An Object-Oriented XML Schema for the MIP Joint Command, Control, and Consultation Information Exchange Data Model

Michael Gerz, FGAN FKIE
Erik Chaum, DMSO – Francisco Loaiza, IDA

2006 CCRTS
June 20 – 22, 2006
Overview

- Motivation
- MIP JC3I EDM
- Use Cases, Requirements & Design Principles
- Transformation Rules
- XML Example
- Tool Support
- Summary
Motivation

- Multinational interest in the use of **XML for C2 info exchange**
  - National and multinational projects
  - NATO XML Registry

- **Semantic interoperability** is critical to information sharing
  - XML does not automatically ensure common understanding

- **Multilateral Interoperability Programme** (MIP)
  - Voluntary multinational C2 Community of Interest (COI)
  - Supported by 25 nations, NATO, and ACT
  - Defines the Joint Command, Control, and Consultation Information Exchange Data Model (JC3I EDM)
MIP Information Exchange Data Model

- Entity-relationship model in IDEF1X notation
- JC3I EDM 3.0 comprises:
  - 241 entities
  - 1244 attributes
  - 7592 fixed domain values
- Semantic definitions & extensive business rules
- Generic - not derived from any specific system
- Edition 3.0 published in December 2005
  - NATO ratification as STANAG 5525

FGAN
Research Institute for Communication, Information Processing and Ergonomics
MIP JC3I EDM - Modeling Capabilities (1)
MIP J C3I EDM - Modeling Capabilities (2)

- **Militarily relevant objects**
  - Current status and location
  - Nominal/actual capabilities & equipment
  - Geopolitical, ethnic, religious, and functional affiliations
  - ORBAT & task organization
  - Addresses

- **Locations/ Geometry**
  - Points, lines, surfaces, volumes
  - Relative vs. absolute locations
MIP JC3I EDM - Modeling Capabilities (3)

- **Actions**
  - Temporal and functional relationships
  - Rules of engagement
  - Targets

- **Meta information**
  - Reporting organization
  - Reporting date & time
  - Accuracy, credibility, reliability
  - Duration of validity
  - etc.
MIP JC3I EDM - XML Use Cases

- **Data Exchange**
  - Web Services (e.g., exchange of business objects)
  - Exchange with non-MIP databases
  - MIP XML Exchange Mechanism

- **Transformation Services**
  - Supporting tactical communications interfaces, ADatP-3 or USMTF (effectively always lossy)
  - Mediation between different versions of the MIP IEDM
  - Export to various output/presentation formats, e.g., STANAG 5500-conformant messages or HTML
Reference MI P XML Schema Definitions

- **Reference MI P XML schema definitions**
  - MI P IEDM defines the C2 XML namespace semantics
  - Avoid diversity of XML vocabularies
  - Collaborative efforts ease national development, improve interoperability, and reduce national costs

- Two reference XSDs for different purposes
  - **RDBMS XSD** – Used for database replication
  - **WS/ OO XSD** – Used for web services/SOAs
WS/ OO XSD - Requirements (1)

- **Message exchange**
  - Referentially complete, self-contained messages
  - Ensure referential integrity within a single XML document
  - Describe cyclic data structures
  - Allow tailoring to specific business object XSDs
  - No fragmentation of data to enhance readability

- **Replication & Query-based communication**
  - Initial synchronization, incremental updates
  - Incomplete query results may refer to unknown objects
  - Support references to external information
  - Relax referential integrity checks (to be ensured by web service & client)
WS/ OO XSD - Requirements (2)

- Consider **Naming and Design Rules** (NDRs)
  - Naming of XML elements & attributes, schema versioning, modularization, namespaces, (restricted) use of XML Schema constructs, etc.
  - Discovery and reuse of common data elements

- **NATO Guidelines for XML Naming and Design** (GXND)
  - Based on ISO/IEC 11179 (Metadata registry) and ISO 15000-5 (ebXML)
WS/ OO XSD - Design Principles

- **Object-Oriented**
  - Structure of instance XML documents matches the natural OO concepts (e.g., inheritance, object identifiers, navigability)
  - Abstraction from the technical aspects of the relational model
  - Abstraction from any underlying persistence mechanism

- Based on **Syntactic Transformations** only
  - Suitable for any version of the MIP IEDM
  - Transformations are easily traceable
  - XSD can be generated automatically from the IDEF1X model
  - Minimizes XSD maintenance cost

- Checks **Semantic Constraints**
  - Domain values, optionality, cardinality of relationships
  - Type-safe document consistency
WS/ OO XSD - Transformation Rules

- **Names**
  - Simple types, codes, attributes, entities

- **Domains**
  - Simple types and codes
  - Based on predefined XML Schema simple types

- **Entities**
  - Optional and mandatory attributes
  - Object identifiers replace JC3I EDM’s synthetic keys

- **Relationships**
  - One-to-many relationships
  - Sub-type relationships
  - Many-to-many relationships (associative entities)
## Transformation Rules - Identifying Relationships

<table>
<thead>
<tr>
<th>Design Pattern</th>
<th>IDEF1X</th>
<th>UML</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDEF1X</strong></td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>&lt;A&gt;</code></td>
</tr>
<tr>
<td>A</td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>A</code></td>
</tr>
<tr>
<td>a-id</td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>&lt;BList&gt;</code></td>
</tr>
<tr>
<td>B</td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>A</code></td>
</tr>
<tr>
<td>a-id (FK)</td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>&lt;BList&gt;</code></td>
</tr>
<tr>
<td><strong>UML</strong></td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>&lt;B&gt;</code>...<code> &lt;/B&gt;</code> (inline)</td>
</tr>
<tr>
<td>A</td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>&lt;B&gt;</code>...<code> &lt;/B&gt;</code> (by reference)</td>
</tr>
<tr>
<td>bList[0..*]:B</td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>&lt;/BList&gt;</code></td>
</tr>
<tr>
<td>0..*</td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><code>&lt;/A&gt;</code></td>
</tr>
</tbody>
</table>
## Transformation Rules - Subtype Relationships

<table>
<thead>
<tr>
<th>IDEF1X</th>
<th>Design Pattern</th>
<th>UML</th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>a-id</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a-category-code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>a-category-code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b-id (FK)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**XML**

```xml
<B>
  <Attr1 of A>…</Attr1 of A>
  <Attr2 of A>…</Attr2 of A>
  ...
  <Attr1 of B>…</Attr1 of B>
  ...
</B>
```

*Drop category codes*
Transformation Rules - Incomplete Subtyping

- MILITARY-ORGANISATION-TYPE
  - military-organisation-type-id (FK)
  - military-organisation-type-category-code
- TASK-FORMATION-TYPE
  - task-formation-type-id (FK)
  - task-formation-type-category-code
- MILITARY-POST-TYPE
  - military-post-type-id (FK)
  - military-post-type-category-code
  - military-post-type-rank-code
- UNIT-TYPE
  - unit-type-id (FK)
  - unit-type-category-code
  - unit-type-arm-category-code
  - unit-type-arm-specialisation-code
  - unit-type-supplementary-specialisation-code
  - unit-type-general-mobility-code
  - unit-type-qualifier-code
  - unit-type-size-code
  - unit-type-principal-equipment-type-id (FK)
- AbstractMilitaryOrganisationType
  - abstract="true"
- (Other)MilitaryOrganisationType
  - abstract="false"
- MilitaryPostType
- TaskFormationType
- UnitType
- ExecutiveMilitaryOrgType

derived by extension
### Transformation Rules - Associative Entities

<table>
<thead>
<tr>
<th>Design Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDEF1X</strong></td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>a-id</td>
</tr>
<tr>
<td>A-B-ASSOC</td>
</tr>
<tr>
<td>a-id (FK)</td>
</tr>
<tr>
<td>b-id (FK)</td>
</tr>
<tr>
<td>some-attr</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>b-id</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>UML</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A-B-ASSOC</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>XML</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;A&gt;</td>
</tr>
<tr>
<td>&lt;ABAsoclnAList&gt;</td>
</tr>
<tr>
<td>&lt;ABAsoclnA&gt;</td>
</tr>
<tr>
<td>&lt;B&gt;</td>
</tr>
<tr>
<td>... &lt;/B&gt;</td>
</tr>
<tr>
<td>&lt;ABAsocln&gt;</td>
</tr>
<tr>
<td>... &lt;/ABAsocln&gt;</td>
</tr>
<tr>
<td>&lt;/ABAsoclnAList&gt;</td>
</tr>
<tr>
<td>&lt;/A&gt;</td>
</tr>
</tbody>
</table>

| <B>             |
|                 |
|   <ABAsoclnBList> |
|     <ABAsoclnB> |
|     <A>         |
|     ... </A>   |
|     <ABAsocln>  |
|     ... </ABAsocln> |
|   </ABAsoclnBList> |
| </B>            |

### Object-to-Association Reasoning
The White Horse Bridge across the Yukon River initially serves as a passage between two minefields located on the north side of the river, one on each side of the bridge. The contingency plan is to use the bridge as part of an obstacle. If the units of the joint task force that are now deployed on the north side of the river need to withdraw, the bridge will be demolished to become part of the main obstacle.

Source: JC3I EDM 3.0 Main Document
WS/ OO Example – J C3I EDM Modeling

FGAN

Research Institute for
Communication, Information Processing and Ergonomics
XML Document - Top-Level Structure

```xml
<?xml version="1.0" encoding="UTF-8"?>
<JC3IEDM xmlns="urn:int:nato:standard:mip:jc3iedm3.0:oo:1.3"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="...">
  <MinefieldLand>
    <OID>O11</OID>
    <NameText>West Minefield</NameText>
  </MinefieldLand>
  ...
  <Bridge>
    <OID>O13</OID>
    <NameText>White Horse Bridge</NameText>
  </Bridge>
  ...
  <MilitaryObstacle>
    <OID>O14</OID>
    <NameText>Obstacle Alpha</NameText>
  </MilitaryObstacle>
  ...
  <ReportingDataAbsoluteTiming>
    <OID>RPTD705</OID>
  </ReportingDataAbsoluteTiming>
  ...
</JC3IEDM>
```

Object Identifiers (OIDs) for referable entities

- Independent entities
- Elements that can be further specified (e.g., associations with statuses)
XML Document - Entities

```xml
<?xml version="1.0" encoding="UTF-8"?>
<J C3I EDM xmlns="urn:int:nato:standard:mip:jc3iedm3.0:oo:1.3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="...">
  <MinefieldLand>
    <OID>OI 1</OID>
    <NameText>West Minefield</NameText>
    ...<LengthDimension>300</LengthDimension>
    <WidthDimension>105</WidthDimension>
    <MineSpacingDimension>10</MineSpacingDimension>
    <DepthPlacementCode>MIXED</DepthPlacementCode>
    <FunctionCode>NUISNC</FunctionCode>
    <PatternCode>REGTHK</PatternCode>
    <PersistenceCode>REMOTE</PersistenceCode>
    <StoppingPowerCode>MEDIUM</StoppingPowerCode>
  </MinefieldLand>
  <Bridge>
    <OID>OI 3</OID>
    <NameText>White Horse Bridge</NameText>
    <LengthDimension>200</LengthDimension>
    <WidthDimension>10</WidthDimension>
    <LongestSpanLengthDimension>40</LongestSpanLengthDimension>
    <SpanQuantity>5</SpanQuantity>
    <UsageCode>RLWYVH</UsageCode>
  </Bridge>
</J C3I EDM>
```

- Inheritance of attributes
- Domain value checks (including NULL)
XML Document - Bridge With Status Information

• Relationships by nesting
• Cardinality/Optionality checks
• Type-safe referential integrity checks
XML Document - Status Update

<?xml version="1.0" encoding="UTF-8"?>
<J C3I EDM xmlns="..." xmlns:xsi="..." xsi:schemaLocation="...">
    <!-- This document is used for the information exchange on 3 November 2003 17:10 -->
    <BridgeRef>
        <OID>OI3</OID>
        <StatusList>
            <Status xsi:type="FacilityStatus">
                <HostilityCode>FR</HostilityCode>
                <ReportingData xsi:type="ReportingData">
                    <OID>RPTD704</OID>
                    <CategoryCode>REP</CategoryCode>
                    <CredibilityCode>RPTFCT</CredibilityCode>
                    <ReportingDatetime>20031103171000.000</ReportingDatetime>
                    <ReportingOrganisationRef>
                        <OID>OI6</OID>
                    </ReportingOrganisationRef>
                    <EntityCategoryCode>OI ASST</EntityCategoryCode>
                    <TimingCategoryCode>TIMNA</TimingCategoryCode>
                </ReportingData>
                <DemolitionStatusCode>PRPEXE</DemolitionStatusCode>
                <MinePresenceCode>NO</MinePresenceCode>
                <OperationalStatusCode>OPR</OperationalStatusCode>
                <OperationalStatusQualifierCode>PASABL</OperationalStatusQualifierCode>
                <CategoryCode>NOS</CategoryCode>
            </Status>
        </StatusList>
    </BridgeRef>
</J C3I EDM>
Tool Support - XML Unmarshalling

XML Parser

- Stream-processing based on Simple API for XML (SAX)
- Object creation/manipulation by Java reflection
- Validation of incoming XML instance documents
- Proves that XML documents can be handled efficiently
  - Nesting of XML elements
  - Associative entities
- Supports incremental updates
- No meta model information required
Findings

- **Naming and Design Rules** only partially suitable
  - JC3I EDM follows its own naming conventions
  - Contradictions between different NDRs

- **Standard XML Binding Frameworks**
  - Resolution of graph structures by XML’s ID/IDREF mechanism
  - Incremental updates require a more flexible reference mechanism

- **W3C XML Schema**
  - Uniqueness & referential constraints by naming conventions
  - Type model of XML Schema cannot be used
  - Technically complicated solution, difficult to extend
WS/ OO XSD - Products

- **XML Design Specification**
  - Standardized as Annex O of the JC3I EDM 3.0

- **WS/ OO Tool Set**
  - XML schema definition, XSD/Java generator, XML examples
  - Generator written in Java 5 SE
  - Eclipse 3.1 project
  - Berkeley Software Distribution (BSD) license

- **Available on the MIP web site**
  - [http://www.mip-site.org](http://www.mip-site.org)
Summary

- **MIP WS/OO XSD** as a means to **harmonize C2I S**
  - XML standard data elements derived from widely accepted data model
  - Based on JC3I EDM semantics

- **Foundation for Web Services and SOAs**
  - Abstracts from ER-specific properties of the MIP IEDM
  - Simplified integration into Non-MIP systems

- **Supports a broad scope of XML applications**
  - Various information exchange mechanisms
  - Builds a bridge between **message-based** and **replication-based communication**

- **Future work**
  - Update to the next version of the JC3I EDM (End of 2006)
  - Formal representation of **business rules**
  - **RDF** and **OWL** representations