The Distributed Ontology, Modeling and Specification Language (DOL)

Till Mossakowski\textsuperscript{1,2} Oliver Kutz\textsuperscript{1} Mihai Codescu\textsuperscript{3}

\textsuperscript{1}Collaborative Research Centre on Spatial Cognition, University of Bremen
\textsuperscript{2}DFKI GmbH Bremen
\textsuperscript{3}University of Erlangen-Nürnberg

WoMo 2013, La Coruna
The Big Picture of Interoperability

Knowledge Infrastructure
- Concepts/Data/Individuals
- Ontology
- Ontology Language/Logic

Service-Oriented Architecture
- Service
- Service Description
- Service Descr. Language

Smart Environment
- Device
- Target Description
- Target Descr. Language

Data
- Concepts/Data/Individuals

Models
- Ontology

Metamodels
- Ontology Language/Logic

Knowledge
- Software Agents

Motivation OntoOp DOL Modular and Heterogeneous Ontologies Distributed ontologies Semantics Conclusion
The Big Picture of Interoperability
Motivation: Diversity of Ontology Languages

A great diversity of ontology languages is in use:

- 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

It is common practice to **informally annotate** OWL ontologies with FOL axioms (e.g. Keet’s mereo-topological ontology, Dolce Lite, BFO-OWL)
Use Case: OMG’s Date-Time Vocabulary

- date-time vocabulary is formulated in different languages: SBVR, Common Logic, IKL, UML+OCL, OWL
- different languages address different audiences
  - SBVR: business users
  - UML+OCL: software implementors
  - OWL: ontology developers and users
  - Common Logic, IKL: (foundational) ontology developers and users

- How can we
  - formally relate the different logical specifications?
  - specify the OWL version to be an approximation of the Common Logic version?
  - extract submodules covering specific aspects?
heterogeneous OWL/FOL ontologies

- e.g. an OWL ontology with some FOL axioms
- ... for use with an OWL reasoner, a FOL theorem prover, and a FOL model finder
  1. OWL reasoner for reasoning about the OWL part
  2. FOL theorem prover for proving consequences of the whole ontology
  3. FOL model finder for finding a model of the whole ontology
  4. FOL model finder for disproving consequences of the whole ontology
More Use Cases

- The use of RDFS or OWL to specify a taxonomy of sorts for a more expressive logic with many-sorted semantics.
- The use of Common Logic to express metadata concerning modelling assumptions for simulation (e.g. climate change) datasets (Datalog expressivity).
  - The Datalog theory describes the **closed world** of the observed dataset.
  - The Common Logic theory is **open-world** and describes the physical laws of the object of observation.
More Use Cases (Modeling and Specification)

- **UML model involving different diagram types**
  - check for semantic consistency
    - (relative to a given formal semantics)
- **temporal logic specification**
  - check against some process model
  - then refine this into some finite automaton
- **UML protocol state machine**
  - (possibly enriched with some UML sequence diagrams and OCL constraints)
  - refine to UML behaviour state machine
Motivation: Diversity of Operations on and Relations among Ontologies

Various operations and relations on ontologies are in use:

- matching and alignment
  - of many ontologies covering one domain
- module extraction
  - get relevant information out of large ontology
- approximation
  - model in an expressive language, reason fast in a lightweight one
- querying
- ontology-based database access/data management
- distributed description logics, $\mathcal{E}$-connections
  - bridges between different modellings
Need for a Unifying Meta Language

Not yet another ontology language, but a meta language covering:

- **diversity of ontology languages**
- **translations** between these
- **diversity of operations on and relations among ontologies**

Current standards like the OWL API or the alignment API only cover parts of this.

The Ontology, Modeling and Specification Integration and Interoperability (OntoIOp) initiative addresses this.
The OntolOp initiative

- started in 2011 as ISO 17347 within ISO/TC 37/SC 3
- now continued as OMG standard
  - OMG has more experience with formal semantics
  - OMG documents will be freely available
  - focus extended from ontologies only to formal models and specifications (i.e. logical theories)
  - request for proposals (RFP) to be issued in fall 2013
  - proposals answering RFP due in December 2014
- 50 experts participate, ~ 15 have contributed
- OntolOp is open for your ideas, so join us!
Requirements in the OMG RFP OntolOp

- provide a **meta-language** for:
  - logically heterogeneous ontologies
  - modular ontologies
  - module extraction, approximation
  - links (interpretations, alignments) between ontologies/modules
  - combination of ontologies along links
- provide an **abstract syntax** as MOF or SMOF model
- provide a **concrete syntax**
- provide a **formal semantics**
  - criteria for logics to conform with OntolOp
  - translations between these logics
- be **logic-agnostic**, e.g. ontologies consist of symbols and axioms
The Distributed Ontology, Modeling and Specification Language (DOL)

- has been prepared within ISO/TC 37/SC 3
- now continued as a proposal for the OMG RFP OntoIoP
  - DOL = one specific answer to the RFP requirements
  - there may be other answers to the RFP (but unlikely)
- DOL is based on some graph of logics and translations
- DOL has a model-theoretic semantics
  - semantics of an ontology:
    - logic
    - signature in that logic
    - class of models over that signature
- analysis and proof tools for DOL exist
An Initial Logic Graph for DOL

- **UML-CD**
- **Schema.org**
- **SKOS**
- **DDL^^\text{\text{O}W}L**
- **EL^^++** (OWL 2 EL)
- **DL-Lite\text{\text{R}}** (OWL 2 QL)
- **DL-RL** (OWL 2 RL)
- **SROIQ** (OWL 2 DL)
- **RDFS**
- **Prop**
- **OBO\text{\text{O}W}L**
- **RDF**
- **FOL\text{\text{E}C}O**
- **FOL\text{\text{E}C}O^\text{\text{F}O}L**
- **FOL^\text{\text{F}O}L**
- **FOL^\text{\text{M}S}**
- **bRDF**
- **CL**
- **CASL**
- **HOL**

**Colors:**
- **grey:** no fixed expressivity
- **green:** decidable ontology languages
- **yellow:** semi-decidable
- **orange:** some second-order constructs
- **red:** full second-order logic

**Arrows:**
- **subinstitute**
- **theoroidal subinstitute**
- **simultaneously exact and model-expansive comorphisms**
- **model-expansive comorphisms**

---

**Motivation**

**OntoOp**

**DOL Modular and Heterogeneous Ontologies**

**Distributed ontologies**

**Semantics**

**Conclusion**

**2013-09-15**

**Mossakowski**

**Distributed Ontology, Modeling and Specification Language (DOL)**
What is an Ontology Language / Logic?

We need the following ingredients:

- sentences, models, satisfaction
- signatures
- signature extensions, signature morphisms
- model reducts
- signature colimits

Details later . . .
Sample Use of DOL

- **DOL**
  - **ASK-IT ontologies**
    - **OWL**
  - **ASK-IT ontologies**
    - **Common Logic**
      - **Import**
  - **DOLCE**
    - **Common Logic**
      - **Import**

*Integrated ontology: ASK-IT ontologies + alignment + DOLCE*

*Serialization: CLIF*
Overview of DOL

1. modular and heterogeneous ontologies
   - basic ontologies
   - extensions, unions, translations, reductions
   - approximations, module extractions
   - minimization
   - combination
   - ontology bridges

2. distributed ontologies (based on 1)
   - ontology definitions (giving a name to an ontology)
   - interpretations (of theories)
   - equivalences
   - module relations
   - alignments
Basic Ontologies

- written in **some ontology language** from the logic graph
- semantics is **inherited** from the ontology language
- e.g. in OWL:

  ```
  Class: Woman   EquivalentTo: Person and Female
  ObjectProperty: hasParent
  ```

- e.g. in Common Logic:

  ```
  (cl-module PreOrder
   (forall (x) (le x x))
   (forall (x y z)
     (if (and (le x y)
               (le y z))
     (le x z))))
  ```
Extensions

- **$O_1$ then $O_2$:** extension of $O_1$ by new symbols and axioms $O_2$
- **$O_1$ then %mcons $O_2$:** model-conservative extension
  - each $O_1$-model has an expansion to $O_1$ then $O_2$
- **$O_1$ then %ccons $O_2$:** consequence-conservative extension
  - $O_1$ then $O_2 \models \varphi$ implies $O_1 \models \varphi$, for $\varphi$ in the language of $O_1$
- **$O_1$ then %def $O_2$:** definitional extension
  - each $O_1$-model has a unique expansion to $O_1$ then $O_2$
- **$O_1$ then %implied $O_2$:** like %mcons, but $O_2$ must not extend the signature

example in OWL:

```
Class Person
Class Female
then %def
Class: Woman EquivalentTo: Person and Female
```
References to Named Ontologies

- Reference to an ontology existing on the Web
- written directly as a URL (or IRI)
- Prefixing may be used for abbreviation

http://owl.cs.manchester.ac.uk/co-ode-files/ontologies/pizza.owl

c-o-ode:pizza.owl
Unions

- $O_1$ and $O_2$: union of two stand-alone ontologies (for extensions $O_2$ needs to be basic)
- Signatures (and axioms) are united
- model classes are intersected

algebra:Monoid and algebra:Commutative
Translations

- $O$ with $\sigma$, where $\sigma$ is a signature morphism
  - semantics: all models whose $\sigma$-reduct is an $O$-model
- $O$ with translation $\rho$, where $\rho$ is a logic translation
  - semantics: all models whose $\rho$-reduct is an $O$-model

ObjectProperty: isProperPartOf
Characteristics: Asymmetric
SubPropertyOf: isPartOf

with translation trans:SROIQtoCL
then
(if (and (isProperPartOf x y) (isProperPartOf y z))
  (isProperPartOf x z))
%%% transitivity; can’t be expressed in OWL together
%%% with asymmetry
Reductions

\[ \sigma \text{ is a signature morphism} \]

\[ \text{semantics: the } \sigma\text{-reducts of all } O\text{-models} \]

\[ \text{forall } x,y,z \]:

\[ 0 + x = x \]

\[ x + (y + z) = (x + y) + z \]

\[ x + \text{inv}(x) = 0 \]

\[ \text{forall } x,y,z \text{, } x = x + 0 \]

\[ O\text{-hide along } \mu \text{, where } \mu \text{ is a logic projection} \]

\[ \text{ops } 0 : \text{Elem} ; + : \text{Elem} \times \text{Elem} \rightarrow \text{Elem} ; \text{inv} : \text{Elem} \rightarrow \text{Elem} \]

\[ \text{sort } \text{Elem} \]

\[ \text{hide } \text{inv} \]

\[ (x) \land \text{inv} + (y) = (y + \text{inv} + x) \]

\[ \text{hide along } O \text{, where } O \text{ is a signature morphism} \]

\[ \text{semantics: the } O\text{-reducts of all } O\text{-models} \]

\[ \text{forall } x,y,z \text{, } x = x + 0 \]

\[ \text{hide } O \text{, where } O \text{ is a signature morphism} \]

\[ \text{is kept} \]

\[ \text{semantic effect of sentences (also those involving these symbols)} \]

\[ \text{i.e. some logical or non-logical symbols are hidden, but the} \]

\[ \text{semantic: the } O\text{-reducts of all } O\text{-models} \]
### Approximations

- $O$ approximate in $\mathcal{L}$ with $m$
- approximation of $O$ in a sublogic $\mathcal{L}$ using method $m$
- sentences not expressible in $\mathcal{L}$ are weakened or removed, as specified by $m$.
- Requirement: $O \models O$ approximate in $\mathcal{L}$ with $m$
- Not necessarily maximal with these properties (indeed, maximal such ontologies not always exist).

DOLCE_Mereology approximate in OWL with luettich
Module Extractions

- $O$ extract $c \Sigma$ with $m$
- $\Sigma$: restriction signature (subsignature of that of $O$)
- $c$: one of $\%mcons$ and $\%ccons$
- $m$: module extraction method

$O$ must be a conservative extension of the resulting extracted module.

co-ode: Pizza extract $\%mcons$
  Class: VegetarianPizza
  Class: VegetableTopping
  ObjectProperty: hasTopping
  with locality
Minimizations

- minimize \{ O \}
- forces minimal interpretation of non-logical symbols in \( O \)

**Class**: Block
- **Individual**: B1 **Types**: Block
- **Individual**: B2 **Types**: Block **DifferentFrom**: B1

then minimize \{ \\
  **Class**: Abnormal \\
  **Individual**: B1 **Types**: Abnormal \}
then

**Class**: Ontable
**Class**: BlockNotAbnormal **EquivalentTo**: Block **and** not Abnormal **SubClassOf**: Ontable
then %implied
- **Individual**: B2 **Types**: Ontable
Ontology definitions

- **ontology** $Id = O$ **end**

- assigns name (resp. IRI) $Id$ to ontology $O$, for later reference.

**ontology** co-code:Pizza =
  Class: VegetarianPizza
  Class: VegetableTopping
  ObjectProperty: hasTopping
  ...  
end
Interpretations

- **interpretation** \( id : O_1 \text{ to } O_2 = \sigma \)
- \( \sigma \) is a signature morphism or a logic translation
- expresses that \( O_2 \models \sigma(O_1) \)

**interpretation** \( i : \text{TotalOrder to Nat} = \text{Elem} \mapsto \text{Nat} \)

**interpretation** \( \text{geometry_of_time} \ %mcons : \)
\[
\text{int:owlttime_le} \\
\text{... that begin and end with an instant as lines} \\
\text{... that are incident with linearly ...} \\
\text{to } \{ \text{ord:linear_ordering and bi:complete_graphical} \\
\text{... ordered points in a special geometry, ...} \\
\text{and int:mappings/owlttime_interval_reduction} \} \\
= \text{ProperInterval} \mapsto \text{Interval} \text{ end} \]
Equivalences

- **equivalence** \( \text{Id} : O_1 \leftrightarrow O_2 = O_3 \)
- expresses that \( O_1 \) and \( O_2 \) have model classes that are in bijective correspondence
- (fragment) ontology \( O_3 \) is such that \( O_i \) then \( O_3 \) is a definitional extension of \( O_i \) for \( i = 1, 2 \).

**equivalence** \( e : \text{algebra:BooleanAlgebra} \)

\[ \text{to algebra:BooleanRing} = \]

<define missing symbols>

end
Module Relations

- **module id c : O₁ of O₂ for Σ**
- O₁ is a module of O₂ with restriction signature Σ and conservativity c
  
  c=%mcons every Σ-reduct of an O₁-model can be expanded to an O₂-model

  c=%ccons every Σ-sentence ϕ following from O₁ already follows from O₁

This relation shall hold for any module O₁ extracted from O₂ using the extract construct.
Alignments

- **alignment** \( Id \) \( card_1 \) \( card_2 : O_1 \text{ to } O_2 = c_1, \ldots, c_n \)
- \( card_i \) is (optionally) one of 1, ?, +, *
- the \( c_i \) are correspondences of form \( sym_1 \) \( rel \) \( conf \) \( sym_2 \)
  - \( sym_i \) is a symbol from \( O_i \)
  - \( rel \) is one of \( >, <, =, \%, \ni, \in, \rightarrow \), or an \( Id \)
  - \( conf \) is an (optional) confidence value between 0 and 1

Syntax of alignments follows the **alignment API**
http://alignapi.gforge.inria.fr

**alignment** Alignment1 : { **Class**: Woman } to { **Class**: Person } = Woman < Person

end
Alignment: Another Example

ontology Onto1 =
  Class: Person
  Class: Woman SubClassOf: Person
  Class: Bank
end

ontology Onto2 =
  Class: HumanBeing
  Class: Woman SubClassOf: HumanBeing
  Class: Bank
end

alignment VAlignment : Onto1 to Onto2 =
  Person = HumanBeing,
  Woman = Woman
end
Combinations

- **combine** $O_1, \ldots, O_n, L_1, \ldots, L_m$
- $L_j$ are links (interpretations, alignments) between ontologies
- The individual ontologies can be prefixed with labels, like $n : O$
- semantics is a colimit

```ontologym

**ontology** AlignedOntology1 = 
**combine** Alignment1

**ontology** VAleignedOntology = 
**combine** 1 : Onto1, 2 : Onto2, VAlignment
  % 1:Person is identified with 2:HumanBeing
  % 1:Woman is identified with 2:Woman
  % 1:Bank and 2:Bank are kept distinct

**ontology** VAleignedOntologyRenamed = 
VAleignedOntology with 1:Bank |-> RiverBank,
  2:Bank |-> FinancialBank, Person_HumanBeing |-> Person
```
Diagram for First Alignment

\{Woman\} \rightarrow Woman \sqsubseteq Person \rightarrow \{Person\}

\{Woman\} \rightarrow \{Woman\} \rightarrow \{Person\} \rightarrow \{Person\}
Colimit for First Alignment
Diagram for Second Alignment

Onto1 \{ Woman, Person\_HumanBeing \} Onto2
Colimit for Second Alignment

\{W\, o\, m\, a\, n, \, P\, e\, r\, s\, o\, n\_H\, u\, m\, a\, n\_B\, e\, i\, n\, g, \, 1 \, : \, B\, a\, n\, k, \, 2 \, : \, B\, a\, n\, k\}\n
Onto1 \rightarrow Onto2

\{W\, o\, m\, a\, n, \, P\, e\, r\, s\, o\, n\_H\, u\, m\, a\, n\_B\, e\, i\, n\, g\}\n
Construction of Diagrams

*O₁* contains, for each *s₁ = s₂* in *A*, a symbol *s₁_s₂*

*O₁' and *O₂'* contain the symbols of *O₁* and *O₂*, respectively, which appear in *A* in a correspondence *s₁ s₂* such that is not equivalence and *B* is an ontology constructed

the signature morphisms *σ₁* and *σ₂* map each symbol *s₁_s₂* to *s₁* and respectively *s₂*. 

Mossakowski
Distributed Ontology, Modeling and Specification Language (DOL)
Ontology Bridges

- $O_1$ bridge with translation $t$ $O_2$
- $t$ is a logic translation
- semantics: $O_1$ with translation $t$ then $O_2$
- $t$ will e.g. translate OWL to some DDL or $\mathcal{E}$-connections
- $O_2$: axioms involving the relations (introduced by $t$) between ontologies in $O_1$. 
Ontology Bridge Example

```plaintext
ontology Publications1 =
  Class: Publication
  Class: Article SubClassOf: Publication
  Class: InBook SubClassOf: Publication
  Class: Thesis SubClassOf: Publication
...
```

```plaintext
ontology Publications2 =
  Class: Thing
  Class: Article SubClassOf: Thing
  Class: BookArticle SubClassOf: Thing
  Class: Publication SubClassOf: Thing
  Class: Thesis SubClassOf: Thing
```
Ontology Bridge Example, cont’d

```
onterology Publications_Combined =
  combine
    1 : Publications1 with translation OWL2MS-OWL,
    2 : Publications2 with translation OWL2MS-OWL
  \%\% implicitly: Article \mapsto 1:Article ...
  \%\% Article \mapsto 2:Article ...
bridge with translation MS-OWL2DDL
  \%\% implicitly added my translation MS-OWL2DDL: binary
  1:Publication \iff 2:Publication
  1:PhdThesis \iff 2:Thesis
  1:InBook \iff 2:BookArticle
  1:Article \iff 2:Article
  1:Article \mapsto 2:Article
```
Qualifications

Qualifications choose the logic, ontology language and/or serialization:

- language
- logic
- serialization

This affects the subsequent definitions in the distributed ontology.
What is the Needed Abstract Infrastructure?

- models, sentences, satisfaction $\Rightarrow$ satisfaction systems
- reducts, (conservative) extensions $\Rightarrow$ institutes
- combinations via colimits $\Rightarrow$ institutions
- ...

Motivation  OntoIOp  DOL  Modular and Heterogeneous Ontologies  Distributed ontologies  Semantics  Conclusion
Satisfaction Systems

A satisfaction system $S = (\text{Sen}, \mathcal{M}, \models)$ consists of

- a class $\text{Sen}$ of sentences;
- a class $\mathcal{M}$ of models;
- a satisfaction relation $\models \subseteq \mathcal{M} \times \text{Sen}$. 

Institutes

An institute \( I = (\text{Sig}, \leq, \text{Sen}, \text{Mod}, \models) \) consists of

- a class \( \text{Sen} \) of sentences;
- a partially ordered class \( (\text{Sig}, \leq) \) of signatures;
- a function \( \text{sig} : \text{Sen} \to \text{Sig} \), giving the (minimal) signature of a sentence (then for each signature \( \Sigma \), let \( \text{Sen}(\Sigma) = \{ \phi \in \text{Sen} \mid \text{sig}(\phi) \leq \Sigma \} \));
- for each signature \( \Sigma \), a partially ordered class \( \text{Mod}(\Sigma) \) of \( \Sigma \)-models;
- for each signature \( \Sigma \), a satisfaction relation \( \models_{\Sigma} \subseteq \text{Mod}(\Sigma) \times \text{Sen}(\Sigma) \);
- for any \( \Sigma_2 \)-model \( M \), a \( \Sigma_1 \)-model \( M|_{\Sigma_1} \) (called the reduct), provided that \( \Sigma_1 \leq \Sigma_2 \).
Institutes (cont’d)

...such that the following properties hold:

1. given $\Sigma_1 \leq \Sigma_2$, for any $\Sigma_2$-model $M$ and any $\Sigma_1$-sentence $\varphi$

   $$M \models \varphi \text{ iff } M|_{\Sigma_1} \models \varphi$$

   (satisfaction is invariant under reduct),

2. for any $\Sigma$-model, $M|_{\Sigma} = M$, and given $\Sigma_1 \leq \Sigma_2 \leq \Sigma$,

   $$(M|_{\Sigma_2})|_{\Sigma_1} = M|_{\Sigma_1}$$

   (reducts are compositional), and

3. for any signatures $\Sigma' \leq \Sigma$, and $\Sigma$-models $M_1 \leq M_2$, we have $M_1|_{\Sigma'} \leq M_2|_{\Sigma'}$ (reducts preserve the model ordering).
Institutions

An institution $\mathcal{I}$ consists of:

- a category $\text{Sign}_\mathcal{I}$ of signatures;
- a functor $\text{Sen}_\mathcal{I} : \text{Sign}_\mathcal{I} \to \text{Set}$, giving a set $\text{Sen}(\Sigma)$ of $\Sigma$-sentences for each signature $\Sigma \in |\text{Sign}_\mathcal{I}|$, and a function $\text{Sen}(\sigma) : \text{Sen}(\Sigma) \to \text{Sen}(\Sigma')$ that yields $\sigma$-translation of $\Sigma$-sentences to $\Sigma'$-sentences for each $\sigma : \Sigma \to \Sigma'$;
- a functor $\text{Mod}_\mathcal{I} : \text{Sign}_\mathcal{I}^{\text{op}} \to \text{Set}$, giving a set $\text{Mod}(\Sigma)$ of $\Sigma$-models for each signature $\Sigma \in |\text{Sign}_\mathcal{I}|$, and a functor $\text{Mod}(\sigma) : \text{Mod}(\Sigma') \to \text{Mod}(\Sigma)$; for each $\sigma : \Sigma \to \Sigma'$;
- for each $\Sigma \in |\text{Sign}_\mathcal{I}|$, a satisfaction relation $\models_{\mathcal{I},\Sigma} \subseteq \text{Mod}_\mathcal{I}(\Sigma) \times \text{Sen}_\mathcal{I}(\Sigma)$ such that for any signature morphism $\sigma : \Sigma \to \Sigma'$, $\Sigma$-sentence $\varphi \in \text{Sen}_\mathcal{I}(\Sigma)$ and $\Sigma'$-model $M' \in \text{Mod}_\mathcal{I}(\Sigma')$:
  $$M' \models_{\mathcal{I},\Sigma} \sigma(\varphi) \iff M'|_\sigma \models_{\mathcal{I},\Sigma} \varphi$$

[Satisfaction condition]
Institutions

Signatures

\[ \Sigma \sigma \rightarrow \Sigma' \]

Sentences

\[ \text{Sen } \Sigma \rightarrow \text{Sen } \Sigma' \]

Satisfaction

\[ \models_{\Sigma} \rightarrow \models_{\Sigma'} \]

Models

\[ \text{Mod } \Sigma \leftarrow \text{Mod } \Sigma' \]
Institution Comorphisms (Translations)

Institution comorphisms

Signatures

\[ \Sigma \xrightarrow{\Phi} \Phi \Sigma \]

Sentences

\[ \text{Sen}^I \Sigma \xrightarrow{\alpha_\Sigma} \text{Sen}^J \Phi \Sigma \]

Satisfaction

\[ \models^I \Sigma \xrightarrow{\beta_\Sigma} \models^J \Phi \Sigma \]

Models

\[ \text{Mod}^I \Sigma \xleftarrow{\beta_\Sigma} \text{Mod}^J \Phi \Sigma \]
Institution Morphisms (Projections)

Institution morphisms

Signatures

\[ \Sigma \xrightarrow{\Phi} \Phi \Sigma \]

Sentences

\[ \text{Sen}^I \Sigma \xrightarrow{\alpha_\Sigma} \text{Sen}^J \Phi \Sigma \]

Satisfaction

\[ \models^I_\Sigma \quad \models^J_{\Phi \Sigma} \]

Models

\[ \text{Mod}^I \Sigma \xrightarrow{\beta_\Sigma} \text{Mod}^J \Phi \Sigma \]
Formal Semantics of DOL Ontologies

- $\Sigma, M, L$
- $L$ is a logic (institute, institution) in the logic graph
- $\Sigma$ is a signature in $L$
- $M$ is a class of $\Sigma$-models in $L$
Challenges

- What is a suitable abstract meta framework for non-monotonic logics and rule languages like RIF and RuleML? Are institutions suitable here? Are the modularity questions for these languages different from those for OWL?
- What is a useful abstract notion of query (language) and answer substitution?
- How to integrate TBox-like and ABox-like ontologies?
- Can the notions of class hierarchy and of satisfiability of a class be generalised from OWL to other languages?
- How to interpret alignment correspondences with confidence other that 1 in a combination?
- Can logical frameworks be used for the specification of ontology languages and translations?
Tool support: Heterogeneous Tool Set (Hets)

- available at hets.dfki.de
- speaks DOL, OWL, Common Logic, and other languages
- analysis
- management of proof obligations
- theorem proving, model finding
Ontohub is a web-based repository engine for distributed heterogeneous (multi-language) ontologies

- prototype available at ontohub.org
- speaks DOL, OWL, Common Logic, and other languages
- mid-term goal: follow the Open Ontology Repository Initiative (OOR) architecture and API
- API is discussed at https://github.com/ontohub/OOR_Ontohub_API
- annual Ontology summit as a venue for review, and discussion
Conclusion

- DOL covers many aspects of modularity that have been discussed in the different WoMos
- DOL is a meta language, focusing on relations between ontologies ("ontology-in-the-large")
- you can help with joining the OntoIOp discussion
  - see ontiop.org