Joint Battle Management Language (JBML) – US Contribution to the C-BML PDG and NATO MSG-048 TA

Who is Part of JBML Phase I

- Mr. Michael Powers, TEC, Program Manager
- Dr. Mark Pullen, GMU, Project Lead
- Dr. Stan Levine, GMU, Project Manager
- Dr. Michael Hieb, GMU, Technical Lead
- Dr. Andreas Tolk, ODU, Standards Lead
- Dr. Harry Keeling, HU, Testbed Lead
- Mr. John Roberts, ACS, Ground Lead
- Mr. Curt Blais, NPS, Maritime Lead
- Mr. David Perme, Gestalt, Air Lead
- Mr. John Kearley, DRC, Scenario Lead
- Ms. Shea Smith, JATTL, JFCOM Coordinator
Presentation Outline

- History of JBML / Relation to other BML Efforts
- JBML Architecture
  - Layers of JBML Services
  - BML Domain Configured Services
  - BML Base Services
  - BML Common Data Access Service
- Results
- Contributions to C-BML

History of JBML

Relationship to other BML Efforts
Objective of Current Efforts

• Build and demonstrate
  – an initial Joint Battle Management Language Capability
  – to transmit Digital Orders to Joint (and Combined) Forces
  – using a Battle Management Language Specification

Genealogy of JBML
JBML Vision

C2 Domain Language(s)

- Ground BML
- Air BML
- Maritime BML
- Logistics BML
- geoBML
- Peacekeeping BML
- Crisis Management BML

JC3IEDM as “first among equals” with other relevant Data Models (e.g., GIS)

Tasks in JBML

- Develop an initial Joint BML capability for
  - Ground BML
  - Air BML
  - Maritime BML
  in one common language

- Define this common Language with several (Service specific) interoperating domains
  - Common components for shared information
  - Service-specific components for unshared information
  - Shared common controlled vocabulary (based on the JC3IEDM definitions)
  - A Lexical grammar constraining the structure
JBML Architecture

Layers of JBML Services
BML Domain Configured Services
BML Base Services
BML Common Data Access Service

Earlier C-BML Recommendations

- Web Services for C-BML Protocol
- JC3IEDM for C-BML Representation
- Grammar to capture C-BML Doctrine
- Ontology to capture C-BML Doctrine
- Layered Web services
  - Atomic web services for propertied concepts (tables)
  - Composite web services for associated concepts (view, transactional)
  - Aggregate services for system access (data mediation)

JBML supports these ideas and modifies them to fulfill the objectives of the project
JBML Service Architecture

Domain Knowledge

fully defines domain language

BML Domain-Configured Service

Web Service Exchange Interface

XML/WSDL

XSD FILES

JBML Service Architecture

Domain Knowledge

fully defines domain language

BML Domain-Configured Service

BML Base Service

Who, What, Where, When, Why

API

XML/WSDL

XSD FILES

WSDL
JBML Service Architecture

1. Web Service Exchange Interface

2. XML/WSDL

3. Domain Knowledge
   XSD FILES
   fully defines domain language

4. BML Domain-Configured Service
   API

5. BML Base Service
   Who, What, Where, When, Why
   API

6. BML Common Data Access Software

7. JC3IEDM Database

Reference Implementation middleware common to all BML domains

Defined Interfaces all layers include validation

NOTE: Interface 5 (push) and Interface 8 (push & pull) are provided for future use (they are not used in JBML Phase 1)

Slide 13
Web Service Overview

- The **BML Domain Configured Services (DCS)** represent the domain-specific language in form of schemas, inspired by grammar-based research, that are implemented by Web services.

- The grammar uses the **BML Base Services (BBS)** which represents the information element groups that are necessary to specify the information objects of interest, such as the 5Ws (who, what, where, when, why) and other constructs of interest.

- The lowest layer represents the information exchange of information elements. This layer is normally hidden from the user. In JBML, these are **BML Common Data Access Services (CDAS)**.

BML Domain Configured Services (DCS)

- The DCS is implemented in the Document-Literal mode by a generic Web service that is driven by an XML schema.

- The initial lexical grammar used is formally described in terms of a number of primitives, that are labeled with:

  - `<task>` (verb)
  - `<tasker-who>`
  - `<taskee-who>`
  - `<affected-who>`
  - `<what>` (action)
  - `<where>`
  - `<start-when>`
  - `<end-when>`
  - `<why>`
  - `<label>`
  - `<modifier>`

- These primitives are represented in the BML Base Services.

- The Domain knowledge produces
  - the XML file defining the DCS information
  - extensions to representing data models (such as JC3I EDM)
XSD Joint Task Type

<xsd:complexType name="TaskType">
  <xsd:choice>
    <xsd:element name="GroundTask" type="GroundTaskType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="AirTask" type="AirTaskType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="MaritimeTask" type="MaritimeTaskType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:choice>
</xsd:complexType>

XSD GroundTaskType

<xsd:complexType name="GroundTaskType">
  <xsd:sequence>
    <xsd:element name="TaskeeWho" type="WhoType"/>
    <xsd:element name="What" type="GroundBMLWhatType"/>
    <xsd:element name="Where" type="WhereType"/>
    <xsd:element name="StartWhen" type="WhenType"/>
    <xsd:element name="EndWhen" type="WhenType" minOccurs="0"/>
    <xsd:element name="AffectedWho" type="WhoType" minOccurs="0"/>
    <xsd:element name="Why" type="GroundWhyType" minOccurs="0"/>
    <xsd:element name="Label" type="LabelType"/>
  </xsd:sequence>
</xsd:complexType>
BML Base Services (BBS)

- BBS provides composite BML elements – such as Who, What, When, Where, and Why – as primitives to the DCS
- Other elements may be introduced for new and existing BML domains as required
- The BBS accesses all of the database tables relating to the composite elements through software that implements the Common Data Access Services
  - Mapping between BBS data elements and CDAS data elements
  - Data mediation – where needed – within the implementing web services
- The standard at this layer will identify the information objects exposed by the database tables to be updated for each BML information element and the validation conditions to be applied

XBML Example of Mapping
Common Data Access Services (CDAS)

- Provide a mechanism for the BBS to both read and update the database tables directly
- For testing and debugging purposes, an inspection mode is implemented
- Generic web service for data access
  - Parameter: table name and attributes
  - Advantage of generic service
    - Efficient access to the JC3IEDM database
    - No updates needed if data model is extended
  - Disadvantage of generic service
    - Data validation only at higher layers
    - Can't provide JC3IEDM interface without database
- Use of database supports asynchronous access (good for development) but precludes faster synchronous C2-simulation connect

JBML Data Representation

- Joint Command, Control and Consultation Information Exchange Data Model (JC3IEDM)
- Extensions and Enhancements derived from the Domain Knowledge
- Additional operationally relevant data models – such as used within Geospatial Information Systems – may be used in addition to the JC3IEDM
- BBS collective update of all tables associated with a given business object (who/what/when/where/why etc) via CDAS ensure consistency
  - Don't allow BBS transactions to be interleaved – can result in inconsistent database state
  - If update is impossible, roll back to original state
JBML Service Architecture

- BML Domain-Configured Service
- BML Base Service
- BML Common Data Access Software

- Domain Knowledge
- JBML Grammar fully defines domain language

- JC3IEDM Domain Specific Extensions
- JC3IEDM Database
- Non-JC3IEDM Databases/Sources

- Reference Implementation Middleware common to all BML domains
- Defined Interfaces all layers include validation
- WSDL
- API
- XML
- SQL

Results

Where are we
Where do we want to go
Demo Concept of the Operation

- **Army** units proceed through **Urban Areas**, seizing designated objectives and destroying enemy forces, in order to reestablish an international border.

- They are preceded by:
  - **Navy** and **Air Force** strikes on key C2 and communication nodes
  - Close Air Support strikes
  - Pre-planned Navy Tomahawk strikes
CJTF-CS Joint orders to component commanders

- **JFLCC** as supported commander directs:
  - **2nd ID** to conduct major ground operations:
    - 1-66 CAB to re-take strategic towns, airfields, railheads, and restore border
  - **JFACC** will conduct offensive operations:
    - Deep strike
    - Close air support
  - **JFMCC** will conduct offensive operations:
    - Provide forces (air and cruise missile) to JFACC for deep strike, close air support
JBML Demo Environment

JBML Follow-on Phase 2

- Phase 2
  - Basic Capability for Joint Operations
    - Adequate Land/Littoral/Sea/Air for exercises
    - Initial PMESII support
    - Two-way flow between C2 and Simulation
  - Evolving standards process
    - First balloted C-BML standard
    - Proof of Principle for NATO
Summary

Contributions to C-BML

What can C-BML get from JBML

• The immediate contribution of the JBML project to C-BML is the service architecture
• The JBML architecture will provide a regular and extensible framework upon which a powerful, flexible and growing family of standards can be created
• Contributions on all identified layers
  – Primitives of the DCS
  – BBS as applicable in the SISO context and
  – CDAS (potentially with transient implementations)
  – Recommended extensions and alternative data models
• JBML Web services are open source
Web Service Infrastructure

• The JBML Web services are available as experimental infrastructure to create a reference implementation of C-BML
• While JBML is designed to exchange information with C2 systems and simulations at the top (DCS) layer, we recognize that C–BML needs to offer flexibility of interfacing
  – Comply with standard by interfacing at any layer
• The JBML Web services therefore will be configurable to expose all three layers
  – Can be configured (and, if necessary modified) to create a reference implementation at every layer

Questions

GEOGE MASON UNIVERSITY
C4I Center