2007

Joint Battle Management Language (JBML) - Phase 1 Development and Demonstration Results

Levine, S.
Joint Battle Management Language (JBML) - Phase 1 Development and Demonstration Results

Fall 2007 Simulation Interoperability Workshop
Paper 07F-SIW-051

September 18, 2007
Outline

• Purpose

• BML Background

• JBML Phase 1 Description

• Demonstration Results

• Summary/Conclusions
Purpose

The purpose of this briefing is to present the results of Phase 1 of the Joint Battle Management Language Program (JBML).
Outline

• Purpose

• BML Background
  • JBML Phase 1 Description
  • Demonstration Results
  • Summary/Conclusions
The Problem

• Our current “BML” is a language tailored to interpersonal communication

• It is riddled with ambiguity and overlapping definitions

• It will not support the integration of advanced modeling and simulation with “digitized” command and control
BML is being developed as a **standard representation** of digitized C2 information for executable **plans, orders, requests and reports** for:

- military units,
- simulated forces, and
- future robotic forces.
Benefits of Adopting BML

• Supports better integration of simulations with C2 systems
• Reduces time and cost to accomplish training / operations
• Reduces number of workstation controllers, unique interfaces, and interpreters
• Improves Service, Joint, Coalition interoperability

Key enabler for battle staff training capability
Outline

• Purpose
• BML Background

• JBML Phase 1 Description
• Demonstration Results
• Summary/Conclusions
Phase 1 Project Definition

- **Build and demonstrate** an initial **Joint BML** capability to transmit digital **orders**:
  - to Joint and Combined Forces
  - using a Battle Management Language specification
  - for Proof of Principle
JBML Phase 1 Products

• Proof of Principle Demo for Joint Capability
  – Representative Land, Sea, Air

• Input to a draft SISO standard
  – Including Web Service infrastructure

• Integrated into NATO MSG-048 planning
JBML Phase 1 Team Leads

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

1. WSDL
2. XML/WSDL
3. XSD FILES

Domain Knowledge

fully defines domain language

Web Service Exchange Interface

BML Domain-Configured Service
JBML Web Service Architecture

Domain Knowledge

fully defines domain language

BML Domain-Configured Service

Web Service Exchange Interface

1 WSDL

2 XML/WSW

BML Base Service

Who, What, Where, When, Why

3 XSD FILES

4 API
The diagram illustrates the JBML Web Service Architecture:

1. **Web Service Exchange Interface (WSDL)**: Defines interfaces for all layers.
2. **XML/WSDL**: Files for domain knowledge.
3. **BML Domain-Configured Service**: Configures the service based on domain knowledge.
4. **BML Base Service**: Provides core functionalities.
5. **API**: Connector for all BML domains.
6. **BML Common Data Access Software**: Provides common data access.
7. **SQL Database**: Stores data for all BML domains.

Key points:
- All layers include validation.
- JC3IEDM Domain Specific Extensions.
- JC3IEDM Database.
JBML Web Service Architecture

1. Web Service Exchange Interface
2. XML/WSDL
3. Domain Knowledge
4. API
5. XML/WSDL
6. API
7. SQL
8. XML/WSDL

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

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

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

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

• Purpose
• BML Background
• JBML Phase 1 Description
  • Demonstration Results
• Summary/Conclusions
JBML Demo Environment

- BML – C2 Interfaces
- C2 Specific Interface
- C2 Specific Interface
- USMTF plus
- XML
- NATIVE BML

- JTC
- GCCS/TBMCS
- C2PC/CAPES

Maritime
Air
Ground
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
Gulf of Caspia Joint Operations

- Tanrykulular Airfield
- Baku-Bina Airfield
- Ground Forces
- Joint Urban Operations Target Area

CVSG

Tomahawk
Russell
Hamilton
Antietam
Santa Fe

Reagan
Lincoln

CVSG

Tomahawk
Mobile Bay
Shoup
Momsen
Chicago
Outline

• Purpose
• BML Background
• JBML Phase 1 Description
• Demonstration Results

• Summary/Conclusions
Available at http://netlab.gmu.edu/JBML (under password control)

- Descriptive documents
  - Architecture overview
  - Code documentation (narrative & Javadoc)
  - XSD Web service schema
  - SQL database schema

- Open source code
  - Latest version of all Web services
  - GUI to inspect JC3IEDM database
Summary

• Phase 1 has been successful in developing a basic Joint integrated capability:

  – ATO, Ground OPORD, TLAM INDIGO

  – **TBMCS, JTC**, C2PC, multiple JSAF

• Basis for Coalition BML Standard

• Basis for follow-on MSG-048 Demo
Questions?
BACKUPS
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 Products
JBML Support to C-BML

- The C-BML Standard will include:
  1. a data model in a subset of JC3IEDM
  2. an Information Exchange content and structure specification in the form of an XML schema
  3. and an Information Exchange mechanism specification embedded into a WSDL document
  4. Guidelines

- The JBML effort will provide:
  1. JC3IEDM subset and extensions
  2. Domain Specific Interface and Functionality
  3. Composite/Atomic interfaces and functionality
  4. Embedded information that can be used to develop a C-BML guideline document.
JBML Domain-Configured Service (DCS)

• Purpose:
  – Provide a high-level, semantically consistent, XML-based language definition
  – Modular and readily extensible
  – Structure based on C2 Lexical grammar of Hieb & Schade
    • Gives high confidence the language will meet BML needs
    • No actual grammar processing; just a way to structure JBML

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

1. WSDL
2. XML/WSDL
3. XSD FILES

Domain Knowledge

fully defines domain language

BML Domain-Configured Service

Web Service Exchange Interface
JBML DCS OrderType

<xsd:annotation>
  <xsd:documentation>Provides basic information that applies to all Tasks in the order</xsd:documentation>
</xsd:annotation>
<xsd:complexType name="OrderType">
  <xsd:sequence>
    <xsd:element name="OrderMode" type="OrderModeType" default="SINGLE" minOccurs="0"/>
    <xsd:element name="TaskersIntent" type="FreeTextType" minOccurs="0"/>
    <xsd:element name="Task" type="TaskType" maxOccurs="unbounded"/>
    <xsd:element name="OrderIssuedWhen" type="WhenType"/>
    <xsd:element name="OrderID" type="OrderIDType"/>
    <xsd:element name="TaskerWho" type="WhoType"/>
    <xsd:element name="TaskOrganization" type="msdl:TaskOrgType" minOccurs="0"/>
    <xsd:element name="EnemyTaskOrg" type="msdl:TaskOrgType" minOccurs="0"/>
    <xsd:element name="ControlMeasures" type="MultipleControlMeasuresType" minOccurs="0"/>
    <xsd:element name="TargetList" type="TargetListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
JBML DCS Joint TaskType

<xsd:annotation>
  <xsd:documentation>
    Defines the domain of warfare associated with the task
  </xsd:documentation>
</xsd:annotation>
<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>
JBML DCS GroundTaskType

<xsd:complexType name="GroundTaskType">
  <xsd:sequence>
    <xsd:element name="TaskeeWho" type="WhoType"/>
    <xsd:element name="What" type="GroundWhatType"/>
    <xsd:element name="Where" type="WhereType"/>
    <xsd:element name="StartWhen" type="WhenType" minOccurs="0"/>
    <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>
JBML BML Base Service (BBS)

• Purpose:
  – Provide basic building blocks that can be used by multiple domains
    • push/pull API and push/pull Web service
  – who/what-when/where/why/controlMeasures etc.
  – Logical transactions that are “atomic” in the Computer Science sense
    • Must be committed to database all-or-nothing
  – Avoids need to recode the building blocks for every new domain
    • “Where” touches up to 14 tables
    • And requires 373 lines of code, including comments/whitespace
    • To be interoperable, this must be done right – why do it over for every system that is interfaced?
JBML mapping to JC3IEDM
Schema field <Why>

Schema Reference:

```xml
<xsd:complexType name="CommandType">
  <xsd:sequence>
    <xsd:element name="Why" type="WhyType" minOccurs="0"/>
    ...
  </xsd:sequence>
</xsd:complexType>
```

Where content-category-code is set to ‘ORDER’ ('ORDER')

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

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

Used as the <OrderId>

Category-code is set to 'ORDER' ('ORD')

2

3

<Label>

<What>

<StartWhen>

<EndWhen>

<WhyType>
JBML mapping to JC3IEDM
Schema field <WhereCategory>

1. action
   - action-id
   - category-code

2. action-context
   - action-id
   - context-id
   - action-context-index
   - category-code

3. action-reference-association
   - action-id (FK)
   - reference-id (FK)
   - action-reference-association-index
   - category-code

4. object-item
   - object-item-id
   - object-type-id
   - category-code

5. context-object-item-association
   - context-id
   - object-item-id
   - category-code

6. object-type
   - object-type-id
   - object-type-id
   - report-data-id
   - category-code

7. feature
   - feature-id
   - category-code

8. object-item-type
   - object-item-type
   - object-type-id
   - index
   - category-code

9. control-feature-type
   - control-feature-type-id
   - category-code

10. route-type
    - route-type-id
    - category-code

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

If CONTROL-FEATURE is a ROUTE-TYPE then category-code is set to 'ROUTE-TYPE' ('RTETYP'), otherwise set to appropriate selection.

If CONTROL-FEATURE is a ROUTE then category-code is set to 'ROUTE' ('ROUTE'), otherwise set to 'NOT OTHERWISE SPECIFIED' ('NOS').

Defines the nature of the ACTION-CONTEXT; category-code will be set to 'Final state, planning' ('FINPLA')

Set to appropriate selection.

Refer to the following schema reference:

<xsd:complexType name="WhereType">
  ...
  <xsd:element name="WhereCategory">
    ...
  </xsd:element>
</xsd:complexType>
JBML Service Architecture

Domain Knowledge

1. WSDL

Web Service Exchange Interface

2. XML/WSDL

Domain Knowledge

3. XSD FILES

BML Domain-Configured Service

4. API

BML Base Service
Who, What, Where, When, Why

fully defines domain language
JBML BBS WhyType

<xsd:annotation>
  <xsd:documentation>Defines Why a Task is undertaken</xsd:documentation>
</xsd:annotation>
<xsd:simpleType name="WhyTypeEffectDescriptionCode">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="DSTRYK"/>
    <xsd:enumeration value="FKIL"/>
    <xsd:enumeration value="IDNT"/>
    <xsd:enumeration value="ILLUMN"/>
    <xsd:enumeration value="INTREC"/>
    <xsd:enumeration value="KILL"/>
    <xsd:enumeration value="LDAM"/>
    <xsd:enumeration value="LGTRST"/>
    <xsd:enumeration value="MKIL"/>
    <xsd:enumeration value="MODDAM"/>
    <xsd:enumeration value="NORSTN"/>
    <xsd:enumeration value="NOS"/>
    <xsd:enumeration value="SDAM"/>
    <xsd:enumeration value="SUPRSD"/>
  </xsd:restriction>
</xsd:simpleType>
JBML Common Data Access Software (CDAS)

• Purpose:
  – Access the JC3IEDM database (push/pull API and WS)
  – Since the database is SQL-based, this only needs to be a wrapper
    • With validation that only the intended tables are accessed
  – We run this in pull-only mode
  – If it is intended to push BML input directly through JC3IEDM, the CDAS push service needs much better validation of data values
  – We didn’t build that sort of service because it lacks the control needed for multiple users to update
    • All tables for a transaction must be updated together with no intervening access by other users
    • We keep this turned off
JBML Service Architecture

- **Domain Knowledge**
  - fully defines domain language

- **BML Domain-Configured Service**

- **BML Base Service**
  - Who, What, Where, When, Why

- **BML Common Data Access Software**

- **JC3IEDM Domain Specific Extensions**

- **JC3IEDM Database**

- **WSDL**

- **XML/WSDD**

- **API**

- **SQL**

Reference Implementation middleware common to all BML domains

= Defined Interfaces

all layers include validation
JBML Service Architecture

1. Web Service Exchange Interface

2. BML Domain-Configured Service

3. BML Base Service

4. BML Common Data Access Software

5. JC3IEDM Domain-Specific Extensions

6. JC3IEDM Database

7. Non-JC3IEDM Databases/Sources

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

- Domain Knowledge
- fully defines domain language
- Domain
- Knowledge
- XSD FILES

- Reference Implementation middleware common to all BML domains

- Delivered to C-BML
- 8 Defined Interfaces
- all layers include validation

- XML/WSDL
- API
- SQL
JBML Web Services

Available at http://netlab.gmu.edu/JBML:

- Descriptive documents
  - Architecture overview
    - Domain Configured Service (DCS); BML Base Service (BBS); Common Data Access Service (CDAS)
  - Code documentation (narrative & Javadoc)
  - XSD Web service schema
  - Supporting JC3IEDM mappings
  - SQL database schema

- Open source code
  - Latest version of all Web services
    - CDAS and BBS made possible by open source bootstrap of VMASC Atomic and Composite services
  - GUI to inspect JCDIEDM database using CDAS
JBML XML Schema Definition (xsd) on JBML website

- DCSOrderPush, DCSOrderPull
  - Top-level definition of Joint order C2 and Simulation services
- OrderTypes
  - Defines Joint operations order consisting of multiple Air, Ground, and Maritime commands
- AirTypes, GroundTypes, MaritimeTypes
  - Define domain-specific information
- FiveWTypes
  - Defines common Who / What-When / Where / Why etc.
- msdlTypes
  - Describes reusable MSDL schema from their webpage
  - This approach implements SISO guidance to make MSDL and C-BML interoperable
JBML Inputs to SISO C-BML

https://netlab.gmu.edu/JBML

- JBML had a successful demonstration on 3 May 07
  - Ground/Air/Maritime C2
  - Two versions of JSAF
  - Proved in principle that JBML can work

- But, realistically, there has not been enough experience with the DCS layer to propose standardization
  - NATO MSG-048 will provide more experience by Dec 07

- The component that is ready to consider for standardization is the BBS mappings
  - We have five years experiences with who/what/when/where/why
  - And JC3IEDM provides a stable model for the database
  - C-BML should vet the mappings, revise if needed, and adopt

- Open Source JBML reference implementation Web Service also should be useful
Major Issues

• Serious issue surfaced working with USMTF
  – BML carries more semantics than JC3IEDM
  – The aggregate schema reached FOUO level
  – How to deal with sensitive information in an open standard?

• In the future, we recommend that SISO develop a set of rules for top-level (DCS) BML and a reference plain-vanilla example schema
  – Similar to the HLA: the standard is in the rules

• And also a standardized set of mid-level (BBS) mappings who/what/when/where/why etc to JC3IEDM
  – Avoids multiple interpretations of JC3IEDM
  – And allows re-use of composite-level code