A Model Driven Approach to Semantic Integration

Evan Wallace
Manufacturing Systems Integration Division
Manufacturing Engineering Laboratory
Technical and Semantic Integration

- **Technical Integration** requires choosing and consistently implementing protocols and on-the-wire data formats so that data can be exchanged without error between systems.

- **Semantic Integration** requires specification of the meaning of the data such that data successfully delivered will then be interpreted the same way by all communicating parties.

- Both types of integration are necessary for successful, error free communication. But the qualities that are important for a language which specifies data formats/messaging and the qualities that are important to specifying semantics are different and often conflicting.
Language characteristics

• Schema languages
  – Assume different names indicate different entities/objects (Unique Names Assumption)
  – Assume completeness of model and data (Closed World Assumption)
  – Assume single interpretation of model and data (single most specific class membership)
  – Assert only necessary conditions in their models
  – Specify constraints for validation checking only

• Semantic languages such as OWL DL
  – Do not assume different names indicate different entities/objects
  – Do not assume completeness of model or data
  – Assumes multiple interpretations could satisfy the model
  – Support specification of necessary and sufficient conditions for inference of class membership
  – Specify constraints supporting consistency checking
Claim

In cases where the domain of discourse to be supported by a standard has significant complexity and the expected life span of a specification is measurable in more than two years, then the development of a separate semantic model will be worth the cost.

Note – when a standard is supporting multiple views/uses, then even though the domain of discourse may appear simple for a single view, the unified view is likely to be complex.
Corresponds to CIM 3 layer architecture

NIST Architecture

 Semantic Model
  Formal Business Process Model
  Reference Ontology
  Business Rules

Semantic Mapping

Technical Interaction Schema

CIM layers

Information

Contextual (profile)

Message Syntax

Grid-Interop 2009
An Architecture

- **Process Model** describing interactions, their order, and content.
- **Reference Ontology** describing the domain of discourse – things and their properties which would be described by data exchanged in interactions.
- **Business Rules** adding constraints to the interactions and data exchanged.
- **Semantic Mapping** relating elements of the above models to elements in the Technical Interaction Schema.
- **Technical Interaction Schema** specifying how data will be encoded and exchanged using a particular technology.
Formalizing Semantic Artifacts

Provides benefits:

- Information technology independence,
- Enable capture of more semantics,
- Reduce ambiguity,
- Increase flexibility and extensibility,
- Enable design time analysis of semantics,
- Potential for automated support for model comparison and evolution,
- Potential for automated testing of interoperability even across technical interaction schemas,
- Potential for use of exchange technologies (e.g. rdf/xml) that bring the flexibility of the semantic model to the technical integration.
Languages for Semantic Artifacts

Process Models:
- UML sequence and collaboration diagrams
- BPMN, BPMN2 or
- An OWL2 vocabulary

Reference Ontology:
- UML class diagrams
- OWL2 or
- both

Business Rules:
- Semantic Web Rule Language (SWRL)
- Newer OWL-ish rule syntaxes (see OWLED 2009)
- SILK Hyper LP language/Rule Interchange Format
- Constrained natural language
Languages for Semantic Artifacts

Semantic Mapping:

Possibilities:
- MOF Query, Views, Transformations (QVT)
- MOF LON

But these work with MOF-based models, and we have had trouble finding tools.

Other languages and vocabularies exist (including one developed at NIST), but they are research efforts.

Need something with tool support, standard mappings to particular technical interaction languages, and isn’t overly difficult to learn/use.
NIST Application: Automated Integration
NIST Application: Testing

Semantic Model

Business Process Model

Reference Ontology

Business Rules

Knowledge Base

Flow Validation

Content Validation

Assertion Validation

Consistency Testing

Message Assertion Base

Sent Message

Message Interpretation Tool

Grid-Interop 2009

National Institute of Standards and Technology • U.S. Department of Commerce
Challenges

- Perception of semantic languages as academic and complicated,
- Lack of community expertise with semantic languages,
- Lack of a clear choice of syntax/notation for sharing semantic models with domain experts,
- Lack of a mature mapping language with tool support.
Conclusions

• PAP 8 CIM for Distribution Management is essentially using our approach
• Potential is great for improved model management using a semantic model driven approach
• Some research, standardization, and development is needed for mapping languages
• We are interested in working with folks in applying this sort of approach to the Smart Grid, as well as working with folks in related research, and/or application of existing tools.