Ontological Product Modeling for Collaborative Design

Conrad Bock, Xuan Zha
Hyowon Suh, Jeahyun Lee
December 11, 2008
Overview

- Goals and approach.
- Desired capabilities.
- Background
  - Ontology modeling
  - Modeling languages
- Proposed Solution
- Summary
Overall Goal and Challenges

- Improved support for collaboration in the design process.
  - Right knowledge at the right time.
  - Avoid backtracking and rework.
  - Especially in global economy.

- Challenges:
  - Combining and refining independently-developed product descriptions.
  - Alignment in interpretation of product descriptions.
General Approach

- Apply ontological techniques …
  - Open world semantics: Multiple product models can describe the same product and be checked for consistency.
  - Rigorously-defined interpretation of ontological languages.

- … and model-driven techniques …
  - Engineering-friendly domain languages specialized from ontological languages.
Applied to Product Modeling

- ... to generalized notions in product modeling:
  - Product models describe (some portion of) the *total system* of device and environment in which it is used.
  - Behaviors include the entities involved in them. Models can describe a portion of a behavior or its entities or both.
  - Interconnections between components have same capabilities as components.
Product Models

- *Product models* describe any aspect of total systems (environment and device).
  - Environment (requirements)
  - Device (designs)
  - Or both.

- No limit on how much or how little of the environment and/or device is described.
- Treat product models as *partial descriptions* of total systems (environment and/or device behavior).
Product Taxonomies

- Specialized model includes the general model.
Interconnections (Not)

- Need connections in the context of an individual assembly.

- Power to wheels on different car than engine

- Individuals:
  - Mary’s Car
    - Engine in Mary’s Car
    - Wheel in Mary’s Car
  - John’s Car
    - Engine in John’s Car
    - Wheel in John’s Car

- Assembly Model:
  - Car
    - Wheel
    - Engine
Interconnected Elements

- Connections in context.
- Reuse of other assemblies.
Interconnected Subelements

- Interconnections between elements of elements ("ports").
Interconnection Inheritance

- Inherit, add, specialize interconnections in taxonomy.
Interconnection Decomposition

- Interconnection has subassemblies and interconnections of its own.
Interconnection Decomposition

- Reusing the same relational decomp.
Alternative Relation Decomp

- Taxonomy of assembly relation decompositions.
Interconnections between Interconnections

- Interconnections can be interconnected.

![Diagram showing interconnections between units](image-url)
Behaviors as Interconnections

- Behaviors relate the objects participating in them.
- Plate and bracket participate in a behavior that keeps their relative position constant.
Alternative Decompositions of Behavior Connections

- Behavior-constrained taxonomies

A

A 1.1

Plate

Bracket

No Movement

RelObj1 :

Fixed Relative Position

RelObj2 :

RelObj1 :

Bolt

Nut

RelObj2 :

Rivet

RelObj2 :

A 1.2

Plate

Bracket
Ontology

- Two kinds of information modeling:
  - Modeling software that carries and manipulates information (*software modeling*).
  - Modeling things that information is about (*ontology modeling*).

- Differ in their styles of classification.
  - Software: classes are “factories” from which software objects are created.
  - Ontology: classes are categories of individuals.
Ontology

- Formalized with set theory.
  - Members of the sets are actual things.
  - \textit{Classes} = rules for membership.

- Rules for membership can be about:
  - One, some, or all aspects of things.
  - Things from the past, present, future.
  - Real, intended, or only imagined things.
  - Physically possible or impossible things.
  - Things with a lot or little in common.

- Power from separating membership rules from members themselves.
Reasoners can operate on classes, without using members.
Ontological Product Modeling

- Classes of what?
  - Physical things, real or intended (cars with serial numbers)
  - Behavior occurrences (John commuting to work on May 18, 2008).

- Members must be the same kind of thing to support reasoning.

- Behavior occurrences involve individual physical things ...

- ... but individual things are involved in many behavior occurrences.
Ontological Product Modeling

Product Models

- Elevation below 5000 m (Requirement)
- Involves dev weighing less than 2000 kg (Design)
- Satisfy only requirements
- Satisfy both
- Satisfy design

Total System Behavior Occurrences (only examples shown)

- Using car with serial# 56678, weighing 2500 kg, at 2000 m.
- Using car with serial# 2345, weighing 1500 kg, at 2000 m.
- Using Mary’s car, weighing 1800 kg, at 1000 m
- Using John’s car, weighing 1000 kg, at 3000 m.
- Using John’s car, weighing 1000 kg, at 6000 m.
- Using Joe’s car, weighing 1500 kg, at 7000 m.

- Classes of behavior occurrences.
Model Levels ("metalayers")

- Each level satisfies the one above it.

**Individuals (M0)**
- John’s Car
- Mary

**Model (M1)**
- Car
- Person
- Drive
- Commute

**Modeling Language (M2)**
- Class
- Relation
- Drive

**Meta Language (M3)**
- Class
- Relation

- Using John’s car, weighing 2000 kg, at 2000 m
- Mary taking the train from work to home, April 24, 2007, 5-5:30pm ET
- Engineer uses ontology language directly.
- M1 product models are classes, can be specialized in M1 and instantiated at M0.
Modeling Language, No Ontology

- Engineer uses familiar language.
- Cannot instantiate and specialize M1 product models (they are individuals, not classes).
Ontology and Modeling Language

- Engineer uses familiar language.
- M1 product models are classes, can be specialized in M1 and instantiated at M0.
Requirements

- A product model might be only requirements, no designs (only about the environment, nothing about the artifact).

![Diagram showing environment, unspecified artifact, and water at different locations.](image-url)
Alternative Designs

- Different artifact designs satisfy requirements in different ways.
- Example: pump uses pressure to move water, Archimedes screw moves containers of water.
- The above behaviors (putting water under pressure, containing water) are specializations of the desired behavior (moving water) that specify more about the participants.
Alternative Designs

- M1 product model that only has requirements is refined to include alternative designs.
- Constrains total systems at M0.
Product model can be requirements and / or design.

M0 total system conforms to the M1 product model (consistency checking).
Requirements “roll down”

- Requirements refined along with design.
Interconnections

- Connectors relate part-whole relations:
  - to identify parts/subassemblies in each individual M0 whole
  - and link with another relation (powers).

- Identifies engine in each individual M0 boat.
- Creates powers link between engine and wheel identified in each individual car
- Identifies wheel in each individual M0 car.
Relations are classes (M1) of M0 links.
Can be specialized at M1 and have conforming M0 links between M0 entities.
Decomposing Relations

- Relations (classes) can have parts (M1).
- Conforming M0 links have M0 parts (interconnection decomposition).
Connectors establish M0 links within instances of the containing class.
Connectors

- Properties are “usages” limited to each individual car:
  - Part-whole relations use engines and wheels.
  - Connector uses powers relation.
- Multiple usages of the same subassembly.
Connector Inheritance

- Inherit connectors as relations.
- Specialize in subassemblies.
Behaviors are classes (M1) of M0 “executions”.
Can be specialized at M1 and have conforming M0 links (modeling behavior occurrences).
Behaviors as Relations

- Behaviors are relations between things participating in them (M1), conforming at M0.
- Applicable to kinematic assemblies.

Relative rotation of Axle #5467 and Axle #2345
Mar. 5, 207, 5:34-5:55pm ET
Alternative Conforming Decompositions of Assembly Relation

- Behavior-constrained taxonomies:

  - **A 1.1**: Bolt : Plate \(\rightarrow\) Bracket
  - No Movement
  - RelObj1 : Plate
  - RelObj2 : Bracket

  - **A 1.2**: Rivet : Plate \(\rightarrow\) Bracket
  - Fixed Relative Position
  - RelObj1 : Plate
  - RelObj2 : Bracket

- **Behavior-constrained taxonomies**
Alternative Conforming Decompositions of Assembly Relation

- Connector specialized by restriction.
- From No Movement to BoltTogether.
Interconnections between Interconnections

- Connectors are part-whole relations.
- Can be connected.
Interconnections between Interconnections

- Two connectors between same part-whole relations.
- Connectors (as part-whole relations) connected by thermal connector.
Product Model and Artifact

M2

Product Model

Behavior

Class

1..*

involves

Requirement

Design

1..*
specifies

Artifact

M1

Car Design

Car

M0

Using John’s car, weighing 900 kg at 2000 m

John’s car
Form Metamodel

M2

Class

Form

Material

Geometry

Steel

Cone

M1

Stainless Steel

Right Circular Cone

M0

An ingot of stainless steel

A piece of wood with right circular cone shape
Material and Geometry Language

(M2)

(M1)

Steal

Block

Engine

Assembly

Car

Wheel

Light Hub

Hub

Aluminum

Part

Simple Part

Geometry

Material

/madeOf

/shapedLike

/assemblyOf

/subartifactOf

Artifact

Assembly

Part

/assemblyOf

engine

_inCar

block

_inEngine

powered

wheel

_inCar

Hub

BRep

68
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>STEP</th>
<th>UML 2 / SysML</th>
<th>CPM 2 / OAM 2</th>
<th>MOKA</th>
<th>OPML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total system (device/environment)</td>
<td>X</td>
<td>X+</td>
<td>X</td>
<td>X</td>
<td>S</td>
</tr>
<tr>
<td>Full interoperability</td>
<td>X</td>
<td>X+</td>
<td>X</td>
<td>X+</td>
<td>S</td>
</tr>
<tr>
<td>Enables consistency checking/reasoning</td>
<td>X</td>
<td>X+</td>
<td>X</td>
<td>X</td>
<td>S</td>
</tr>
<tr>
<td>Composition/assembly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interconnection of elements</td>
<td>S-</td>
<td>S</td>
<td>S-</td>
<td>X</td>
<td>S</td>
</tr>
<tr>
<td>Multiple usages of the same kind</td>
<td>X+</td>
<td>S</td>
<td>v1: X</td>
<td>X</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>v2 : S-</td>
<td></td>
</tr>
<tr>
<td>Generalization/refinement</td>
<td>X+</td>
<td>S</td>
<td>X</td>
<td>X+</td>
<td>S</td>
</tr>
<tr>
<td>Relation/connector decomposition</td>
<td>X+</td>
<td>UML 2 : X</td>
<td>X</td>
<td>X</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SysML : S-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interconnections of interconnections</td>
<td>X</td>
<td>UML 2 : X</td>
<td>X</td>
<td>X</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SysML : S-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Behaviors as constraints on M0</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X+</td>
</tr>
<tr>
<td>Behaviors as relations/connectors</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

- **S**: Full support
- **S-**: Support with exceptions
- **X+**: Does not support, with exceptions
- **X**: Does not support
Summary

- Combine ontology and modeling languages:
  - Open world for combining partial product models and consistency checking.
  - Modeling for engineering-friendly languages.
  - Taxonomies at M2 and M1.

- Product models describe (a portion of)
  - Total systems (environment and/or device).
  - Behavior occurrences (including objects involved)

- Relations are Classes, Connectors and Behaviors are Relations.