OntologySummit2009: Toward Ontology-based Standards

EADS inputs based on Inputs from research activities performed by EADS and LIRIS
Presented by Nicolas Figay, EADS IW - Parisa Ghodous LIRIS
Potential usage of ontology for standard

- **Annotate schemas with information not modeled in APs**
  - such as modules, UoF, CC, Conformance Option, definition, AP/Module structure
  - ⇒ Application protocols becoming knowledge based that can be “queried”, validated from a logical point of view, displayed with visualization tools, enriched and annotated, etc
  - ⇒ Modules and usage recommendations (e.g. DEXs, RDL, Services) that can be “cross-checked” (federated models through annotation)

- **Distributed semantic Store available for intelligent agent**

- **Mapping rules formalized using DL**
  - • most of the time OK going from ARM to AIM

- **Insuring semantic preservation and coherency for different formalization of the same manufacturing concepts** (extended hyper models)
  - • e.g. EXPRESS, UML, OWL, XML Schema, Programming languages, service description language…

- **Basis for producing frame to insure coherency of formal description of family of standards**
  - • e.g. AP214, PDT.net, PLM services, VDA Ecr…

- **Formal validation and improvement of existing protocols**
  - • E.g. rationalize the way xAssignment and xyRelationship entities are produced
Issues on developed semantic web technologies

- Limitation of Descriptive Logic – extensions required for
  - Computation
  - Equivalence of constructions
  - Semantic service missing
  - Breakdown constructs not available while required for product engineering

- Improvement and extension of formal explicit visual representation languages
  - Important to make the link between representation for people, and representation for machines, with good alignment
  - E.g: BPMN vs XPDL/BPEL, starting from different conceptual models and difficult to align, while describing the same things.

- Better inter-relation between languages
  - Extending and reusing standards and not recreating with huge inconsistent overlaps.
Drawback slides

More information on previous projects and thinking relating ontology and standards
Knowledge base as semantic graph (ATHENA)

• Formalized information within AP which are not within the EXPRESS model
• Reasoning not required, just relate heterogenous distributed source of knowledge
• Should allow usage of SparQL to “query” Application Protocol document

ATHENA:
• opportunity to work on:
  – STEP binding to produce Semantic Graphs based on OWL Full from Express and P21 files
  – Enriched with external knowledge (Definition, UoF, CC)
  – Navigated a graphical way (OWLviz, Jambalaya, etc)
⇒ automated transformation from EXPRESS not appropriate
  difference of expressivity between EXPRESS and OWL
  • no construct exist in EXPRESS to deal with relations/properties
  • While relations/properties core construct of OWL
    • allows dealing very simply with xyRelationship, xAssignment entities defined in STEP application protocols.
    • Allows dealing simply with the SELECT types in EXPRESS.

• Choice not to use OWL DL
  • DL constraint (partitioning of things as individual, property or class) not appropriate for AP (coexistence of product family, product and product instance within the same model)
  • Establishment of a semantic graph brought a sufficient value without adding some complexity trying to produce DL models, with not yet mature open source engines.
Logically federated models
(based on OWL2 and Large Triple stores …)

- Schemas and AP content published as OWL models for annotation of distributed resources
- Allowing SparQL querying, PLM data annotation, usage of Reasoning Engine, intelligent agents, pervasive product and standard knowledge for people and tools

WITH OWL2, new large triple store, improved reasoning engines
- More mature modeling tools
  - Protégé 4
- More mature reasoning engines
  - Pellet, Fact++
- Emerging large RDF stores
  - To deal with large amount of data for Aerospace product description
- Some improvement
  - better management of import/export
  - annotations and subProperties

=> reconsidering OWL DL as a target is today more relevant

Potential usage
- formalize logical mappings in OWL
- take advantage of reasoning engines for transformation through inferencing
- logic validation of mapping between models
Coherency of representations of a same knowledge using different formalism (languages) for different purpose and automates (software)

- EXPRESS us UML us XML Schema us RDF us OWL us DL us programming language us SQL
- Describe us Reasoning us Computing us Structure us Manage
- Classification us Decomposition us Aggregate
- Most of constructs provided by ontology insufficient for Product Design

Some open questions
- all the rules formalized with EXPRESS can't be formalized in OWL
- descriptions of operations and functions on literals is not supported by OWL.
- How to establish equivalence of models where some set of literals is equivalent to another as it can be obtained by functions?
- An example is definition of a circle, which can be obtained and is fully defined by different sets of parameters and associated way to construct it.
- If two modeling languages are not using the same, do we consider they are not equivalent?
- And when willing to transform the data from one to the other, reasoning is not sufficient, as we also need to ... calculate.

Studying OMG MDA, UML, XMI and MOF
- both EXPRESS and OWL very poor with specialized relationships required for Product Development.
- Nor EXPRESS nor OWL are providing dedicated constructs for breakdowns, being aggregates or compositions
- while UML do
- Within application protocols, such constructs proposed (metamodel level) while part of modeling constructs with UML or SysML.
- Idem for OWL
- In the reverse, UML being for design so very poor to deal with individuals
- Way to formalize logic constraints is very complicated compared to OWL.
- XMI is syntactic, not semantic
- Why to use within Manufacturing community EXPRESS, OWL or UML to describe the same world.
- A response is may be "let's use together for appropriate usage".
- Most of the studies conclusion is: "languages are not equivalent but complementary". So why to choose?
Impedance misMESS – semantic preservation

- Language mappings against coherent multi-formalism and multi-representation of a domain of knowledge according different viewpoints to address for enterprise technical application

- One issue,
  - "Impedance mismatch" (Object Relational Mapping community)
  - "we loose information when translating"
- Over numerous languages to use => going toward "impedance misMESS"?
- Semantic preservation going from a representation of the same reality using one language to a second representation of the same reality using another language
- Semantic preservation more and more important
  - To avoid "formal language silos"
  - To produce set of representations using different languages but insuring coherency of these representations
  - Effective usage of the produced formal models is also expected.
- Using more and more using COTS
- Focusing on our core activities
- Our providers are not using the same language than us
- How to deal with reconciliation of enterprise, application users, software product and developers viewpoints and make them communicate together?

Industrial context and viewpoint
- Today several initiatives are trying to define a framework to deal with numerous manufacturing eBusiness standards (ASD SSG, EADS SSC), with a difficulty due to usage of heterogeneous modeling languages based on different paradigms.
  - Encompass
    - Organization, Process, Information and ICT
    - Data, Services (set of published and consumed operations), Process (behavior)
- What about set of coherent standardized languages
  - covering complete spectrum of needs and phases of application lifecycle
  - Selecting already existing and relevant languages
  - Using them together
- EXPRESS, UML and OWL are candidates to be part of such a set
  - Product Data Exchange
  - Software engineering for component and model based software engineering
  - Semantic WEB
- But should be completed by emerging SOA and BPM related languages which are not information centric, but are focusing on other aspect than information models.
- It is nevertheless a pity that SOA W3C standards are syntactic, and not semantic.
- What about W3C recommendation for semantic services?
- Can we imagine to “ontologize” existing W3C standards
1. Sharing data objects semantics between designers
   ⇒ STEPMOD electronic technical dictionary in English

2. Sharing semantics for applications
   ⇒ transformation in XML for Shared Electronic Documents on WEB
   ⇒ transformation in OWL for knowledge sharing and semantic PLM on the WEB
   ⇒ UML transformation for application engineering
   ⇒ Completion with services and processes: Model Driven Approach based on PIM4SOA

Semantic sharing

- Neutral Shared Technical Dictionaries For Virtual Enterprise
- Reference Data Libraries In OWL
- Knowledge Sharing Semantic PLM on the WEB
- Open Execution Platform For Collaboration Space Based on ICT standards
- Shared Electronic Documents On the WEB (Part 28)
- Formal Explicit Description Of Entities In Express
- Platform Independent Model For Service Oriented Architecture In UML
- PLM Services
  - Configuration Management & Collaboration Processes
  - Business Entities
  - Business Services
  - Business Processes
  - Information repository
  - Application Server
  - Process Execution Engine
Semantic Repository: usage of reasoning engine
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:label</td>
<td>rdf:type</td>
<td>owl:AnnotationProperty</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>rdf:type</td>
<td>owl:AnnotationProperty</td>
</tr>
<tr>
<td>owl:versionInfo</td>
<td>rdf:type</td>
<td>owl:AnnotationProperty</td>
</tr>
<tr>
<td>xsd:string</td>
<td>rdf:type</td>
<td>rdfs:Datatype</td>
</tr>
<tr>
<td>Cylindricity_Tolerance</td>
<td>rdf:type</td>
<td>owl:Class</td>
</tr>
<tr>
<td>Cylindricity_Tolerance</td>
<td>rdfs:subClassOf</td>
<td>T2_Geometric_tolerance</td>
</tr>
<tr>
<td>T2_Geometric_tolerance</td>
<td>rdfs:subClassOf</td>
<td>CC19_configuration_controlled_process_planning_of_components_and_assemblies_with_3D_shape_representation_including_f</td>
</tr>
<tr>
<td>T2_Geometric_tolerance</td>
<td>rdfs:subClassOf</td>
<td>CC14_feature_based_design</td>
</tr>
<tr>
<td>T2_Geometric_tolerance</td>
<td>rdfs:subClassOf</td>
<td>CC20_Conformance_Class_for_Data_Storage_and_retrieval_systems</td>
</tr>
<tr>
<td>T2_Geometric_tolerance</td>
<td>rdfs:subClassOf</td>
<td>CC13</td>
</tr>
<tr>
<td>T2_Geometric_tolerance</td>
<td>rdfs:subClassOf</td>
<td>CC12_Process_planning_of_components_with_form_feature_and_tolerance_data</td>
</tr>
<tr>
<td>T2_Geometric_tolerance</td>
<td>rdfs:subClassOf</td>
<td>CC15_Feature_based_design_with_flexible_feature_placement</td>
</tr>
<tr>
<td>T2_Geometric_tolerance</td>
<td>rdfs:comment</td>
<td>G_geometry</td>
</tr>
<tr>
<td>T2_Geometric_tolerance</td>
<td>rdfs:comment</td>
<td>Geometric tolerance</td>
</tr>
</tbody>
</table>

This unit of functionality specifies the representation of geometric tolerances with a datum reference, such as parallelism or coaxiality, geometric tolerances without a datum reference, such as straightness or flatness. The tolerances are defined relative to the target object. This unit allows the definition of up to three datum references that are either single datum references, compound datum references consisting of additional datum targets and additionally the specification of tolerance zones. The definition of the presentation of the Tolerance information associates an annotation UoF (D2).

Organization_relationship.related       | rdf:type                               | owl:ObjectProperty                     |
Organization_relationship.related       | rdf:type                               | owl:FunctionalProperty                 |
Organization_relationship.related       | rdfs:comment                           | The related specifies the second Organization in an Organization_relationship. NOTE The semantics of this attribute are defined by the 'relation_type'. See organization_relationship to organization for the application assertion. Each Organization_relationship is related to an Organization. Each Organization is referenced by zero or more Organization_relationship objects as related. NOTE 1 The relation

http://www.plm-interop.net/semanticRepository/AP214ARMed2.owl#CC15_Feature_based_design_with_flexible_feature_placement
Extended Hyper models in several “Grounds” with associated meta-workbench

Hyper Modèles dans divers sols

 Méta-atelier UML2

 Méta-atelier OWL