Model Interchange Working Group (MIWG)
December 7, 2009

http://www.omgwiki.org/model-interchange

Sanford Friedenthal
MIWG Chair / SE DSIG Chair
sanford.friedenthal@lmco.com

Copyright © 2009 by Object Management Group.
Model Interchange Working Group (MIWG) Objectives and Approach

• Objectives
  – Enhance and demonstrate interoperability of MOF/XMI-based tools with initial focus on model interchange among UML, SysML, and UPDM-capable tools

• Approach
  – Multiple vendors involved
  – Each release incrementally adds test cases
  – Each test case tests additional interchange capabilities among vendor tools
  – Process includes identification and resolution of interchange issues, and demonstration of interchange capability

Copyright © 2009 by Object Management Group.
Model Interchange Via XMI

Modeling Tool

Model/Data Interchange

Modeling Tool

Copyright © 2009 by Object Management Group.
Model Interchange vs Diagram Interchange

- Model interchange with XMI exchanges model information (e.g., classes, associations, activities)
- Interchange does not include diagram layout information
- Most tools provide auto-layout capability to quickly generate the diagrams from the model information
- Future OMG efforts will leverage XMI to also include exchange of diagram layout information

Copyright © 2009 by Object Management Group.
Current Vendor Demo Participants

- Artisan® Studio
- IBM RSx
- IBM Rhapsody
- NoMagic MagicDraw
- SOFTEAM Modelio
- Sparx Systems Enterprise Architect
MIWG Results

• MIWG kickoff 1 year ago (Dec ‘08)
• Completed 6 test cases to date
• General exchange capability demonstrated among vendors, but some issues remain
• Vendors continue to update their tools to address interchange issues
• Refinements to UML spec identified to reduce ambiguity and correct errors
• Guidelines being established for vendor interoperability
• Expect 80+% test case coverage of UML by end Q1 2010
### Incremental Testing Process

<table>
<thead>
<tr>
<th>Test Team</th>
<th>Producer Tool</th>
<th>Consumer Tool</th>
<th>Implementor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define &amp; Validate Test Case Reference Model</td>
<td>Create Diagram &amp; Export XMI</td>
<td>Validate XMI with Validator</td>
<td>Post Model to CM Repository</td>
</tr>
<tr>
<td></td>
<td>Validate XMI with Validator</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Post Model to CM Repository</td>
<td>Import XMI &amp; Create Diagram</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compare to Reference Model</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Capture, Analyze &amp; Identify Issues</td>
<td>Resolve Issues</td>
</tr>
</tbody>
</table>

Copyright © 2009 by Object Management Group.
Example Test Case #2
Reference Diagram (partial)
Example Test Case #2
Reference XMI (partial)

<?xml version="1.0" encoding="UTF-8"?>
<uml:Model xmi:version="2.1" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
xmlns:uml="http://schema.omg.org/spec/UML/2.1.1" xmi:id="_0" name="TestCase2">
<packagedElement xmi:type="uml:Package" xmi:id="Package2" name="Package2">
<packagedElement xmi:type="uml:Package" xmi:id="Package2-MergedPackage" name="MergedPackage">
<packagedElement xmi:type="uml:Class" xmi:id="Package2-MergedPackage-ClassA" name="ClassA"/>
<packagedElement xmi:type="uml:Class" xmi:id="Package2-MergedPackage-ClassB" name="ClassB">
</packagedElement>
</packagedElement>
</packagedElement>
<packagedElement xmi:type="uml:PrimitiveType" xmi:id="Package2-PrimitiveType1" name="PrimitiveType1"/>
<packagedElement xmi:type="uml:Package" xmi:id="Package2-ReceivingPackage" name="ReceivingPackage" visibility="private">
<packagedElement xmi:type="uml:Class" xmi:id="Package2-ReceivingPackage-ClassA" name="ClassA"/>
</packagedElement>
<packagedElement xmi:type="uml:Class" xmi:id="Package2-Class1" name="Class1" clientDependency="Package2-Class1-_interfaceRealization.0">

...........

For illustration only

Copyright © 2009 by Object Management Group.
Incremental Test Cases
Progress To Date

Baseline Specifications: UML 2.1.1 / XMI 2.1

- Test Case 1 - Basic class modeling (complete)
- Test Case 2 - Advanced class modeling (complete)
- Test Case 3 - Profile definition and application (complete)
- Test Case 4 - Simple activity *(complete)
- Test Case 5 - Advanced activity *(complete)
- Test Case 6 – Composite structure (complete)

* Included testing of activity execution using reference implementation for the “Semantics of a Foundational Subset for Executable UML Models” (fUML) specification
Vendor Interchange Capability Matrix  
(December 7, 2009)

Baseline Specifications: UML 2.1.1 / XMI 2.1  
• Test Case 1 - Basic class modeling  
• Test Case 2 - Advanced class modeling  
• Test Case 3 - Profile definition and application  
• Test Case 4 - Simple activity  
• Test Case 5 - Advanced activity  
• Test Case 6 – Composite structure

Disclaimer: This matrix represents preliminary interchange testing results from test cases 1-6 as of December 7, 2009. The model interchange working group (MIWG) participating vendors provided an assessment of their ability to import the other vendors XMI for each test case using the legend indicated, by inputting the data in the row corresponding to their name. The assessment represents the consuming tool import capability and the producing tool export capability.

Copyright © 2009 by Object Management Group.
Incremental Test Cases
Planned

UML 2.2 / XMI 2.1
- Complete UML (80+%)  
  - Release 6 (State machines, Use Cases)
  - Release 7 (Interactions, Collaborations)
  - Release 8 (Retest of test cases 1-6 using UML 2.2)
  - Release 9 (Composite test case)
- SysML  
  - Release 10 (Blocks/parts, Activity swim lanes)
  - Release 11 (Requirements, Additional UML functionality)
  - Release 12 (Parametrics, Additional UML functionality)
  - Release 13 (Allocations, Additional UML functionality)
- UPDM  
  - Release 14
  - Release ....

Copyright © 2009 by Object Management Group.
Summary

• Defined and validated incremental testing process

• Conducted Test Cases 1-6
  – Exchange of models based on class diagrams, activity diagrams, composite structure, and profile mechanism
  – Demonstrated significant interchange capability among multiple vendors

• Continue Incremental Testing
  – Extend coverage across UML, SysML, and UPDM
  – Specification updates
  – Vendor tool updates

Commitment from Vendor Community
Resulting in Good Progress