Ontologies and Model-Based Systems Engineering

Steven Jenkins
Principal Engineer
Systems and Software Division
Jet Propulsion Laboratory
California Institute of Technology

Copyright © 2010 California Institute of Technology. Government sponsorship acknowledged.
Models Are Information Structures

- Every model, whether it’s a differential equation, a simulation, or a SysML drawing, organizes concepts and properties into meaningful relationships.
- These concepts, properties, and relationships can be unique to a model, or they can be common to a family of models.
- To the degree that they’re common:
  - models can be compared, contrasted, and reused.
  - engineers can understand what’s communicated by a model without retraining.
  - engineers can focus on creating and understanding, not explaining.
- Can we come up with a common set of concepts, properties, and relationships for systems engineering? **Absolutely.**
  - That doesn’t mean everyone is saddled with one-size-fits-all.
  - The common set is for common things. Unique things are handled by extensions.
- These concept, property, and relationship definitions are called an **ontology**.
Names vs Classifiers

• Suppose we have a thing named “Star Tracker A”.
• Without an ontology, what can we infer about this thing?
• We just have to “know” that it’s probably
  – a piece of hardware with a reference identifier
  – a mechanical component with mass properties
  – a electronic component with power properties
  – a thing with interfaces
• Too much of this knowledge is implicit. If we wanted to automate production of a master equipment list, how would the algorithm know to include this thing and not another thing called “flight software”? How would it ask for the thing’s mass?
• This is easy to solve if we keep separate
  – what the thing is called (to a first order, we don’t care)
  – what kind of a thing it is (from a controlled vocabulary of concepts)
• A hierarchy of components provides a structure for properties
An Example Classification Hierarchy

Each concept is a specialization of the concept above it.
Some Example Properties

Each concept has all the properties of all its ancestors, plus its own unique properties.
Building It Out

Single, well-known property name for common properties (e.g., mass, power load)

Component
- name, id

HardwareComponent
- reference designator

FlightHardwareComponent
- mass, center of mass, moments of inertia

FlightAvionicsComponent
- power load

StarTracker
- sensitivity

WhizBangMkIVStarTracker
- whiz factor

Transponder
- power output

UltraTurboTransponder
- turbo boost
Relationships Are Also Properties

Integrated Model-Centric Engineering

Jet Propulsion Laboratory

Relationships Are Also Properties

Component Interface Function Requirement

compresents performs specifies

Interface Component Function

Requirement

2010-03-10 USC CSSE ARR
Putting It To Use

• We can express facts in these terms and store them in a repository:
  – $x$ has type WhizBangMkIVStarTracker
  – $x$ has name “Star Tracker A”
  – WhizBangMkIVStarTracker is a subclass of StarTracker
  – StarTracker is a subclass of FlightAvionicsComponent
  – and so on….
  – Facts are expressed in triples of the form $(subject, predicate, object)$.
• We can then ask simple questions like “What is the sensitivity of the WhizBangMkIVStarTracker named ‘Star Tracker A’?”
• If the repository can draw inferences, then we can ask things like “What is the sensitivity of the StarTracker named ‘Star Tracker A’?”
• Then the master equipment list is simple: “Find all FlightHardwareComponents and print their names and masses.”
• Note that these are mission-independent (i.e., reusable) procedures.
The Semantic Web World

- **Standards**
  - Resource Description Framework (RDF)
    - Statements of the form \((subject, predicate, object)\)
    - Simple class hierarchies
  - Web Ontology Language (OWL)
    - RDF vocabulary for formal logic
  - SPARQL Query Language for RDF
    - Powerful language for querying RDF/OWL databases

- **Technology**
  - Ontology editors (Protégé, TopBraid Composer, etc.)
  - Knowledge repositories (Sesame, Oracle Semantic Database, Mulgara, etc.)
  - Application frameworks (Sesame, Jena, TopBraid Suite, etc.)

- **Community**
  - Journals, conferences, workshops
  - W3C standards working groups
  - ODM
Semantic Web Example

- OWL information base describing formulation-phase design information for Phoenix mission to Mars
  - 9583 triples
  - Built on multiple ontologies, including rdf, base (JPL), and mission (JPL)
  - Stored in an open-source Sesame2 repository

- Simple query: find id and name for the component whose name is “Spacecraft System”

<table>
<thead>
<tr>
<th>SPARQL Query</th>
<th>Result</th>
</tr>
</thead>
</table>
Semantic Web Example

- More complex query: find id and name for all functions performed by the component whose name is “Spacecraft System”

<table>
<thead>
<tr>
<th>SPARQL Query</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F.1.2.2.3 Execute Commands</td>
</tr>
<tr>
<td></td>
<td>F.1.2.2.4 Maintain Communication</td>
</tr>
<tr>
<td></td>
<td>F.1.1.2    Transport Payload to Martian Surface</td>
</tr>
<tr>
<td></td>
<td>F.1.2.2    Maintain Operational State</td>
</tr>
</tbody>
</table>
Semantic Web Example

- Still more complex query: find id and name for all requirements specifying functions performed by the component whose name is “Spacecraft System”

<table>
<thead>
<tr>
<th>SPARQL Query</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L.3.FS.42  First TCM Time</td>
</tr>
<tr>
<td></td>
<td>L.3.FS.43  Last TCM Time</td>
</tr>
<tr>
<td></td>
<td>L.3.FS.147 Command Durations During Cruise</td>
</tr>
<tr>
<td></td>
<td>L.3.FS.144 Command Processing Capability</td>
</tr>
<tr>
<td></td>
<td>…</td>
</tr>
</tbody>
</table>
## Some Relative Priorities

<table>
<thead>
<tr>
<th>Topic</th>
<th>Semantic Web</th>
<th>UML/SysML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics</td>
<td></td>
<td>++</td>
</tr>
<tr>
<td>Tooling</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Querying</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Inferences</td>
<td>++</td>
<td></td>
</tr>
<tr>
<td>Interchange</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Expression</td>
<td>++</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>+</td>
<td>++</td>
</tr>
</tbody>
</table>
Ontologies and SysML

• SysML has an ontology, if not by that name
  – Block, Interface, Activity, Requirement, etc.
• An organization seeking to build long-lived, interchangeable models will likely need to build more ontological structure beneath these high-level concepts:
  – Work Breakdown Structure, Hardware, Software, Stakeholder, Concern, etc.
  – Plus specialized associations (authorizes, represents, specifies, etc.)
• JPL is developing its ontologies using OWL2 and translating them to SysML conceptual models and profiles.
  – To us, OWL2 (and the interoperability it implies) are more fundamental than SysML
  – Model-based engineering includes many domain-specific models and tools, including system models and SysML tools
• We expect to exchange model data between SysML and semantic tools for analysis, validation, product generation, etc.