</td>
<td>anders.als</td>
</tr>
<tr>
<td>TechGroup</td>
<td>394</td>
<td>2</td>
<td>months</td>
<td>anders.als</td>
</tr>
<tr>
<td>InstructionsForConfiguringGMU_NMSG-048-SVN-Client.doc</td>
<td>461.0 KB</td>
<td>381</td>
<td>3 months</td>
<td>hkevin:</td>
</tr>
<tr>
<td>logos.ppt</td>
<td>417.5 KB</td>
<td>459</td>
<td>9 days</td>
<td>nclark1:R</td>
</tr>
</tbody>
</table>
Trac Bug Reporting

- With 8 project teams finding and reporting bugs, it was very difficult to manage resolution
- MSG-048 used the bug reporting and management system built into Trac
  - Collects all pertinent details of problem
  - Informs people who need to know
  - Maintains status accessible online
  - Programmer posts when problem resolved
Initialization

• Getting all connected systems to have a common state is a significant challenge
• In 2009, MSG-048 settled for providing a mechanism to initialize the BML Server
  – “NewWho” service pushed values of all needed JC3IEDM attributes needed for a unit
    • Some simulations actually did this while running
  – Much more work is needed to initialize all systems automatically
  – Start/stop and logging of operation also needed
Conclusion

• Operating a BML Coalition requires careful attention to infrastructure
• BML Schema and Server are essential factors in success
• Server performance and stability critical
  – Ease of adapting server also important while BML is experimental
  – Publish/subscribe capability essential
• Other supporting software also is essential to distributed development of and by coalition