IMCE MOF2 / OWL2 Integration

Nicolas Rouquette
System Architectures & Behaviors Group, 313K

2011-12-13
Copyright 2011, California Institute of Technology
Government Sponsorship Acknowledged
31 Systems + Software

- #1 Previous work in JPL’s Integrated Model-Centric Engineering (IMCE) initiative for model-based SE
- #5 Systems Engineering with Controlled Vocabularies
- #9 Controlled vocabularies as sentences
- #12 Correspondences: MOF2 vs OWL2 with reification
- #13 MOF2/OWL2 reification for SysML BDDs & IBDs
- #17 Tool support
- #18 SysML 1.4 RTF proposals
  - Property reference path / block-specific type
  - Ontological behavior modeling extension for sysml
What is the IMCE MagicDraw Toolkit?

IMCE’s focus:
- IMCE’s methodology for MBSE is based on the integration of ontology-based modeling and modeling standards
  - Ontology-based modeling is essential for analyzability
  - Standards-based modeling is essential for affordability

IMCE’s strategy:
- Focus on MBSE capabilities that respond to projects’ needs and advance the state of practice for ontology-based MBSE @ JPL

IMCE MagicDraw Toolkit:
- A tactical reduction of the gap towards ontology-based MBSE:
  - SSCAE’s standards-based SysML tool MagicDraw
  - IMCE’s ontology-based SysML tool IMCE MagicDraw Toolkit

This was the plan...
• IMCE plugin key features focus on the small subset of UML that is key to IMCE’s ontology-based SysML, e.g.:
  – Using SysML Blocks for modeling structure, ontologically
  – Using Associations for modeling relationship, ontologically
  – Using Instances for modeling descriptions, ontologically

• The ontological flavor of modeling makes sense
  – see Conrad Bock’s Ontological Behavior Modeling paper

• Challenge: how do we get there for JPL’s MBSE practices?
  – Long term: standards is the only technology NASA can afford
  – Short term: standards have flaws; tools have bugs

• IMCE MagicDraw Toolkit: it’s an evolving balancing act...
• An SE vocabulary is defined as an OWL2 ontology

• Emphasis on organizing the vocabulary in terms of SE “sentences” of the form:
  
  – <Subject> <Predicate (verb)> <Object>

• The S/P/O organization has been very useful:
  
  – Systems engineers prefer using a controlled vocabulary of English to using UML/SysML
  
  – S/P/O sentences have the structure of a triple in RDF

• The challenge has been how to adapt an ontology-friendly organization of a controlled vocabulary (S/P/O sentences) as a rigorous interpretation of UML/SysML
Our IMCE Ontology Documentation doesn’t show this important information (currently)
Even OMG standards do not show this either…

Figure 7.9 - Classifiers diagram of the Kernel package
JPL’s IMCE MagicDraw Toolkit can…
(but it gets crowded quickly!)

package Example [ Class2 ]

- «metaclass»
  - Classifier
    - +redefinedClassifier
      - 0..* [(subsets redefinedElement)]
      - A_redefinedClassifier_classifier
        - 0..* [(subsets redefinedElement)]
    - +general
      - 0..* [(result = self.parents())]
      - A_general_classifier
        - +classifier
          - 0..*

- «metaclass»
  - RedefinableElement
    - +redefinedElement
      - 0..*
    - +A_redefinedElement_redefinableElement
      - 0..*
        - +redefinedElement
          - 0..*
            - (readOnly, union)
A motivating example...

Example of a controlled vocabulary sentence:
- **Subjects:** Function, Item, ...
- **Predicates:** sends, ...

A picture...

```
Send Telemetry (Function)  (sends)  Telemetry Message (Item)
```

```
Send Telemetry (Function)

somePortName_sends_: Telemetry Message (Item)

telemetry: Telemetry Message

SomeInterface

telemetry
```
How it is done under the covers…
(metamodelling)
Applying MOF2/OWL2 reification to all OMG-grade MOF2.4.1 models…

- Example for UML’s Element ownership relation
Object Property Reification: MOF2 & OWL2 similarities

- **Left = MOF2**
- **Right = OWL2**

**IMCE’s MOF2/OWL2 integration makes object property reification explicit**

```
package uml:element [ uml:element mof2/owl2 ]

<metaclass> Element
  0..1
  +/owner (readOnly,union)

/A_ownedElement_owner
  (readOnly,union)
  +/ownedElement 0..*

<metaclass> Element

uml:Element
  <rdfs:range>
  <owl:ObjectPropertyChainLink>
    (index = 1, isInverse)

<refiedObjectPropertySource>
  uml:A_ownedElement_owner.owner
    <rdfs:domain>

<refiedObjectPropertyClass>
  uml:A_ownedElement_owner
    <rdfs:domain>

<refiedObjectPropertyTarget>
  uml:A_ownedElement_owner.ownedElement
    <rdfs:range>

<owl:ObjectPropertyChainExpression>
  uml:A_ownedElement_owner

<owl:ObjectPropertyChainLink>
  (index = 2)
```

2011-12-13
The scope of MOF2/OWL2 Reification needed for SysML’s BDDs and IBDs

Extension = This would be Element @ M3 if we could represent this concept.
The notation for this relationship is the circled- “+” (cross) line.
Ontology Summary

Integrated Model-Centric Engineering

2011-12-12

Table of Contents

1. Introduction
2. Foundation Ontologies
   2.1. Base Ontology
   2.2. Mission Ontology
   2.3. Analysis Ontology
   2.4. Project Ontology
3. Discipline Ontologies
   3.1. State Analysis Ontology
   3.2. State Analysis Functional Design Ontology
   3.3. Electrical Ontology
4. Application Ontologies
   Index

Non-OMG vocabularies

• Currently (Dec. 2011):
  – 4 foundation ontologies
  – 3 discipline ontologies

• Each ontology includes:
  – Vocabulary definition (*.owl)
  – Reification mapping (*.mapping.owl)

• We use continuous integration techniques to statically analyze well-formedness properties of MOF2/OWL2 vocabularies & mappings
Results of static analysis of MOF2/OWL2 vocabularies & reification mappings
• Compute all possible combinations of legal sentences:
  – Given a subject => legal predicates
  – Given a subject + predicate => legal objects

• ...

Compiled MOF2/OWL2 vocabularies
• A MOF2.4.1-compliant metamodel for OWL2
  – Use net.sourceforge.owlapi (v3.2.3) to import/export OWL2 models

• QVT Operational transformations for:
  – Applies the MOF2-OWL2-mapping (meta) ontology to convert:
    • MOF2.4.1 => OWL2 (with MOF2-OWL2 reification)
    • OWL2 (w/ MOF2-OWL2 reification) => MOF2.4.1

• Additional tool support based on SPARQL, Pellet and SESAME

• Current status (Dec. 2011)
  – OMG UML 2.4.1 (MOF2.4.1) => OMG UML2.4.1 (OWL2)
  – OMG SysML 1.3 (MOF2.4.1) => OMG SysML1.3 (OWL2)
  – + similar for all OWL2 reified vocabularies defined at JPL

  – OWL2 w/ MOF2-OWL2 reification => MOF2.4.1 profiles
    • This applies to all “bundled” OWL2 reified vocabularies defined at JPL
• **What:**
  - Modeling structure, ontologically
  - Currently called the “Block Specific Type” (BST)

• **How:**
  - Model-to-Model transformations to expand recursively the structure of a block based on inheritance & specialization
  - Recursive expansion may be a better term if not obscure

• **Example: modeling design variants**
  - Modeling variants of a S/C instrument package
    • light science option (A only)
    • rich science option (A, B, C)

• **Recently (Dec. 2011)**
  - Use BST to define the semantics of property reference “dot” path notation
Suppose A, B, C, V are elements in a common model...

We need A except for the 1st part p of A, p[1], where for its r sub-part, p[1].r, we want to change the type r's x value property, p[1].r.x, from V to V'.

In textual notation: change the type of p[1].r.x from V to V'.

Proposed notation for an expanded feature reference path:

```
p_1.r.x : V' [1][specificFeature = x]
```
Recursive expansion using Block & Property-Specific Types
Recursive Block Specific Type expansion for resolving proposed reference paths
SysML1.4: Expanded Feature Reference construct & reference path notation

Proposal based on input from the SysML 1.4 RTF, including S. Friedenthal, N. Jankevicius

Segment #1 (designate 1 of [2])

Segment #2

Segment #3

[Segment#1, Segment#2, Segment#3]

TODO: <ExpandedFeatureReference> such that the notation on the left corresponds to the ordered sequence of reference path segments on the right.
1: Enhance SysML Activities for ontological behavior modeling

2: Define a SysML diagram for ontological behavior occurrences (a trace of a SysML Activity)

(this would require using DI/DD to specify this new SysML diagram)