

Ontology Integration and Interoperability (OntoOp)

OMG Analysis & Design TF

Fabian Neuhaus, Till Mossakowski, Michael Gruninger

OMG Technical meeting, Santa Clara, 2013-12-11

Scope of presentation

- Motivation for OntoIOP RFP
- High-level overview of goals and requirements
- What changed since the New Brunswick meeting?

The Interoperability Challenge

Truism: Ontologies are no silver bullets

Two ontologies (models, specifications) may be incompatible because

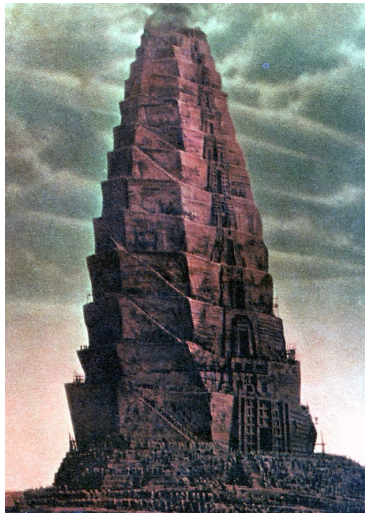
- Incompatible content
- Different languages

Diversity of KR Languages

- OWL, RDF, OBO
- UML class diagrams
- RIF (Rule Interchange Format)
- EER (Enhanced Entity-Relationship Diagrams), Datalog, ORM (object role modeling)
- the meta model of schema.org
- SKOS (Simple Knowledge Organization System)
- FOL, F-logic, Common Logic

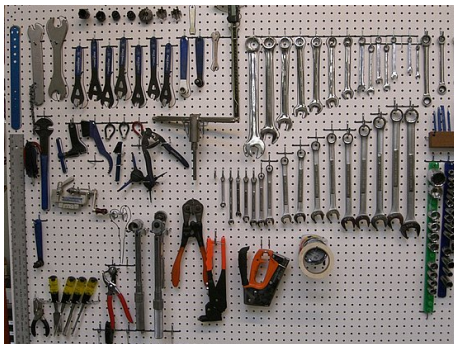
Diversity of Languages: Curse or Blessing?

Babelonian confusion?



Diversity of Languages: Curse or Blessing?

Set of tools!



Example: OMG's Date-Time Vocabulary (DTV)

The Date-Time Vocabulary is a heterogenous ontology:

- SBVR: very expressive, readable for business users
- UML: graphical representation
- OWL: formal semantics, computationally tractable
- Common Logic: formal semantics, very expressive

Example: OMG's Date-Time Vocabulary (DTV)

The Date-Time Vocabulary is a heterogenous ontology:

- SBVR: very expressive, readable for business users
- UML: graphical representation
- OWL: formal semantics, computationally tractable
- Common Logic: formal semantics, very expressive

DTV combines the advantages of different languages

Challenge for DTV

How does the UML DTV parts relate to each other?

- Are the SBVR axioms and the OWL axioms logically consistent?
- Is everything in the OWL ontology logically entailed by the Common Logic ontology?
- Synchronization has to be checked manually relying on intuition.

Challenge for DTV

How does the UML DTV parts relate to each other?

- Are the SBVR axioms and the OWL axioms logically consistent?
- Is everything in the OWL ontology logically entailed by the Common Logic ontology?
- Synchronization has to be checked manually relying on intuition.

DTV has the parts, but cannot glue them together to a whole

What is missing to address DTV's challenge?

A metalanguage that enables

- the combination of ontologies (that may be written in different languages) into a larger ontologies
- the specification the intended relationships between two different ontologies (e.g., that one is a fragment of another)

What is missing to address DTV's challenge?

A metalanguage that enables

- the combination of ontologies (that may be written in different languages) into a larger ontologies
- the specification the intended relationships between two different ontologies (e.g., that one is a fragment of another)

Goal of OntoOP: provide a language that enables these functionalities for ontologies, specifications, and models (OSMs).

Not Yet Another Ontology Language

- OntoOp does not look for a ‘Lingua Franca’
- OntoOp asks for a *metalanguage* for talking about OSMs

Requirements: Relationships between OSMs

Proposals shall provide a specification of a metalanguage for the following relationships

- **logically heterogeneous OSMs**
- **modular OSMs**
- **module extraction, approximation**
- **links** (imports, interpretations, equivalences, renamings, alignments) between OSMs
- **combination** of OSMs along links

Goal

- The OntoOp metalanguage is not supposed to ‘magically’ create interoperability between OSMs.
- It enables the formal specification of the relationships between OSMs.
- This can be used by tools (like Hets) to create interoperability (when possible).

What changed since New Brunswick? I

Scope is made more explicit:

- connections between logical theories
- logical theories: ontologies, specifications, models
- requirement: language must be logic-based
- proposals are required to illustrate solution with a small set of important languages
- solution is required to be expendable

What changed since New Brunswick? II

- 5 use cases for ontologies
- 2 use cases for specifications
- 3 use cases for models

Thank you