The Distributed Ontology, Modeling and Specification Language (DOL)
Recent developments

Till Mossakowski[1] Oliver Kutz[1]
Christoph Lange[2] Mihai Codescu[1]

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
INF FAKULTÄT FÜR INFORMATIK

[1]University of Magdeburg
[2]University of Bonn

IFIP WG 1.3 meeting, 2014-09-02

Motivation

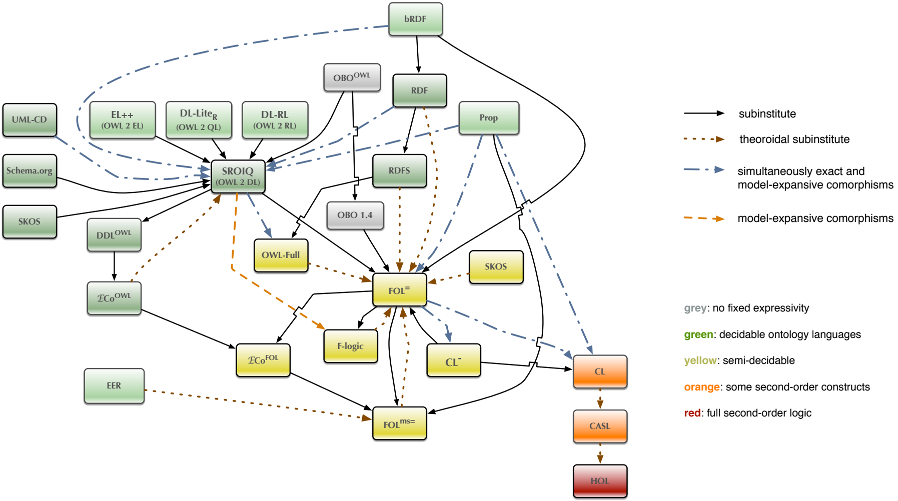
The Big Picture of Interoperability

Modeling	**Specification**	**Knowledge engineering**
Objects/data	Software	Concepts/data
Models	Specifications	Ontologies
Metamodels	Specification languages	Ontology languages

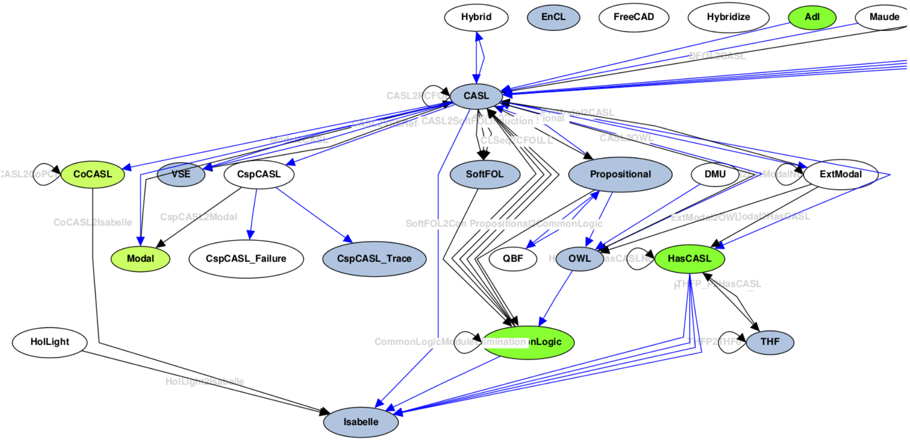
Diversity and the need for interoperability occur at all these levels!

(Formal) ontologies, (formal) models and (formal) specifications will henceforth be abbreviated as OMS.

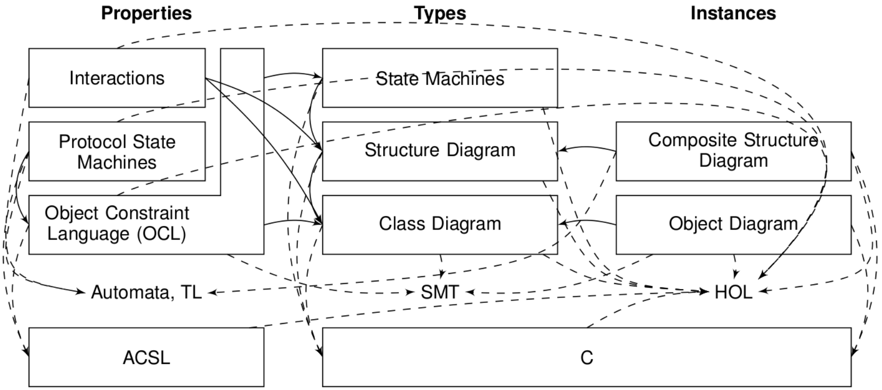
Ontologies: An Initial Logic Graph



Specifications: An Initial Logic Graph



UML models: An Initial Logic Graph



Motivation: Diversity of Operations on and Relations among OMS

Various operations and relations on OMS are in use:

- structuring: union, translation, hiding, ...
- refinement
- matching and alignment
  - of many OMS covering one domain
- module extraction
  - get relevant information out of large OMS
- approximation
  - model in an expressive language, reason fast in a lightweight one
- ontology-based database access/data management
- distributed OMS
  - bridges between different modellings

OntoIOp

Need for a Unifying Meta Language

Not yet another OMS language, but a meta language covering

- diversity of OMS languages
- translations between these
- diversity of operations on and relations among OMS

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

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

The OntoIOp initiative (ontoiop.org)

- 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) has been issued in December 2013
  - proposals answering RFP due in December 2014
- 50 experts participate, $\sim$ 15 have contributed
- OntoIOp is open for your ideas, so join us!
- Distributed Ontology, Modeling and Specification Language
  - DOL = one specific answer to the RFP requirements
  - there may be other answers to the RFP
  - DOL is based on some graph of institutions and (co)morphisms
  - DOL has a model-level and a theory-level semantics

DOL

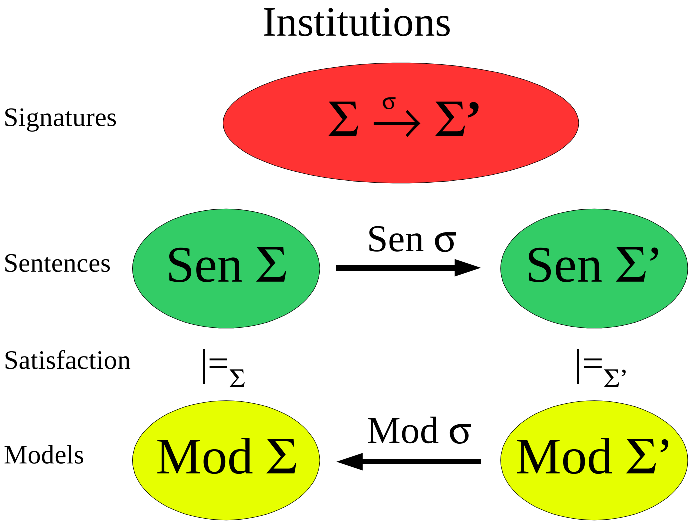
Overview of DOL

1. modular and heterogeneous OMS
   - basic OMS (flattenable)
   - references to named OMS
   - extensions, unions, translations (flattenable)
   - reductions (elusive)
   - approximations, module extractions (flattenable)
   - minimization, maximization (elusive)
   - combination, OMS bridges (flattenable)

   only OMS with flattenable components are flattenable

   flattenable = can be flattened to a basic OMS
2. OMS declarations and relations (based on 1)
   - OMS definitions (giving a name to an OMS)
   - interpretations (of theories), equivalences
   - module relations
   - alignments

Institutions (intuition)



Institutions (formal definition)

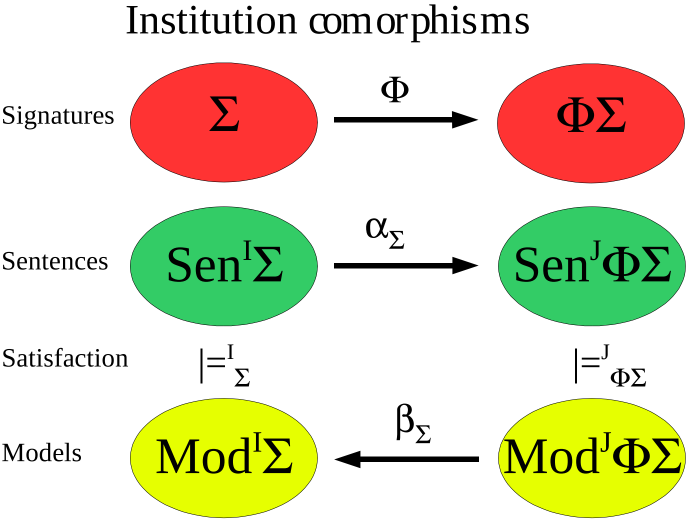
An institution $\mathcal{I} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ consists of:

- a category **Sign** of signatures;
- a functor $\mathbf{Sen} \colon \mathbf{Sign} \to \mathbf{Set}$,giving a set $\mathbf{Sen}(\Sigma)$ of Σ-sentences for each signature $\Sigma \in |\mathbf{Sign}|$, and a function $\mathbf{Sen}(\sigma) \colon \mathbf{Sen}(\Sigma) \to \mathbf{Sen}(\Sigma')$ that yields σ-translation of Σ-sentences to Σ'-sentences for each $\sigma \colon \Sigma \to \Sigma'$;
- a functor $\mathbf{Mod} \colon \mathbf{Sign}^{op} \to \mathbf{Set}$,giving a set $\mathbf{Mod}(\Sigma)$ of Σ-models for each signature $\Sigma \in |\mathbf{Sign}|$, and a functor $\_|_\sigma = \mathbf{Mod}(\sigma) \colon \mathbf{Mod}(\Sigma') \to \mathbf{Mod}(\Sigma)$; for each $\sigma \colon \Sigma \to \Sigma'$;
- for each $\Sigma \in |\mathbf{Sign}|$, a satisfaction relation $\models_{\mathcal{I},\Sigma} \; \subseteq \mathbf{Mod}(\Sigma) \times \mathbf{Sen}(\Sigma)$

such that for any signature morphism $\sigma \colon \Sigma \to \Sigma'$, Σ-sentence $\varphi \in \mathbf{Sen}(\Sigma)$ and Σ'-model $M' \in \mathbf{Mod}(\Sigma')$:

$$M' \models_{\mathcal{I},\Sigma'} \sigma(\varphi) \iff M'|_\sigma \models_{\mathcal{I},\Sigma} \varphi \qquad \text{[Satisfaction condition]}$$

Institution comorphisms (embeddings, encodings)



Institution comorphisms (embeddings, encodings)

Definition

Let $\mathcal{I} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ and $\mathcal{I}' = \langle \mathbf{Sign}', \mathbf{Sen}', \mathbf{Mod}', \langle \models'_{\Sigma'} \rangle_{\Sigma' \in |\mathbf{Sign}'|} \rangle$ be institutions. An *institution comorphism* $\rho \colon \mathcal{I} \to \mathcal{I}'$ consists of:

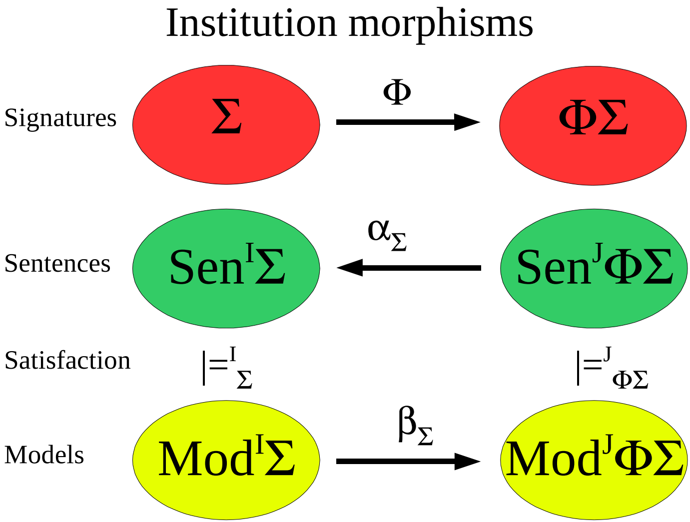
- a functor $\rho^{Sign} \colon \mathbf{Sign} \to \mathbf{Sign}'$;
- a natural transformation $\rho^{Sen} \colon \mathbf{Sen} \to \rho^{Sign} ; \mathbf{Sen}'$, and
- a natural transformation $\rho^{Mod} \colon (\rho^{Sign})^{op} ; \mathbf{Mod}' \to \mathbf{Mod}$,

such that for any $\Sigma \in |\mathbf{Sign}|$, any $\varphi \in \mathbf{Sen}(\Sigma)$ and any $M' \in \mathbf{Mod}'(\rho^{Sign}(\Sigma))$:

$$M' \models'_{\rho^{Sign}(\Sigma)} \rho^{Sen}_\Sigma(\varphi) \iff \rho^{Mod}_\Sigma(M') \models_\Sigma \varphi$$

$$[Satisfaction\ condition]$$

Institution morphisms (projections)



Institution morphisms (projections)

Definition

Let $\mathcal{I} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ and $\mathcal{I}' = \langle \mathbf{Sign}', \mathbf{Sen}', \mathbf{Mod}', \langle \models'_{\Sigma'} \rangle_{\Sigma' \in |\mathbf{Sign}'|} \rangle$ be institutions. An *institution morphism* $\mu \colon \mathcal{I} \to \mathcal{I}'$ consists of:

- a functor $\mu^{Sign} \colon \mathbf{Sign} \to \mathbf{Sign}'$;
- a natural transformation $\mu^{Sen} \colon \mu^{Sign} ; \mathbf{Sen}' \to \mathbf{Sen}$, and
- a natural transformation $\mu^{Mod} \colon \mathbf{Mod} \to (\mu^{Sign})^{op} ; \mathbf{Mod}'$,

such that for any signature $\Sigma \in |\mathbf{Sign}|$, any $\varphi' \in \mathbf{Sen}'(\mu^{Sign}(\Sigma))$ and any $M \in \mathbf{Mod}(\Sigma)$:

$$M \models_\Sigma \mu^{Sen}_\Sigma(\varphi') \iff \mu^{Mod}_\Sigma(M) \models'_{\mu^{Sign}(\Sigma)} \varphi'$$

$[Satisfaction\ condition]$

Unions, differences and inclusive institutions

We assume that for each institution, there exists (possibly partial) union and difference operations on signatures. E.g. an inclusion system on signatures would be a good framework where this can be required.

Definition (adopted from Goguen, Roșu)

An *weakly inclusive category* is a category having a broad subcategory which is a partially ordered class.
An *wekaly inclusive institution* is one with an inclusive signature category such that the sentence functor preserves inclusions.

We also assume that model categories are weakly inclusive.

$M|_\Sigma$ means $M|_\iota$ where $\iota : \Sigma \rightarrow Sig(M)$ is the inclusion.

Semantic domains of DOL

- semantics of a flattenable OMS has form (I, Σ, Ψ) (theory-level)
- semantics of an elusive OMS has form $(I, \Sigma, \mathcal{M})$ (model-level)
  - institution I
  - signature Σ in I
  - set Ψ of Σ-sentences
  - class $\mathcal{M}$ of Σ-models

  We can obtain the model-level semantics from the theory-level semantics by taking $\mathcal{M} = \{M \in \mathbf{Mod}(\Sigma) \mid M \models \Psi\}$.
- semantics of a OMS declaration/relation has form $\Gamma\colon IRI \longrightarrow (OMS \uplus OMS \times OMS \times SigMor)$
  - OMS is the class of all triples (I, Σ, Ψ), $(I, \Sigma, \mathcal{M})$
  - for interpretations etc., domain, codomain and signature morphism is recorded: $OMS \times OMS \times SigMor$

Modular and Heterogeneous OMS

Basic OMS

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

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

- e.g. in Common Logic:

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

Semantics of basic OMS

We assume that $[\![O]\!]_{basic} = (I, \Sigma, \Psi)$ for some OMS language based on I. The semantics consists of

- the institution I
- a signature Σ in I
- a set Ψ of Σ-sentences

This direct leads to a theory-level semantics for the OMS:

$$[\![O]\!]^T_\Gamma = [\![O]\!]_{basic}$$

Generally, if a theory-level semantics is given: $[\![O]\!]^T_\Gamma = (I, \Sigma, \Psi)$, this leads to a model-level semantics as well:

$$[\![O]\!]^M_\Gamma = (I, \Sigma, \{M \in Mod(\Sigma) \mid M \models \Psi\})$$

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 φ 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 %implies** 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
```

Semantics of extensions

O_1 flattenable $[\![O_1 \textbf{ then } O_2]\!]^T_\Gamma = (I, \Sigma_1 \cup \Sigma_2, \Psi_1 \cup \Psi_2)$

where

- $[\![O_1]\!]^T_\Gamma = (I, \Sigma_1, \Psi_1)$
- $[\![O_2]\!]_{basic} = (I, \Sigma_2, \Psi_2)$

O_1 elusive $[\![O_1 \textbf{ then } O_2]\!]^M_\Gamma = (I, \Sigma_1 \cup \Sigma_2, \mathcal{M}')$

where

- $[\![O_1]\!]^M_\Gamma = (I, \Sigma_1, \mathcal{M}_1)$
- $[\![O_2]\!]_{basic} = (I, \Sigma_2, \Psi_2)$
- $\mathcal{M}' = \{M \in \mathbf{Mod}(\Sigma_1 \cup \Sigma_2) \mid M \models \Psi_2, M|_{\Sigma_1} \in \mathcal{M}_1\}$

Semantics of extensions (cont'd)

%mcons (%def, %mono) leads to the additional requirement that

> *each model in $\mathcal{M}_1$ has a (unique, unique up to isomorphism) $\Sigma_1 \cup \Sigma_2$-expansion to a model in $\mathcal{M}'$.*

%implies leads to the additional requirements that

> $\Sigma_2 \subseteq \Sigma_1$ *and* $\mathcal{M}' = \mathcal{M}_1$.

%ccons leads to the additional requirement that

> $\mathcal{M}' \models \varphi$ *implies* $\mathcal{M}_1 \models \varphi$ *for any* Σ_1*-sentence* φ.

Theorem

%mcons implies %ccons, but not vice versa.

References to Named OMS

- Reference to an OMS 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

co-ode:pizza.owl
```

Semantics Reference to Named OMS: $[\![iri]\!]_\Gamma = \Gamma(iri)$

Unions

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

```
algebra:Monoid and algebra:Commutative
```

Semantics of unions

O_1, O_2 flattenable $[\![O_1 \textbf{ and } O_2]\!]^T_\Gamma = (I, \Sigma_1 \cup \Sigma_2, \Psi_1 \cup \Psi_2)$, where

- $[\![O_i]\!]^T_\Gamma = (I, \Sigma_i, \Psi_i)$ $(i = 1, 2)$

one of O_1, O_2 elusive $[\![O_1 \textbf{ and } O_2]\!]^M_\Gamma = (I, \Sigma_1 \cup \Sigma_2, \mathcal{M})$, where

- $[\![O_1]\!]^M_\Gamma = (I, \Sigma_i, \mathcal{M}_i)$ $(i = 1, 2)$
- $\mathcal{M} = \{M \in \mathbf{Mod}(\Sigma) \mid M|_{\Sigma_i} \in \mathcal{M}_i, i = 1, 2\}$

Translations

- **O with σ**, where σ is a signature morphism
- **O with translation ρ**, where ρ is an institution comorphism

```
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
```

Semantics of translations

O flattenable Let $[\![O]\!]^T_\Gamma = (I, \Sigma, \Psi)$

- homogeneous translation
  $[\![O \textbf{ with } \sigma : \Sigma \to \Sigma']\!]^T_\Gamma = (I, \Sigma', \sigma(\Psi))$
- heterogeneous translation
  $[\![O \textbf{ with translation } \rho : I \to I']\!]^T_\Gamma = (I', \rho^{Sig}(\Sigma), \rho^{Sen}(\Psi))$

O elusive Let $[\![O]\!]^M_\Gamma = (I, \Sigma, \mathcal{M})$

- homogeneous translation
  $[\![O \textbf{ with } \sigma : \Sigma \to \Sigma']\!]^M_\Gamma = (I, \Sigma', \mathcal{M}')$
  where $\mathcal{M}' = \{M \in \textbf{Mod}(\Sigma') \mid M|_\sigma \in \mathcal{M}\}$
- heterogeneous translation
  $[\![O \textbf{ with translation } \rho : I \to I']\!]^M_\Gamma = (I', \rho^{Sig}(\Sigma), \mathcal{M}')$ where
  $\mathcal{M}' = \{M \in \textbf{Mod}^{I'}(\rho^{Sig}(\Sigma)) \mid \rho^{Mod}(M) \in \mathcal{M}\}$

Hide – Extract – Forget – Filter

	hide/reveal	remove/extract	forget/keep	filter
semantic background	model reduct	conservative extension	uniform interpolation	theory difference
relation to original	interpretable	subtheory	interpretable	subtheory
approach	model level	theory level	theory level	theory level
type of OMS	elusive	flattenable	flattenable	flattenable
signature of result	$= \Sigma$	$\geq \Sigma$	$= \Sigma$	$= \Sigma$
change of logic	possible	not possible	possible	not possible
application	specification	ontologies	ontologies	blending

Reduction: Hide/reveal

- intuition: some logical or non-logical symbols are hidden, but the semantic effect of sentences (also those involving these symbols) is kept
- O **reveal** Σ, where Σ is a subsignature of that of O
- O **hide** Σ, where Σ is a subsignature of that of O
- O **hide along** μ, where μ is an institution morphism

Reduction: example

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem; inv:Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
                  . x+inv(x)=0
hide inv
```

Semantics: class of all monoids that can be extended with an inverse, i.e. class of all groups. The effect is second-order quantification:

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem;
exists inv:Elem->Elem .
  forall x,y,z:elem . 0+x=x
                    /\ x+(y+z) = (x+y)+z
                    /\ x+inv(x)=0
```

Semantics of reductions

Let $[\![O]\!]^M_\Gamma = (I, \Sigma, \mathcal{M})$

- homogeneous reduction
  $[\![O \textbf{ reveal } \Sigma']\!]^M_\Gamma = (I, \Sigma', \mathcal{M}|_{\Sigma'})$
  $[\![O \textbf{ hide } \Sigma']\!]^M_\Gamma = [\![O \textbf{ reveal } \Sigma \setminus \Sigma']\!]^M_\Gamma$
- heterogeneous reduction
  $[\![O \textbf{ hide along } \rho : I \to I']\!]^M_\Gamma = (I', \rho^{Sig}(\Sigma), \rho^{Mod}(\mathcal{M}))$

$\mathcal{M}|_{\Sigma'}$ may be impossible to capture by a theory (even if $\mathcal{M}$ is).
The proof calculus for refinements involving reduction needs invention of some OMS O'':

$$\frac{O \rightsquigarrow O''}{O \textbf{ hide } \Sigma \rightsquigarrow O'} \quad \text{if } \iota\colon O' \longrightarrow O'' \text{ is a conservative extension}$$

where $\iota : \Sigma \to Sig(O)$ is the inclusion

Module Extraction: remove/extract

O extract Σ

- Σ: restriction signature (subsignature of that of O)
- O must be a conservative extension of the resulting extracted module. (If not, the module is suitably enlarged.)
- Dually: **O remove Σ**
- Note: The extraction methods from the literature all guarantee model-theoretic conservativity.

Module Extraction: example

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem; inv:Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
                  . x+inv(x) = 0
remove inv
```

The semantics is the following theory:

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem; inv:Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
                  . x+inv(x) = 0
```

The module needs to be enlarged to the whole OMS.

Module Extraction: 2nd example

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem; inv:Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
                  . x+inv(x) = 0
                  . exists y:Elem . x+y=0
remove inv
```

The semantics is the following theory:

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
                  . exists y:Elem . x+y=0
```

Here, adding `inv` is conservative.

Modules

Definition

$O' \subseteq O$ is a Σ-module of (flat) O iff O is a model-theoretic Σ-conservative extension of O', i.e. for every model M of O', $M|_\Sigma$ can be expanded to an O-model.

Depleting modules

Definition

Let O_1 and O_2 be two OMS and $\Sigma \subseteq Sig(O_i)$.
Then O_1 and O_2 are Σ-inseparable ($O_1 \equiv_\Sigma O_2$) iff

$$Mod(O_1)|_\Sigma = Mod(O_2)|_\Sigma$$

Definition

$O' \subseteq O$ is a depleting Σ-module of (flat) O iff $O \setminus O' \equiv_{\Sigma \cup Sig(O')} \emptyset$.

Theorem

1. *Depleting Σ-modules are Σ-conservative.*
2. *The minimum depleting Σ-module always exists.*

Semantics of module extraction (remove/extract)

Note: O must be flattenable!

Let $[\![O]\!]^T_\Gamma = (I, \Sigma, \Psi)$.
$[\![O \textbf{ extract } \Sigma_1]\!]^T_\Gamma = (I, \Sigma_2, \Psi_2)$
where $(\Sigma_2, \Psi_2) \subseteq (\Sigma, \Psi)$ is the minimum depleting Σ_1-module of (Σ, Ψ)

$[\![O \textbf{ remove } \Sigma_1]\!]^T_\Gamma = [\![O \textbf{ extract } \Sigma \setminus \Sigma_1]\!]^T_\Gamma$

Tools can extract any module (i.e. using locality). Any two modules will have the same Σ-consequences.

Interpolation: forget/keep

- O **keep in** Σ, where Σ is a subsignature of that of O
- O **keep in** Σ **with** I, where Σ is a subsignature of that of O, and I is a subinstitution of that of O
    - intuition: theory of O is interpolated in smaller signature/logic
- dually
    - O **forget** Σ
    - O **forget** Σ **with** I

Interpolation: example

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem; inv:Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
                  . x+inv(x) = 0
forget inv
```

The semantics is the following theory:

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
                  . exists y:Elem . x+y=0
```

Computing interpolants can be hard, even undecidable.

Semantics of interpolation (forget/keep)

Note: O must be flattenable!
Let $[\![O]\!]_\Gamma^T = (I, \Sigma, \Psi)$.

- homogeneous interpolation
  $[\![O \textbf{ keep in } \Sigma']\!]_\Gamma^T = (I, \Sigma', \{\varphi \in \textbf{Sen}(\Sigma') \mid \Psi \models \varphi\})$
  (note: any logically equivalent theory will also do)
  $[\![O \textbf{ forget } \Sigma']\!]_\Gamma^T = [\![O \textbf{ keep in } \Sigma \setminus \Sigma']\!]_\Gamma^T$
- heterogeneous interpolation
  $[\![O \textbf{ keep in } \Sigma' \textbf{ with } I']\!]_\Gamma^T =$
  $(I', \Sigma', \{\varphi \in \textbf{Sen}^{I'}(\Sigma') \mid \Psi \models \rho^{Sen}(\varphi)\})$
  where $\rho : I' \to I$ is the inclusion
  and Σ' is such that $\rho^{Sig}(\Sigma') \subseteq \Sigma$
  $[\![O \textbf{ forget } \Sigma' \textbf{ with } I']\!]_\Gamma^T = [\![O \textbf{ keep in } \Sigma \setminus \Sigma' \textbf{ with } I']\!]_\Gamma^T$

Filtering

- O **filter** T, where T is a subtheory (fragment) of that of O
  - intuition: all axioms involving symbols in $Sig(T)$ are deleted
  - moreover, all axioms contained in T are deleted as well
- A dual notion does not make much sense
  (indeed, just T would be delivered).

Filtering: example

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem; inv:Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
                  . x+inv(x) = 0
filter inv
```

The semantics is the following theory:

```
sort Elem
ops 0:Elem; __+__:Elem*Elem->Elem
forall x,y,z:elem . 0+x=x
                  . x+(y+z) = (x+y)+z
```

Semantics of filtering

Note: O must be flattenable!

Let $[\![O]\!]_\Gamma^T = (I, \Sigma, \Psi)$.
$[\![O \text{ **filter** } (\Sigma', \Phi)]\!]_\Gamma^T = (I, \Sigma', Sen(\iota)^{-1}(\Psi) \setminus \Phi)$
where $\iota : \Sigma' \rightarrow \Sigma$ is the inclusion

Hide – Extract – Forget – Filter

	hide/reveal	remove/extract	forget/keep	filter
semantic background	model reduct	conservative extension	uniform interpolation	theory difference
relation to original	interpretable	subtheory	interpretable	subtheory
approach	model level	theory level	theory level	theory level
type of OMS	elusive	flattenable	flattenable	flattenable
signature of result	$= \Sigma$	$\geq \Sigma$	$= \Sigma$	$= \Sigma$
change of logic	possible	not possible	possible	not possible
application	specification	ontologies	ontologies	blending

Relations among the different notions

$$
\begin{array}{ll}
 & Mod(O \textbf{ hide } \Sigma) \\
= & Mod(O \textbf{ extract } \Sigma)|_{Sig(O)\setminus\Sigma} \\
\subseteq & Mod(O \textbf{ forget } \Sigma) \\
\subseteq & Mod(O \textbf{ filter } \Sigma)
\end{array}
$$

Pros and Cons

	hide/reveal	remove/extract	forget/keep	filter
information loss	none	none	minimal	large
computability	bad	good/depends	depends	easy
signature of result	$= \Sigma$	$\geq \Sigma$	$= \Sigma$	$= \Sigma$
change of logic	possible	not possible	possible	not possible
conceptual simplicity	simple (but unintuitive)	complex	farily simple	simple

Minimizations (circumscription)

- O_1 **then minimize {** O_2 **}**
- forces minimal interpretation of non-logical symbols in O_2

```
  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
```

Semantics of minimizations

Let $[\![O_1]\!]^M_\Gamma = (I, \Sigma_1, \mathcal{M}_1)$
Let $[\![O_1 \textbf{ then } O_2]\!]^M_\Gamma = (I, \Sigma_2, \mathcal{M}_2)$
Then

$$[\![O_1 \textbf{ then minimize } O_2]\!]^M_\Gamma = (I, \Sigma_2, \mathcal{M})$$

where
$\mathcal{M} = \{M \in \mathcal{M}_2 \mid M \text{ is minimal in } \{M' \in \mathcal{M}_2 \mid M'|_{\Sigma_1} = M|_{\Sigma_1}\}\}$

Dually: maximization.

Freeness

- O_1 **then free {** O_2 **}**
- forces initial interpretation of non-logical symbols in O_2

```
sort Elem
then free {
   sort Bag
   ops mt:Bag;
       __union__:Bag*Bag->Bag, assoc, comm, unit mt
         }
```

Cofreeness

- O_1 **then cofree** { O_2 }
- forces final interpretation of non-logical symbols in O_2

```
sort Elem
then cofree {
   sort Stream
   ops head:Stream->Elem;
       tail:Stream->Stream
         }
```

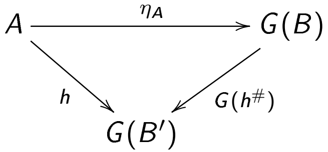
Semantics of freeness

Let $[\![O_1]\!]^M_\Gamma = (I, \Sigma_1, \mathcal{M}_1)$
Let $[\![O_1 \textbf{ then } O_2]\!]^M_\Gamma = (I, \Sigma_2, \mathcal{M}_2)$
Let $\iota : \Sigma_1 \to \Sigma_2$ be the inclusion
Then

$$[\![O_1 \textbf{ then free } O_2]\!]^M_\Gamma = (I, \Sigma_2, \mathcal{M})$$

where $\mathcal{M} = \{M \in \mathcal{M}_2 \mid M \text{ is } Mod(\iota)\text{-free over } M|_\iota \text{ with unit } id\}$

Given a functor $G\colon \mathbf{B} \longrightarrow \mathbf{A}$, an object $B \in \mathbf{B}$ is called *G-free (with unit* $\eta_A\colon A \longrightarrow G(B)$*) over* $A \in \mathbf{A}$, if for any object $B' \in \mathbf{B}$ and any morphism $h\colon A \longrightarrow G(B')$, there is a unique morphism $h^\#\colon B \longrightarrow B'$ such that $\eta_A; G(h^\#) = h$.



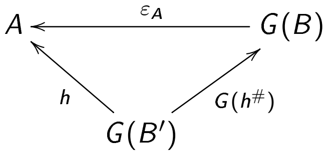
Semantics of cofreeness

Let $[\![O_1]\!]_\Gamma^M = (I, \Sigma_1, \mathcal{M}_1)$
Let $[\![O_1 \textbf{ then } O_2]\!]_\Gamma^M = (I, \Sigma_2, \mathcal{M}_2)$
Let $\iota : \Sigma_1 \to \Sigma_2$ be the inclusion
Then

$$[\![O_1 \textbf{ then cofree } O_2]\!]_\Gamma^M = (I, \Sigma_2, \mathcal{M})$$

$\mathcal{M} = \{M \in \mathcal{M}_2 \mid M \text{ is } Mod(\iota)\text{-cofree over } M|_\iota \text{ with counit } id\}$

Given a functor $G\colon \mathbf{B} \longrightarrow \mathbf{A}$, an object $B \in \mathbf{B}$ is called *G-cofree (with counit $\varepsilon_A\colon G(B) \longrightarrow A$) over $A \in \mathbf{A}$*, if for any object $B' \in \mathbf{B}$ and any morphism $h\colon G(B') \longrightarrow A$, there is a unique morphism $h^\#\colon B' \longrightarrow B$ such that $G(h^\#); \varepsilon_A = h$.



OMS declarations and relations

OMS definitions

- **OMS** *IRI* = *O* **end**
- assigns name *IRI* to OMS *O*, for later reference $\Gamma(IRI) := [\![O]\!]_\Gamma$

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

Interpretations

- **interpretation** Id : O_1 **to** $O_2 = \sigma$
- σ is a signature morphism or a logic translation
- expresses that O_2 logically implies $\sigma(O_1)$

```
interpretation i : TotalOrder to Nat = Elem ↦ Nat
interpretation geometry_of_time %mcons :
 %% Interpretation of linearly ordered time intervals..
  int:owltime_le
 %% ... that begin and end with an instant as lines
 %% that are incident with linearly ...
  to { ord:linear_ordering and bi:complete_graphical
%% ... ordered points in a special geometry, ...
       and  int:mappings/owltime_interval_reduction }
  = ProperInterval ↦ Interval end
```

Semantics of interpretations

Let $[\![O_i]\!]^M_\Gamma = (I, \Sigma_i, \mathcal{M}_i)$ $(i = 1, 2)$

$$[\![\textbf{interpretation } IRI : O_1 \textbf{ to } O_2 = \sigma]\!]^M_\Gamma$$

is defined iff

$$Mod(\sigma)(\mathcal{M}_2) \subseteq \mathcal{M}_1$$

In this case, $\Gamma(IRI) := ((I, \Sigma_1, \mathcal{M}_1), (I, \Sigma_2, \mathcal{M}_2), \sigma)$.

Graphs (diagrams)

```
graph G =
```
$G_1, \ldots, G_m, O_1, \ldots, O_n, M_1, \ldots, M_p$
excluding $G'_1, \ldots, G'_i, O'_1, \ldots, O'_j, M'_1, \ldots, M'_k$

- G_i are other graphs
- O_i are OMS (possibly prefixed with labels, like $n : O$)
- M_i are mappings (views, interpretations)

Combinations

- **combine** G
- G is a graph
- semantics is the (a) colimit of the diagram G

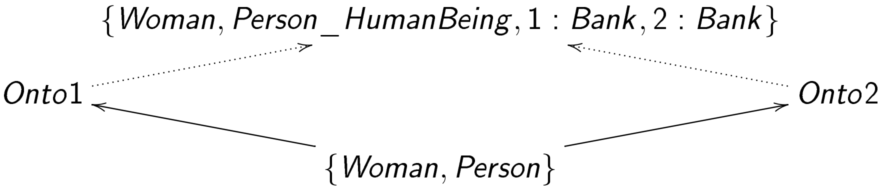
```
ontology AlignedOntology1 =
  combine G
```

There is a natural semantics of diagrams: compatible families of models.
Then in exact institutions, models of diagrams are in bijective correspondence to models of the colimit.

Sample combination

```
ontology Source =
  Class: Person
  Class: Woman SubClassOf: Person
ontology Onto1 =
  Class: Person          Class: Bank
  Class: Woman SubClassOf: Person
interpretation I1 : Source to Onto1 =
   Person |-> Person, Woman |-> Woman
ontology Onto2 =
  Class: HumanBeing      Class: Bank
  Class: Woman SubClassOf: HumanBeing
interpretation I2 : Source to Onto2 =
   Person |-> HumanBeing, Woman |-> Woman
ontology CombinedOntology =
  combine Source, Onto1, Onto2, I1, I2
```

Resulting colimit



Alignments

- **alignment** *Id* $card_1$ $card_2$: O_1 **to** O_2 $= c_1, \ldots c_n$
  assuming SingleDomain | GlobalDomain | ContextualizedDomain
- $card_i$ 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 $>$, $<$, $=$, %, $\ni$, $\in$, $\mapsto$, 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: Example

```
ontology S =  Class: Person
  Individual: alex Types: Person
  Class: Child

ontology T =  Class: HumanBeing
  Class: Male SubClassOf: HumanBeing
  Class: Employee

alignment A : S to T =
  Person = HumanBeing
  alex in Male
  Child < not Employee
  assuming GlobalDomain
```

Graphs (diagrams), revisited

graph G =
$G_1, \ldots, G_m, O_1, \ldots, O_n, M_1, \ldots, M_p, A_1, \ldots, A_r$
excluding $G'_1, \ldots, G'_i, O'_1, \ldots, O'_j, M'_1, \ldots, M'_k$

- G_i are other graphs
- O_i are OMS (possibly prefixed with labels, like $n : O$)
- M_i are mappings (views, equivalences)
- A_i are alignments

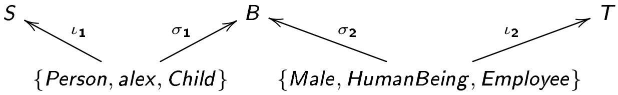
The resulting diagram G includes (institution-specific) W-alignment diagrams for each alignment A_i. Using **assuming**, assumptions about the domains of all OMS can be specified:

SingleDomain aligned symbols are mapped to each other

GlobalDomain aligned OMS a relativized

ContextualizedDomain alignments are reified as binary relations

Diagram of a SingleDomain alignment



where

ontology B =

Class: *Person_HumanBeing*
Class: *Employee*
Class: *Child*
SubClassOf: ¬ *Employee*
Individual: *alex*
Types: *Male*

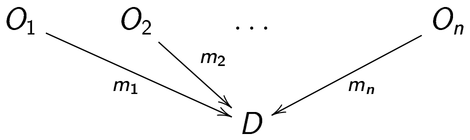
Resulting colimit

The colimit ontology of the diagram of the alignment above is:

ontology B = **Class:** *Person_HumanBeing*
Class: *Employee*
Class: *Male* **SubClassOf:** *Person_HumanBeing*
Class: *Child* **SubClassOf:** ¬ *Employee*
Individual: *alex* **Types:** *Male*, *Person_HumanBeing*

Background Simple semantics of diagrams

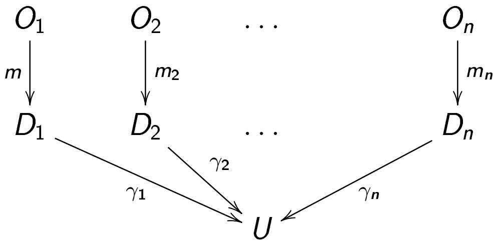
Framework: institutions like OWL, FOL, ...
Ontologies are interpreted over the same domain



- model for A: (m_1, m_2) such that $m_1(s)\ R\ m_2(t)$ for each $s\ R\ t$ in A
- model for a diagram: family (m_i) of models such that (m_i, m_j) is a model for A_{ij}
- local models of O_j modulo a diagram: jth-projection on models of the diagram

Integrated semantics of diagrams

Framework: different domains reconciled in a global domain



- model for a diagram: family (m_i) of models with equalizing function γ such that $(\gamma_i m_i, \gamma_j m_j)$ is a model for A_{ij}

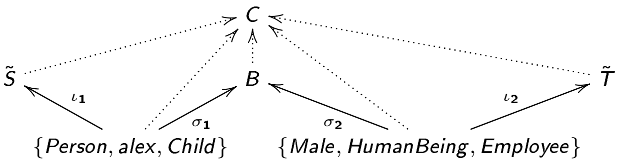
Relativization of an OWL ontology

Let O be an ontology, define its relativization $\tilde{O}$:

- concepts are concepts of O with a new concept $\top_O$;
- roles and individuals are the same
- axioms:
  - each concept C is subsumed by $\top_O$,
  - each individual i is an instance of $\top_O$,
  - each role r has domain and range $\top_O$.

  and the axioms of O where the following replacement of concept is made:
  - each occurence of $\top$ is replaced by $\top_O$,
  - each concept $\neg C$ is replaced by $\top_O \setminus C$, and
  - each concept $\forall R.C$ is replaced by $\top_O \sqcap \forall R.C$.

Example: integrated semantics



where

ontology B =

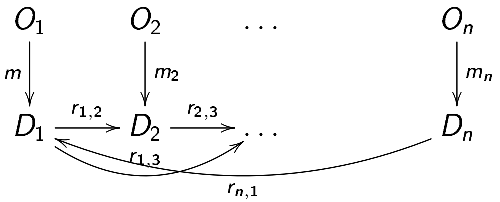
Class: $Thing_S$ **Class:** $Thing_T$
Class: *Person_HumanBeing* **SubClassOf:** $Thing_S$, $Thing_T$
Class: *Male* **Class:** *Employee*
Class: *Child* **SubClassOf:** $Thing_T$ **and** $\neg$ *Employee*
Individual: *alex* **Types:** *Male*

Example: integrated semantics (cont'd)

ontology C =
Class: *ThingS*
Class: *ThingT*
Class: *Person_HumanBeing* **SubClassOf:** *ThingS*, *ThingC*
Class: *Male* **SubClassOf:** *Person_HumanBeing*
Class: *Employee* **SubClassOf:** *ThingT*
Class: *Child* **SubClassOf:** *ThingS*
Class: *Child* **SubClassOf:** *ThingT* **and** ¬ *Employee*
Individual: *alex* **Types:** *Male*, *Person_HumanBeing*

Contextualized semantics of diagrams

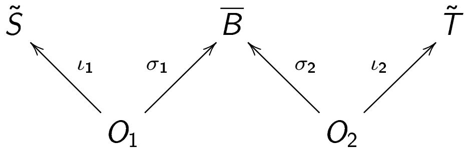
Framework: different domains related by coherent relations



such that

- r_{ij} is functional and injective,
- r_{ii} is the identity (diagonal) relation,
- r_{ji} is the converse of r_{ij}, and
- r_{ik} is the relational composition of r_{ij} and r_{jk}
- model for a diagram: family (m_i) of models with coherent relations (r_{ij}) such that $(m_i, r_{ji}m_j)$ is a model for A_{ij}

Contextualized semantics of diagrams, revisited



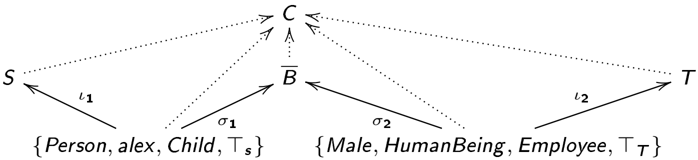
where $\overline{B}$ modifies B as follows:

- r_{ij} are added to $\overline{B}$ as roles with domain $\top_S$ and range $\top_T$
- the correspondences are translated to axioms involving these roles:
  - $s_i = t_j$ becomes $s_i \; r_{ij} \; t_j$
  - $a_i \in c_j$ becomes $a_i \in \exists r_{ij}.c_j$
  - ...
- the properties of the roles are added as axioms in $\overline{B}$

Adding domain relations to the bridge

ontology $\overline{B}$ =
Class: *ThingS*
Class: *ThingT*
ObjectPropery: r_{ST} **Domain:** *ThingS* **Range:** *ThingT*
Class: *Person* **EquivalentTo:** r_{ST} **some** *HumanBeing*
Class: *Employee*
Class: *Child* **SubClassOf:** r_{ST} **some** $\neg$ *Employee*
Individual: *alex* **Types:** r_{ST} **some** *Male*

Example: contextualized semantics



where

ontology C =

Class: *ThingS*
Class: *ThingT*
ObjectPropery: r_{ST} **Domain:** *ThingS* **Range:** *ThingT*
Class: *Person* **EquivalentTo:** r_{ST} **some** *HumanBeing*
Class: *Employee*
Class: *Child* **SubClassOf:** r_{ST} **some** ¬ *Employee*
Individual: *alex* **Types:** r_{ST} **some** *Male*, *Person*

Queries

DOL is a logical (meta) language

- focus on ontologies, models, specifications,
- and their logical relations: logical consequence, interpretations, ...

Queries are different:

- answer is not "yes" or "no", but an answer substitution
- query language may differ from language of OMS that is queried

Sample query languages

- conjunctive queries in OWL
- Prolog/Logic Programming
- SPARQL

Syntax of queries in DOL

New OMS declarations and relations:

```
query qname = select vars where sentence in OMS
                [along language-translation]
substitution sname : OMS1 to OMS2 = derived-symbol-map
result rname = sname_1, ..., sname_n for qname
              %% result is a substitution
```

New sentences (however, as structured OMS!):

```
apply(sname,sentence)     %% apply substition
```

For derived signature morphisms, see my WADT talk.
Open question: how to deal with “construct” queries?

Semantics of queries in DOL

Based on: R. Diaconescu: Herbrand theorems in arbitrary institutions. Information Processing Letters 90 (2004) 29–37.

query qname = **select vars where** sentence **in** OMS

$\exists\chi$.sentence, where $\chi\colon Sig[OMS] \to Sig[OMS] \cup vars$ is a signature morphism

substitution sname : OMS1 **to** OMS2 = derived-symbol-map

Same semantics as interpretation or view. Semantics of derived signature morphisms are abstract substitutions, see paper.

result rname = sname_1, ..., sname_n **for** qname

is well-defined iff

$$OMS \models \forall\chi.apply(sname\_i, sentence)$$

Semantics of queries in DOL, cont'd

```
result rname = sname_1, ..., sname_n for qname
                                           %complete%
```

Is well-defined iff ($OMS \models \forall\chi.apply(\theta, sentence)$ iff θ is among sname_1, ..., sname_n)

```
apply(sname,sentence)
```

$Sen(\psi)$(sentence), where ψ is the abstract substitution corresponding to derived-symbol-map.

Conclusion

Challenges

- What is a suitable abstract meta framework for non-monotonic logics and rule languages like RIF and RuleML? Are institutions suitable here? 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 OMS?
- 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 OMS languages and translations?
- Proof support

Tool support: Heterogeneous Tool Set (Hets)

- available at `hets.dfki.de`
- speaks DOL, HetCASL, CoCASL, CspCASL, MOF, QVT, OWL, Common Logic, and other languages
- analysis
- computation of colimits
- management of proof obligations
- interfaces to theorem provers, model checkers, model finders

Tool support: Ontohub web portal and repository

Ontohub is a web-based repository engine for distributed heterogeneous (multi-language) OMS

- 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

Equivalences

- **equivalence** *Id* : $O_1 \leftrightarrow O_2 = O_3$
- (fragment) OMS O_3 is such that O_i **then %def** O_3 is a definitional extension of O_i for $i = 1, 2$;
- this implies that O_1 and O_2 have model classes that are in bijective correspondence

```
equivalence e : algebra:BooleanAlgebra
                ↔ algebra:BooleanRing =
    x∧y = x·y
    x∨y = x+y+x·y
    ¬x = 1+x
    x·y = x∧y
    x+y = (x∨y) ∧ ¬(x∧y)
end
```

Module Relations

- **module** *Id* c : O_1 **of** O_2 **for** Σ
- O_1 is a module of O_2 with restriction signature Σ and conservativity c

  c=%mcons every Σ-reduct of an O_1-model can be expanded to an O_2-model

  c=%ccons every Σ-sentence φ following from O_1 already follows from O_1

This relation shall hold for any module O_1 extracted from O_2 using the **extract** construct.

Conclusion

- DOL is a meta language for (formal) ontologies, specifications and models (OMS)
- DOL covers many aspects of modularity of and relations among OMS ("OMS-in-the large")
- DOL will be submitted to the OMG as an answer to the OntoIOp RFP
- you can help with joining the OntoIOp discussion
  - see `ontoiop.org`

Related work

- Structured specifications and their semantics (Clear, ASL, CASL, . . .)
- Heterogeneous specification (HetCASL)
- modular ontologies (WoMo workshop series)