An Algebra of Lightweight Ontologies

Marco A. Casanova

Departamento de Informática, PUC-Rio

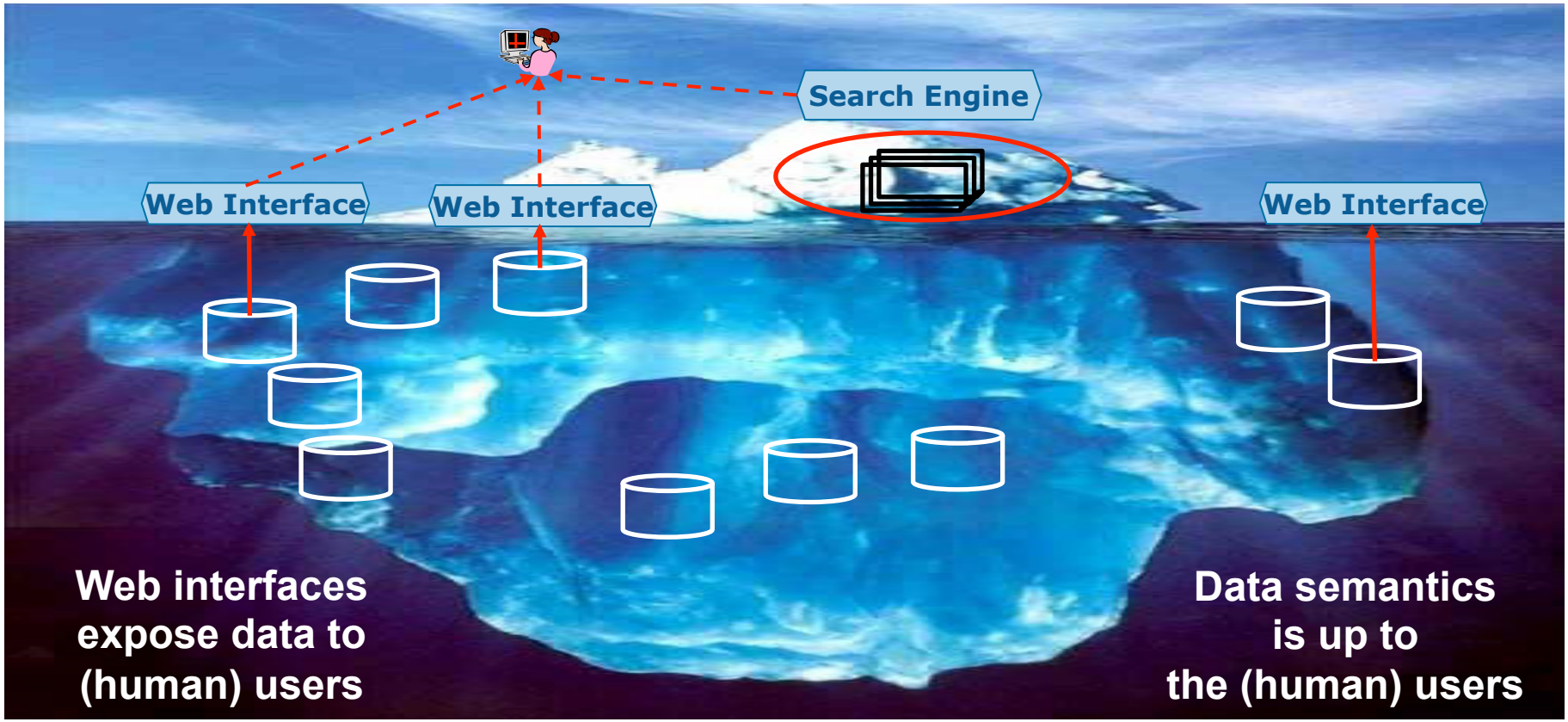
Topics

- **Introduction**
- **A Formal Framework**
- **Operations over Ontologies**
- **Implementation of the Operations**
- **Conclusions**

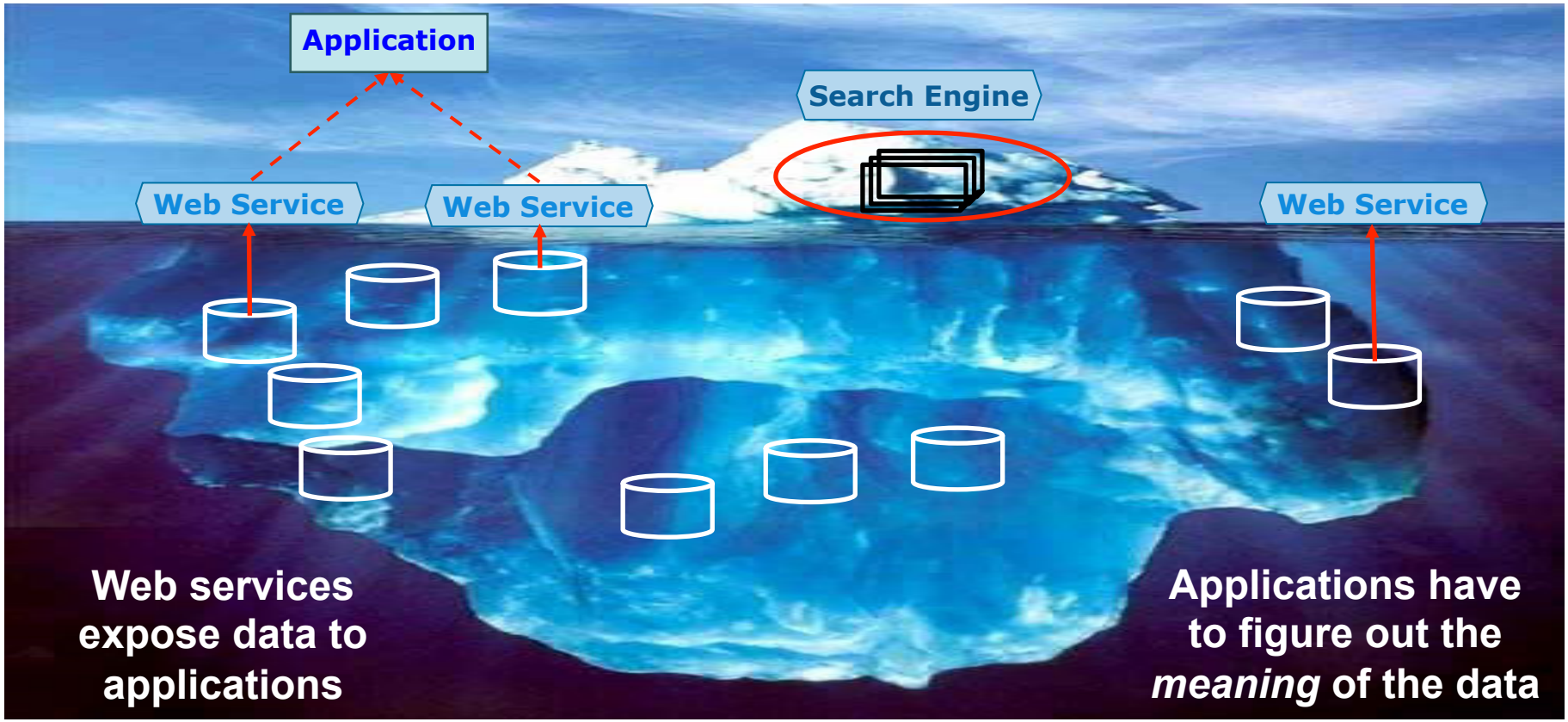
Introduction

- **Problem addressed**
  - How to design databases to be published on the Web (so that software agents can understand the data)
- **Contexts**
  - Databases in a traditional Web environment
  - Triplesets in the Linked Data Cloud

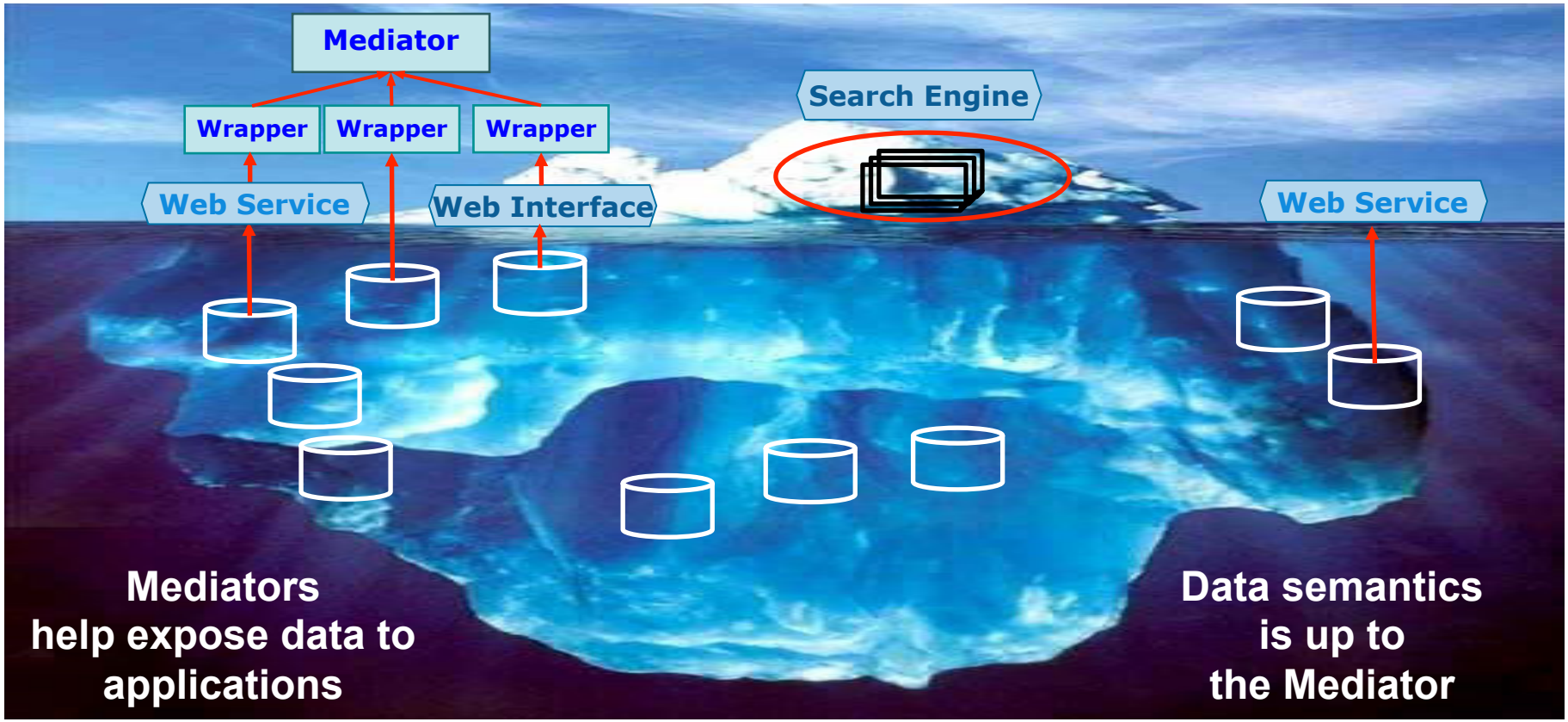
Introduction



Introduction



Introduction



(Digression)

- **Question**

What is the meaning of "meaning"?

(Digression)

- **Question**

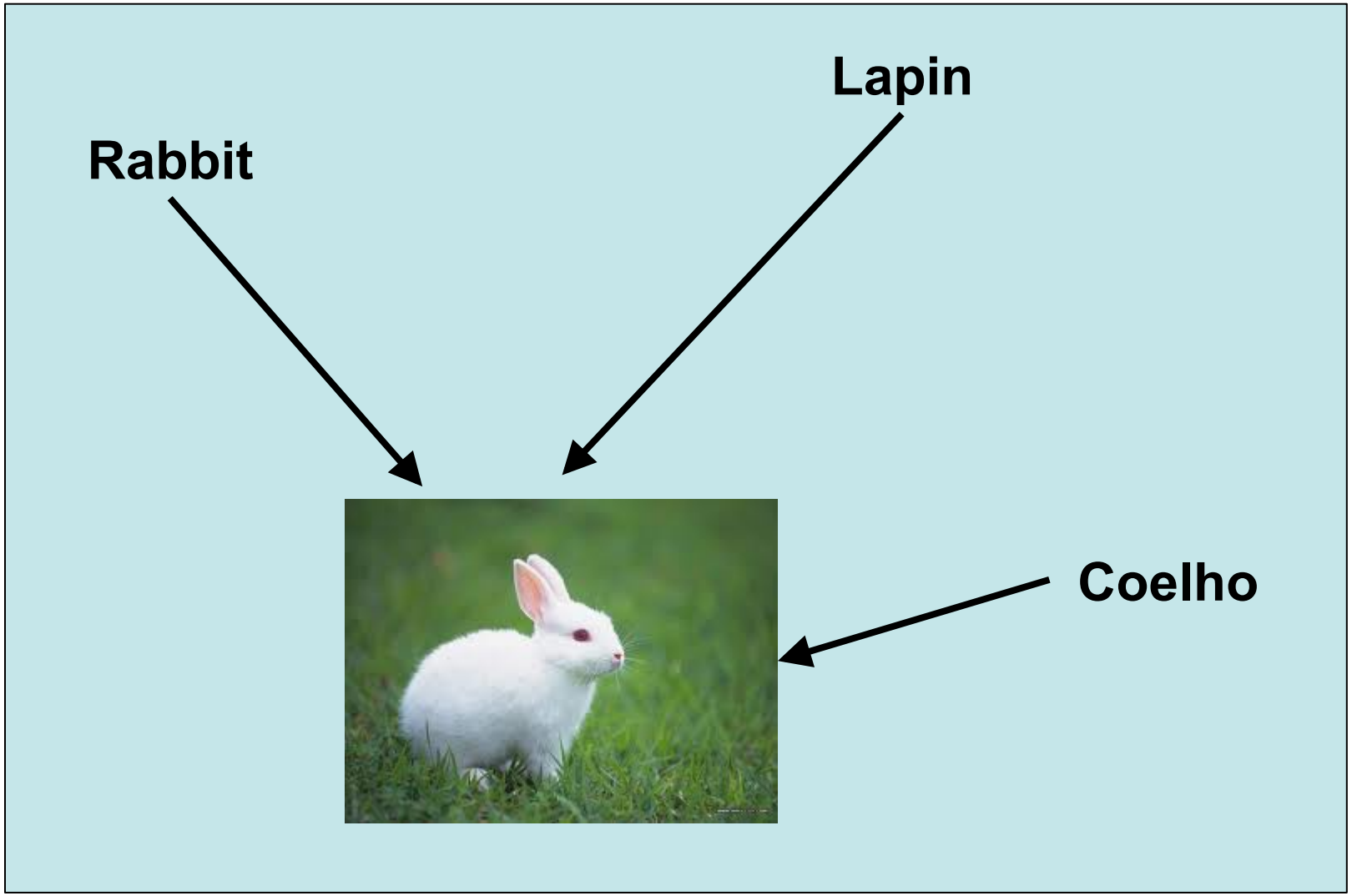
What is the meaning of "meaning"?

- **Answer**

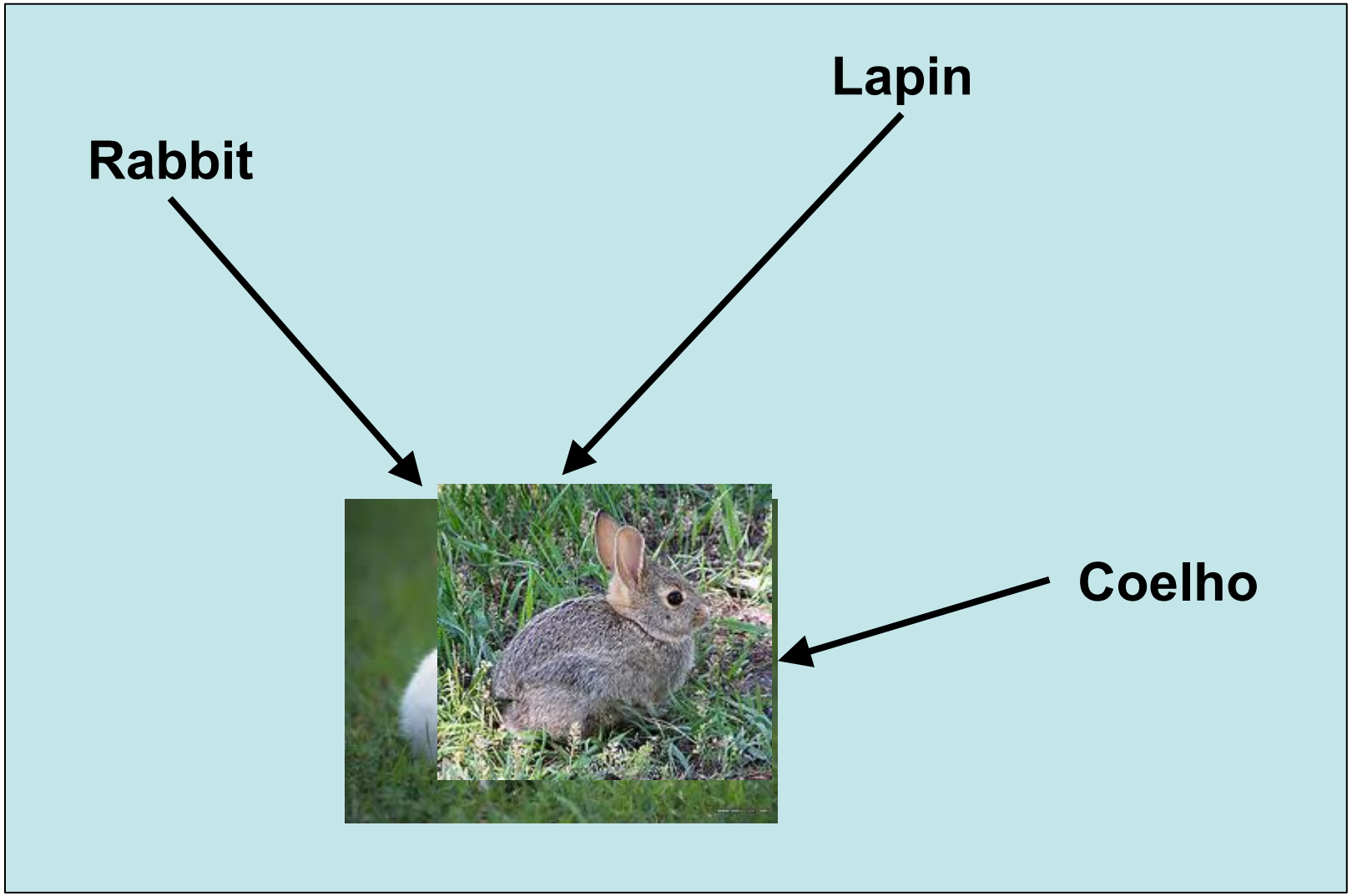
"Terms have the same extension when true of the same things"

Quine, W. V. (1968). Ontological relativity. *The Journal of Philosophy, 65(7):185–212.*

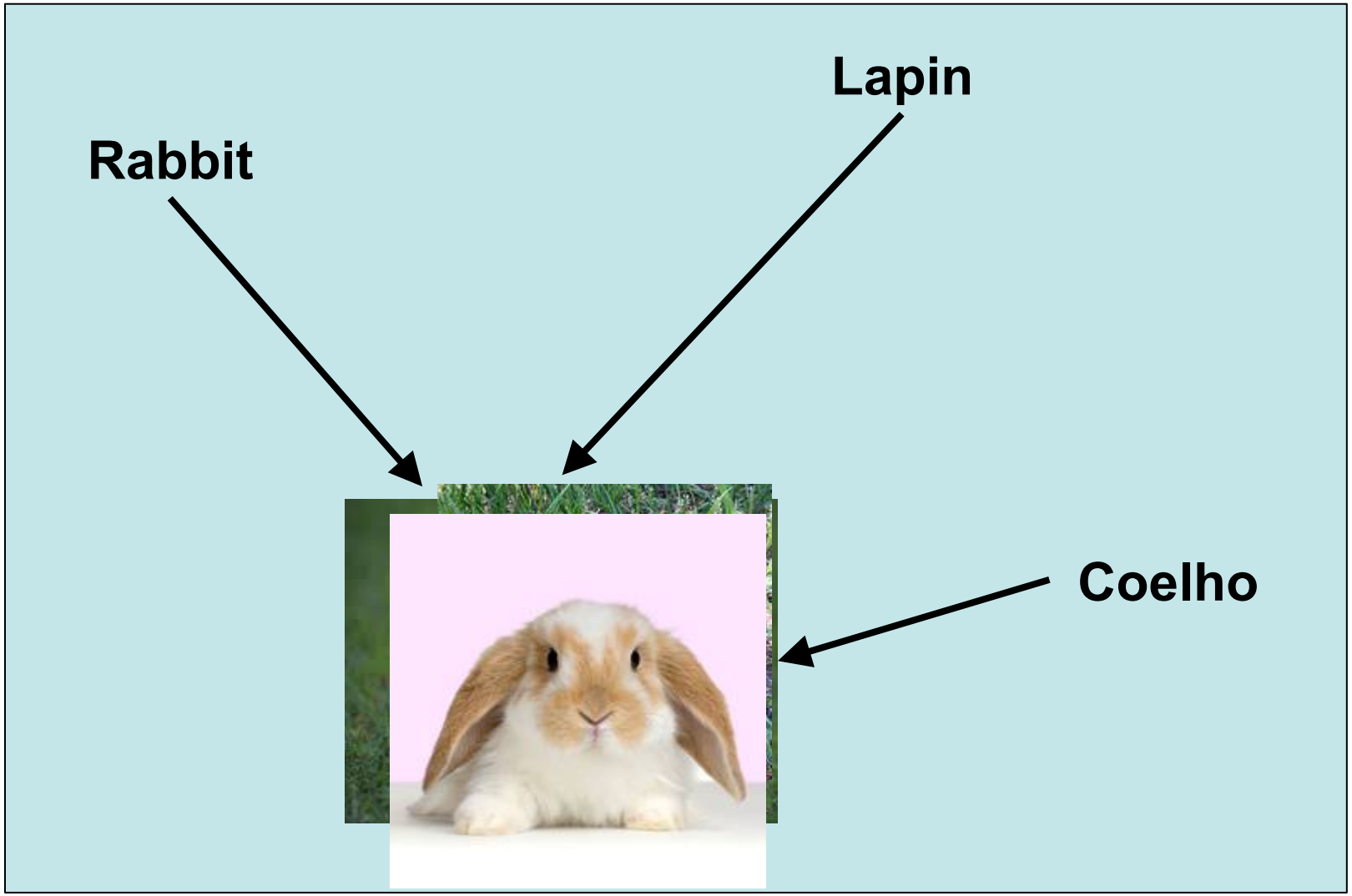
(Digression)



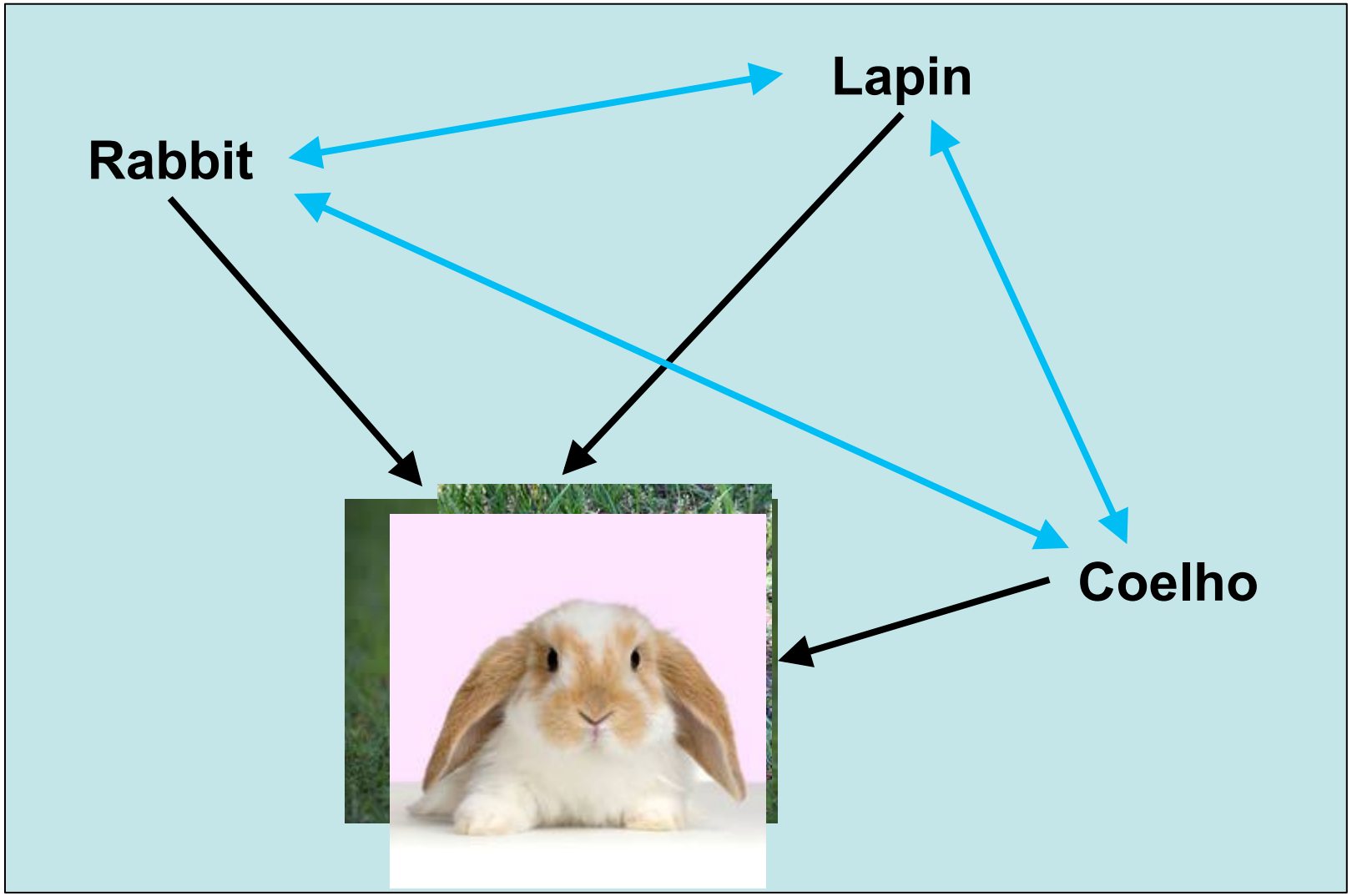
(Digression)



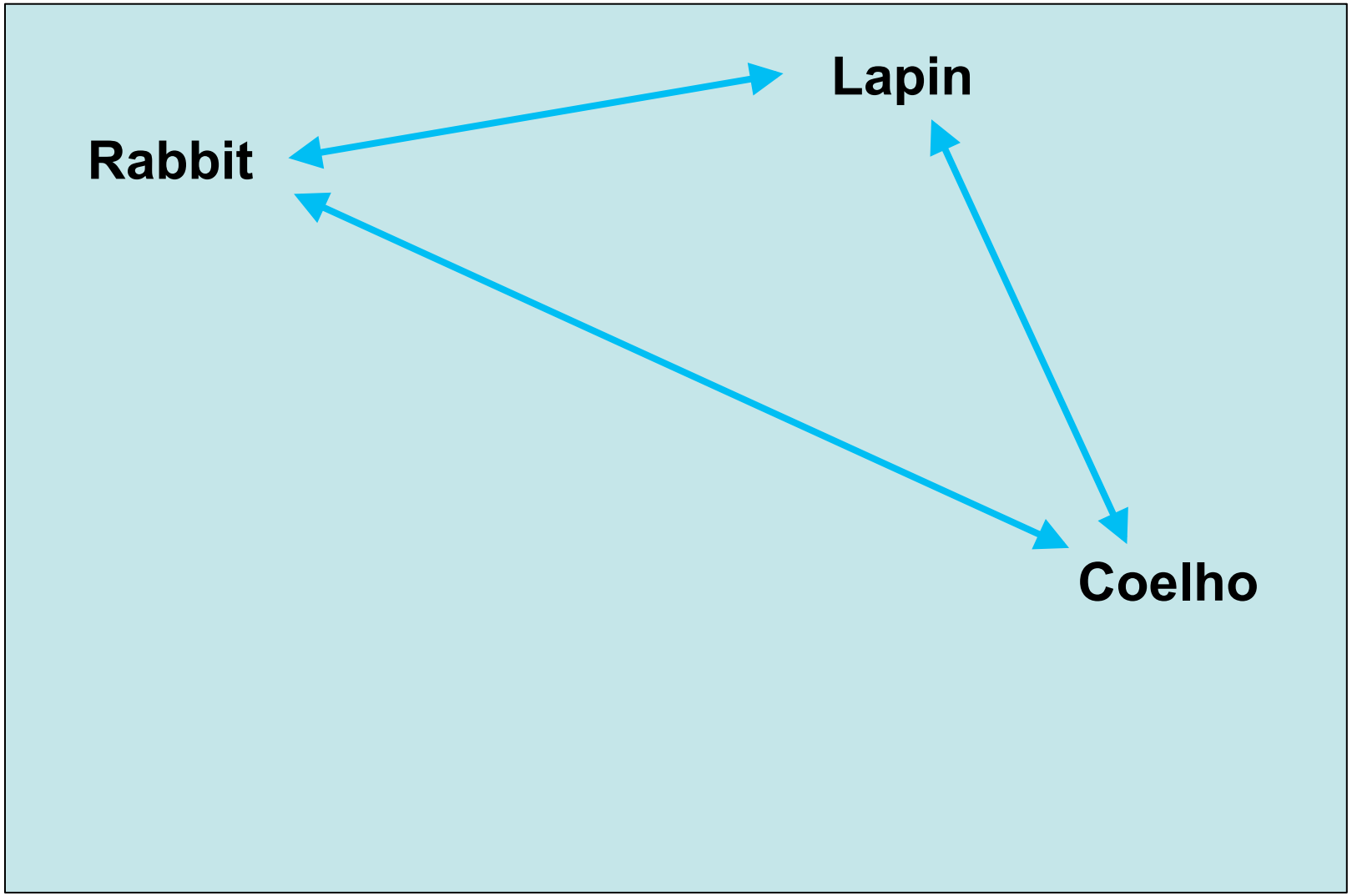
(Digression)



(Digression)



(Digression)



(Digression)

- **Question**

How an application figures out the meaning of the data?

(Digression)

- **Question**

How an application figures out the meaning of the data?

- **Suggested Answer**

"By matching its own conceptual schema with the database external schema"

(Digression)

- **Problem**

Schema matching
is an intractable problem!

(Digression)

- **Problem**

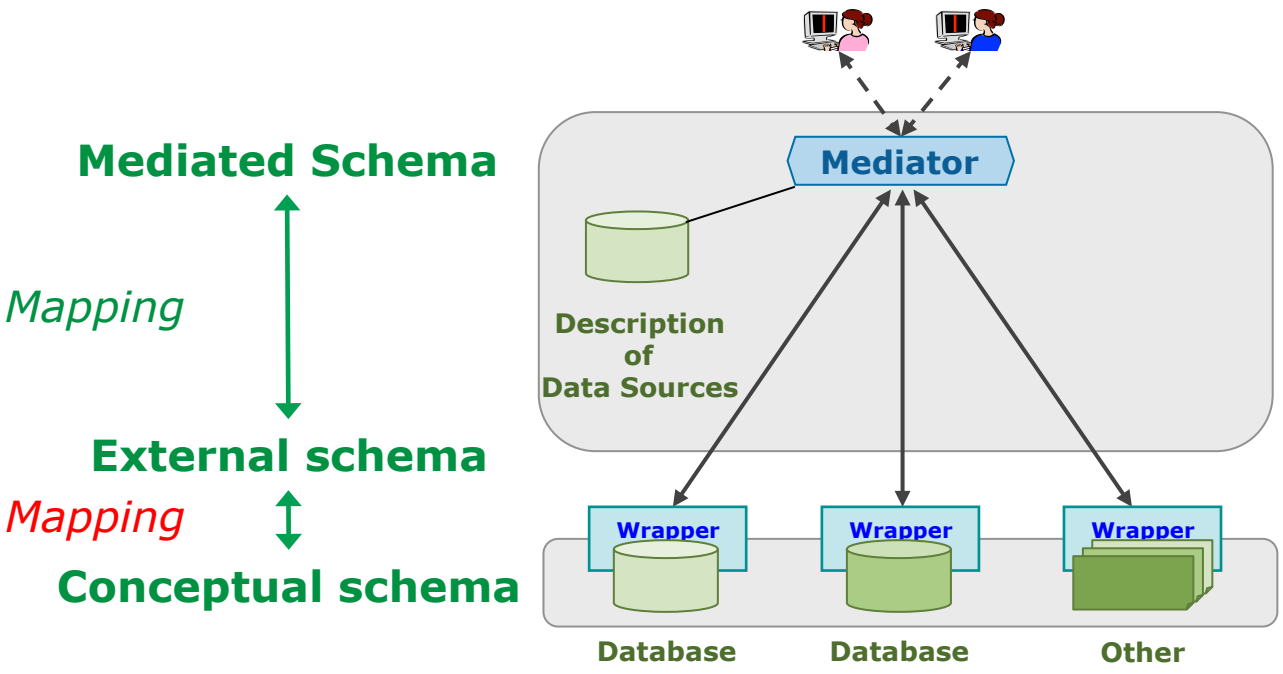
Schema matching is an intractable problem!

- **Suggested Solution**

Ontologies!

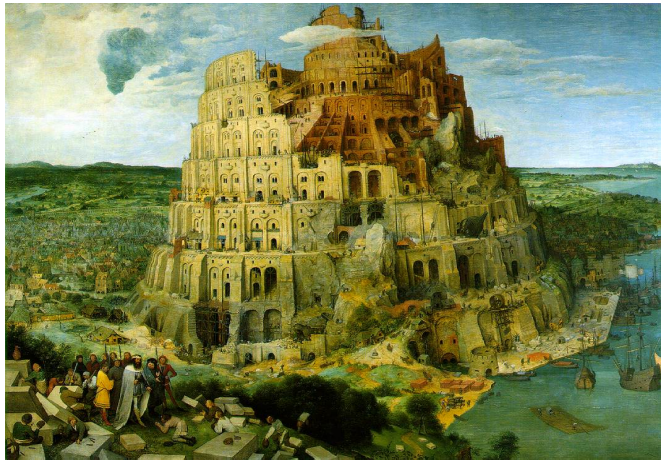
Introduction

- **Mediators**



Introduction

- **Suggested Solution**
  - An external schema should be
    a combination of fragments
    of one or more domain ontologies



Bruegel, Pieter. The Tower of Babel. c 1563

Oil on oak panel. 114 x 155 cm

Kunsthistorisches Museum Wien, Vienna

Introduction

- **Suggested Solution**
  - An external schema should be
    a combination of fragments
    of one or more domain ontologies

    so that matching
    the application schema
    with the external schema
    should become a *non-problem*
    (or it will remain an intractable problem)

Bruegel, Pieter. The Tower of Babel. c 1563

Oil on panel. 60 × 74.5 cm

Museum Boijmans Van Beuningen, Rotterdam, Netherlands

Introduction

- **Suggested Solution**
  - An external schema should be
    a combination of fragments
    of one or more domain ontologies

    so that matching
    the application schema
    with the external schema
    should become a *non-problem*
    (or it will remain an intractable problem)
  - **"Standards for everything"**
    - **Domain ontologies**
    - **Object Ids**
    - **…**

van Valckenborch , Lucas . The Tower of Babel. 1568
Oil on panel. 41 × 56 cm
Galerie de Jonckheere, Paris, France

Introduction

- **Problem**
  - What is the *meaning* of "a combination of fragments of one or more domain ontologies"?

    (So that matching the application schema with the external schema becomes a ***non-problem***)

van Valckenborch I, Marten. The Building of the Tower of Babel. 1595

(more paintings of the Tower of Babel at http://babelstone.blogspot.com.br/2008/12/72-more-views-of-tower-of-babel.html)

Topics

- Introduction
- **A Formal Framework**
  - A Simple Example
  - Constraints
  - Constraint Graphs
- Operations over Ontologies
- Implementation of the Operations
- Conclusions

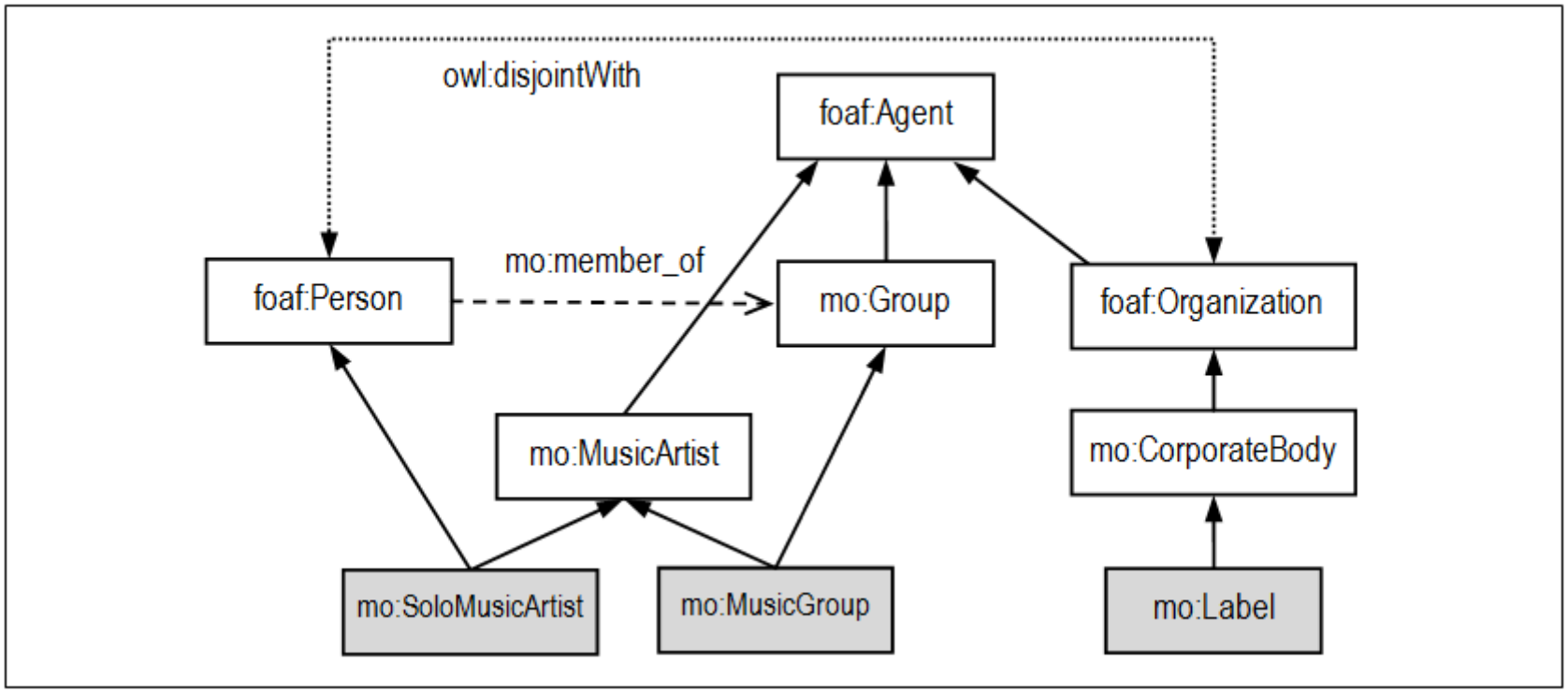
A Simple Example

A Formal Framework

- **Example:**
  - Domain ontology:
    - The Music Ontology
  - External schema:
    - a fragment of the Music Ontology
    - includes solo artists, music groups and (record) labels

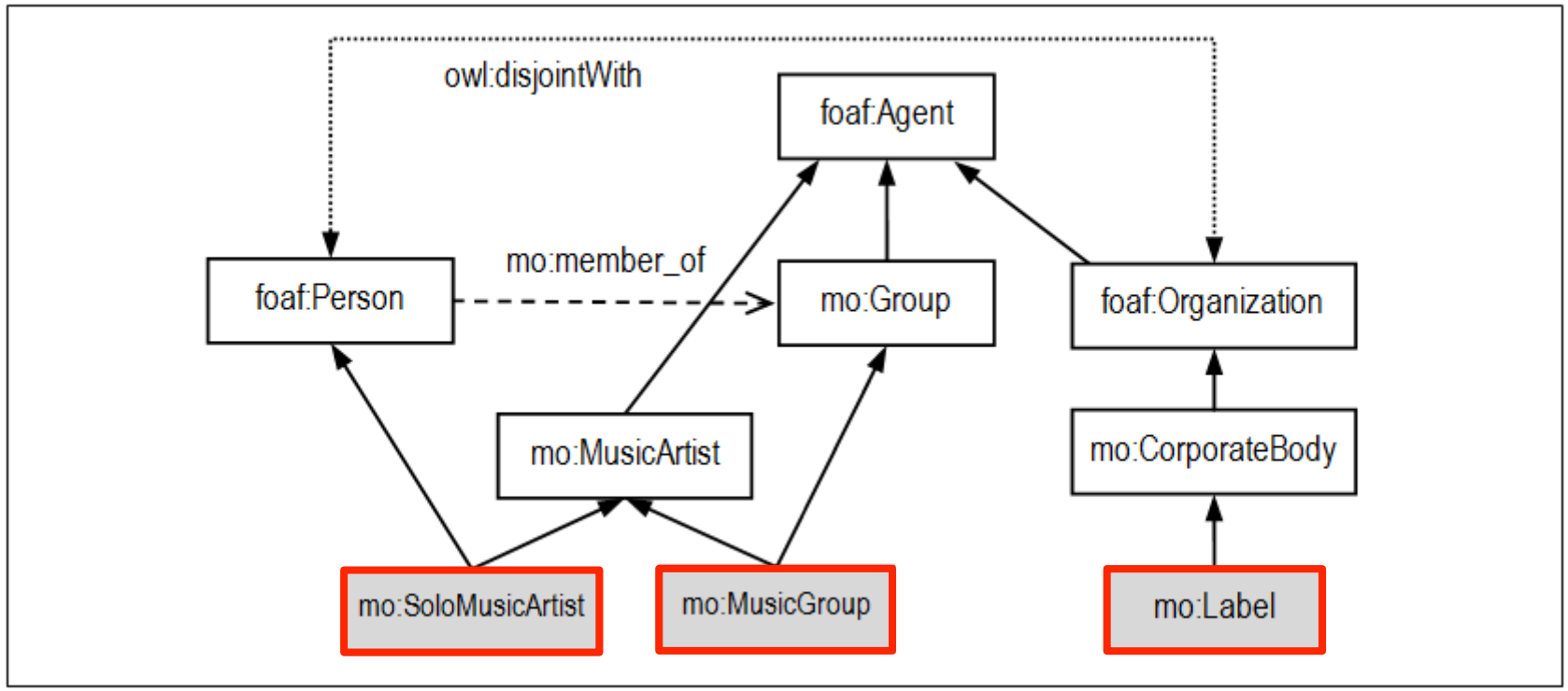
A Simple Example

A Formal Framework



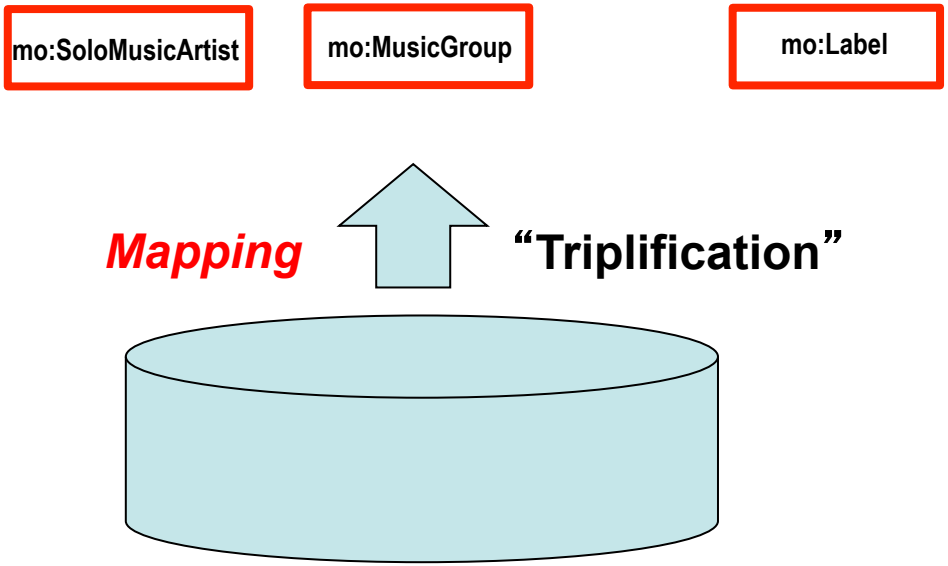
A Simple Example

A Formal Framework



A Simple Example

A Formal Framework



A Simple Example

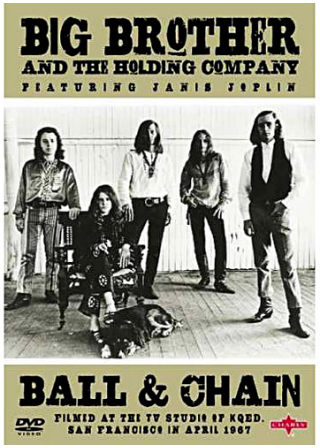
A Formal Framework

mo:SoloMusicArtist

mo:MusicGroup

mo:Label

(uri1, mo:SoloMusicArtist, “Janis Joplin”)
(uri2, mo:MusicGroup, “Big Brother and the Holding Company”)
(uri3, mo:Label, “Columbia”)



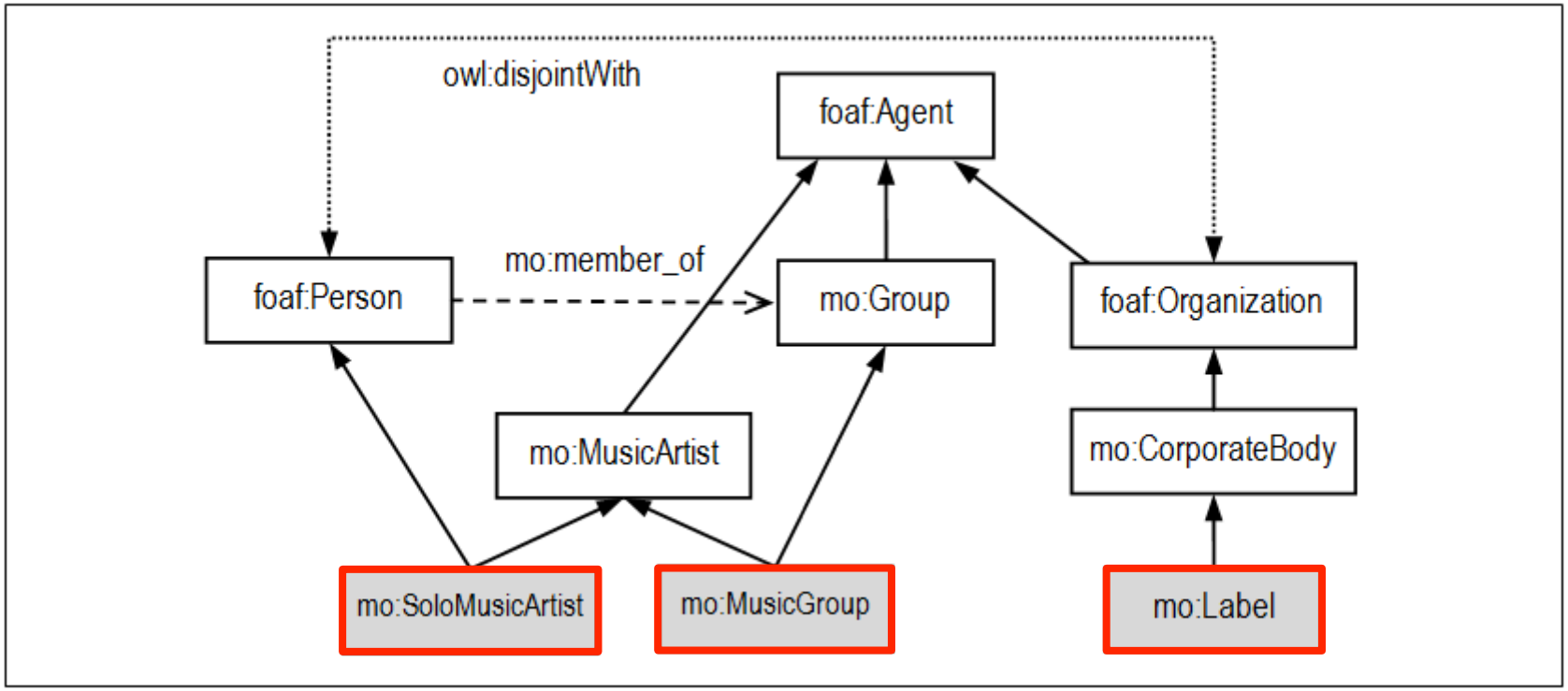
A Simple Example

A Formal Framework

Done!

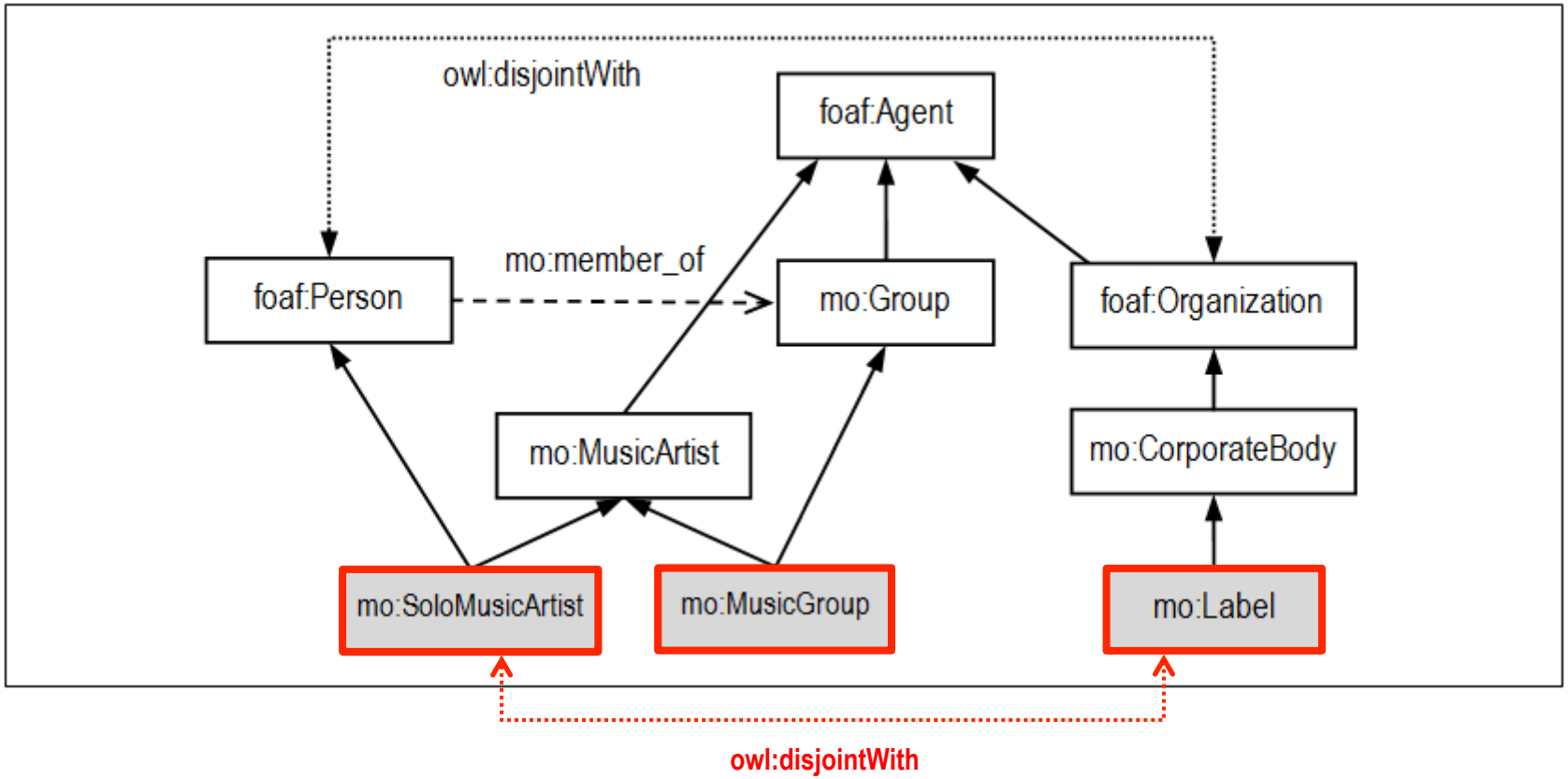
A Simple Example

A Formal Framework



A Simple Example

A Formal Framework



(Intermezzo)

- **What is the point here?**
  - the domain ontology constraints play an essential role when designing the external schema
    - they carry the semantics of the terms of the domain ontology vocabulary
    - the design process cannot be reduced to merely copying the constraints from the domain ontology to the external schema
    - the design process must take into account constraints derived from those of the domain ontology

(Intermezzo)

- **Design process:**
  - Designer ***constructs***
    the vocabulary of the external schema
    by selecting classes and properties from
    the vocabulary of the domain ontology
  - "Black Box" ***constructs***
    the constraints of the external schema
    as the set of all constraints that are derivable from
    the constraints of the domain ontology
    and that apply to the vocabulary of the external schema

Constraints

A Formal Framework

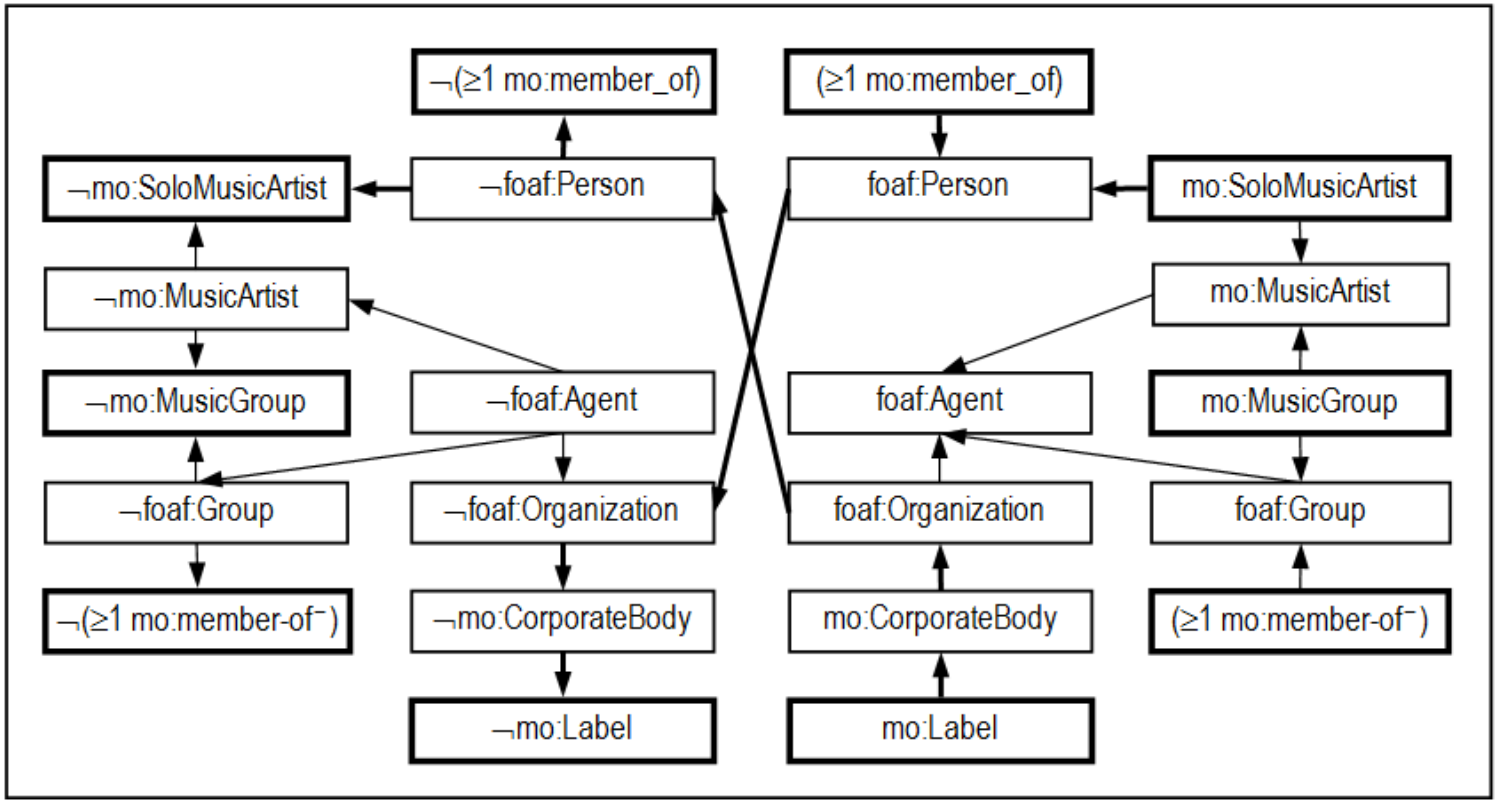
- **Lightweight constraints**

Constraint Type	Formalization	Unabbreviated form	Informal semantics
Domain Constraint	$\exists P \sqsubseteq D$	$(\geq 1\ P) \sqsubseteq D$	property P has class D as domain, that is, if (a,b) is a pair in P, then a is an individual in D
Range Constraint	$\exists P^- \sqsubseteq R$	$(\geq 1\ P^-) \sqsubseteq R$	property P has class R as range, that is, if (a,b) is a pair in P, then b is an individual in R
minCardinality Constraint	$C \sqsubseteq (\geq k\ P)$ or $C \sqsubseteq (\geq k\ P^-)$		property P or its inverse P^- maps each individual in class C to at least k distinct individuals
maxCardinality Constraint	$C \sqsubseteq (\leq k\ P)$ or $C \sqsubseteq (\leq k\ P^-)$	$C \sqsubseteq \neg(\geq k+1\ P)$ or $C \sqsubseteq \neg(\geq k+1\ P^-)$	property P or its inverse P^- maps each individual in class C to at most k distinct individuals
Subset Constraint	$E \sqsubseteq F$		each individual in E is also in F, that is, class E denotes a subset of class F
Disjointness Constraint	$E \mid F$	$E \sqsubseteq \neg F$	no individual is in both E and F, that is, classes E and F are disjoint

Constraint Graphs

A Formal Framework

$A \cap B = \emptyset$
iff
$A \subseteq \neg B$
iff
$B \subseteq \neg A$



A Decision Procedure

A Formal Framework

- **The IMPLIES procedure**
  - A sound and complete procedure to test logical implication for lightweight constraints

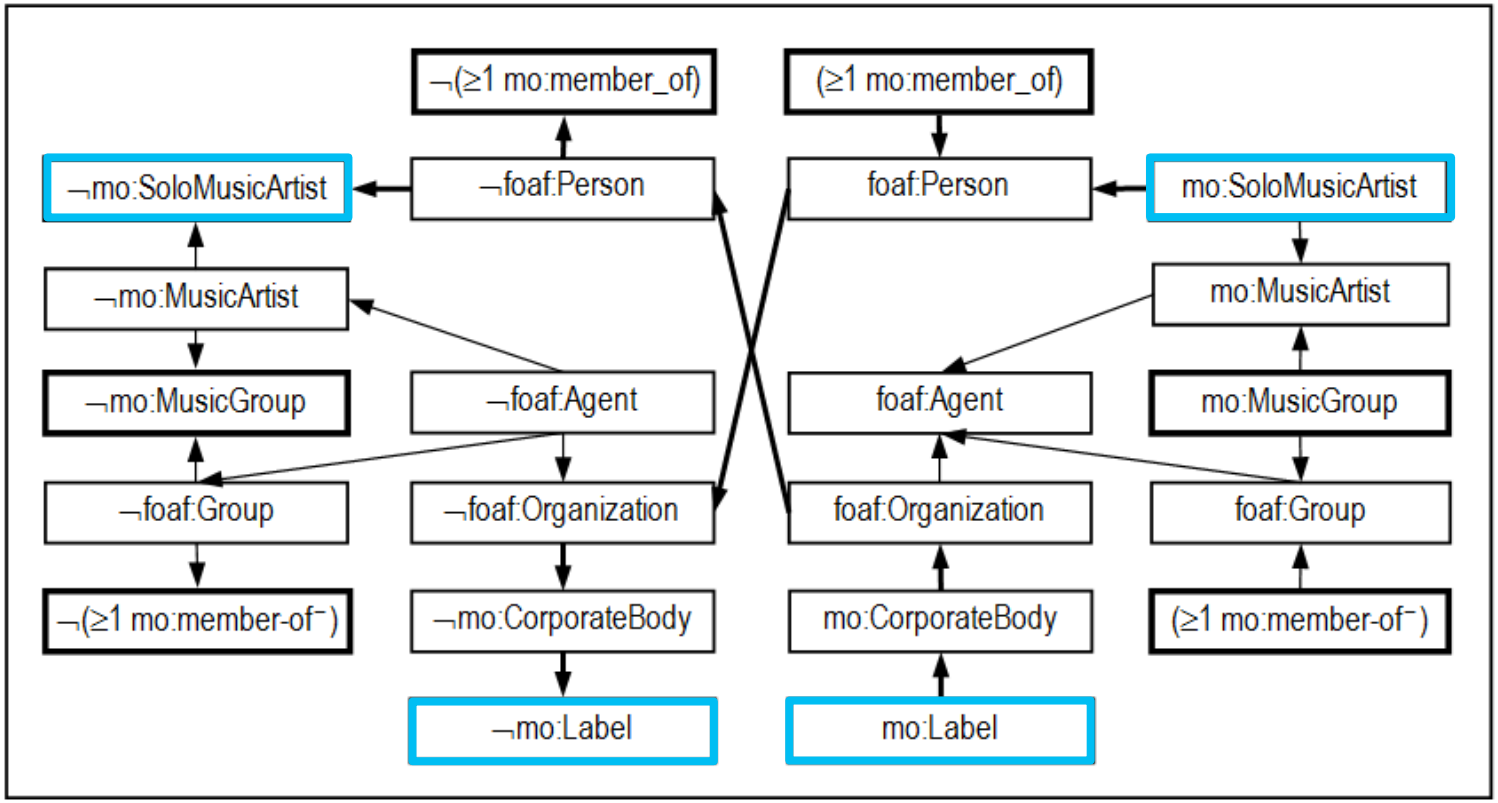
IMPLIES($\Sigma, e \sqsubseteq f$)
input: a set Σ of unabbreviated inclusions and an unabbreviated inclusion $e \sqsubseteq f$
output: "YES - Σ logically implies $e \sqsubseteq f$"
"NO - Σ does not logically imply $e \sqsubseteq f$"

begin Construct $G(\Sigma, \{e,f\})$, the constraint graph for Σ and $\{e,f\}$;
if the node of $G(\Sigma, \{e,f\})$ labeled with e is a $\bot$-node, or
the node of $G(\Sigma, \{e,f\})$ labeled with f is a $\top$-node, or
there is a path in $G(\Sigma,\{e,f\})$ from the node labeled with e
to the node labeled with f,
then return "YES - Σ logically implies $e \sqsubseteq f$";
else return "NO - Σ does not logically imply $e \sqsubseteq f$";
end

A Decision Procedure

A Formal Framework

$A \cap B = \emptyset$
iff
$A \subseteq \neg B$
iff
$B \subseteq \neg A$



A Decision Procedure

A Formal Framework

- **Constraint graphs**
  - inspired on a 2-SAT solver

- **Completeness proof**
  - Constructs a Herbrand model
    - Use constants to represent classes
    - Use function symbols to represent number restrictions
  - Depends heavily on the fact that the left-hand side of a lightweight inclusion is a positive expression
  - *(Seems to carry on to DL-lite core with arbitrary number restrictions)*

Topics

- Introduction
- A Formal Framework
- **Operations over Ontologies**
  - Definition of the operations
  - Computing the operations
  - Optimization
- Implementation of the Operations
- Conclusions

Operations over Ontologies

- **Questions**
  - How to design an external schema?
  - How to compare two ontologies?
  - How to design a mediated ontology?

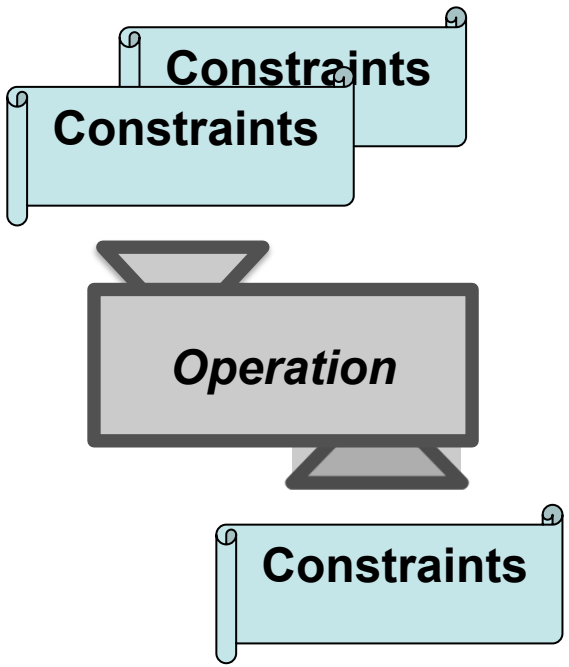
- **Answer**
  - Operations over ontologies

van Valckenborch I, Marten. The Building of the Tower of Babel. 1595

(more paintings of the Tower of Babel at http://babelstone.blogspot.com.br/2008/12/72-more-views-of-tower-of-babel.html)

Operations over Ontologies

- **Operations**
  - Create new ontologies, including their constraints, out of other ontologies
  - Treat an ontology $O=(V,\Sigma)$ as a theory, i.e., a set of constraints $\tau[\Sigma]$



Operations over Ontologies

- **Useful operations**:
  - *Projection*
    - The *projection* of $O_1 = (V_1,\Sigma_1)$ over W, denoted $\pi[W](O_1)$, returns the ontology $O_P = (V_P,\Sigma_P)$, where $V_P = W$ and Σ_P is the set of constraints in $\tau[\Sigma_1]$ that use only classes and properties in W
  - *Deprecation*
    - The *deprecation* of Ψ from $O_1 = (V_1,\Sigma_1)$, denoted $\delta[\Psi](O_1)$, returns the ontology $O_D = (V_D,\Sigma_D)$, where $V_D = V_1$ and $\Sigma_D = \Sigma_1 - \Psi$

Operations over Ontologies

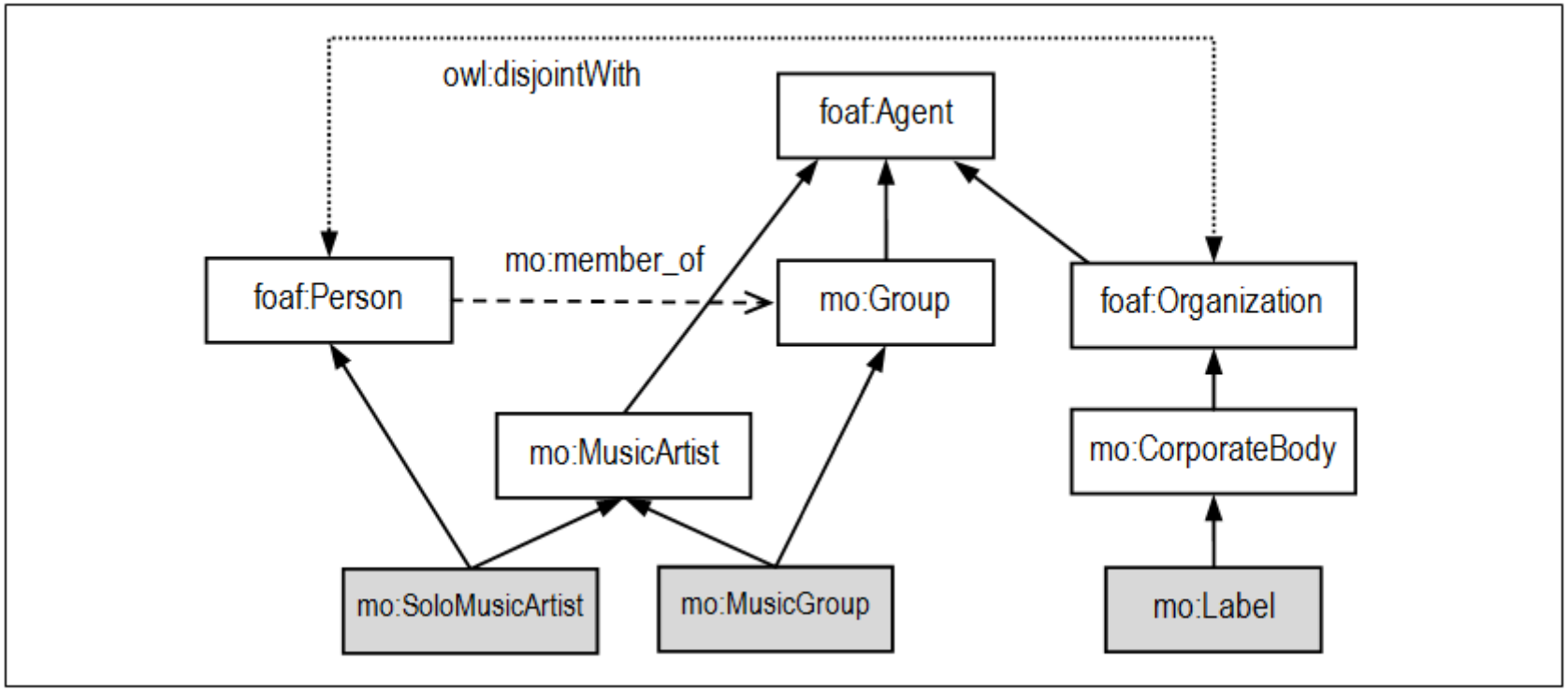
- *Union*
  - The *union* of $O_1 = (V_1,\Sigma_1)$ and $O_2 = (V_2,\Sigma_2)$, denoted $O_1 \cup O_2$, returns the ontology $O_U = (V_U,\Sigma_U)$, where $V_U = V_1 \cup V_2$ and $\Sigma_U = \Sigma_1 \cup \Sigma_2$
- *Intersection*
  - The *intersection* of $O_1 = (V_1,\Sigma_1)$ and $O_2 = (V_2,\Sigma_2)$, denoted $O_1 \cap O_2$, returns the ontology $O_N = (V_N,\Sigma_N)$, where $V_N = V_1 \cap V_2$ and $\Sigma_N = \tau[\Sigma_1] \cap \tau[\Sigma_2]$
- *Difference*
  - The *difference* of $O_1 = (V_1,\Sigma_1)$ and $O_2 = (V_2,\Sigma_2)$, denoted $O_1 - O_2$, returns the ontology $O_F = (V_F,\Sigma_F)$, where $V_F = V_1$ and $\Sigma_F = \tau[\Sigma_1] - \tau[\Sigma_2]$

Operations over Ontologies

Question	Operations
How to design an external schema	Projection, Union, Deprecation
How to compare two ontologies	Intersection Difference
How to design a mediated ontology	Intersection

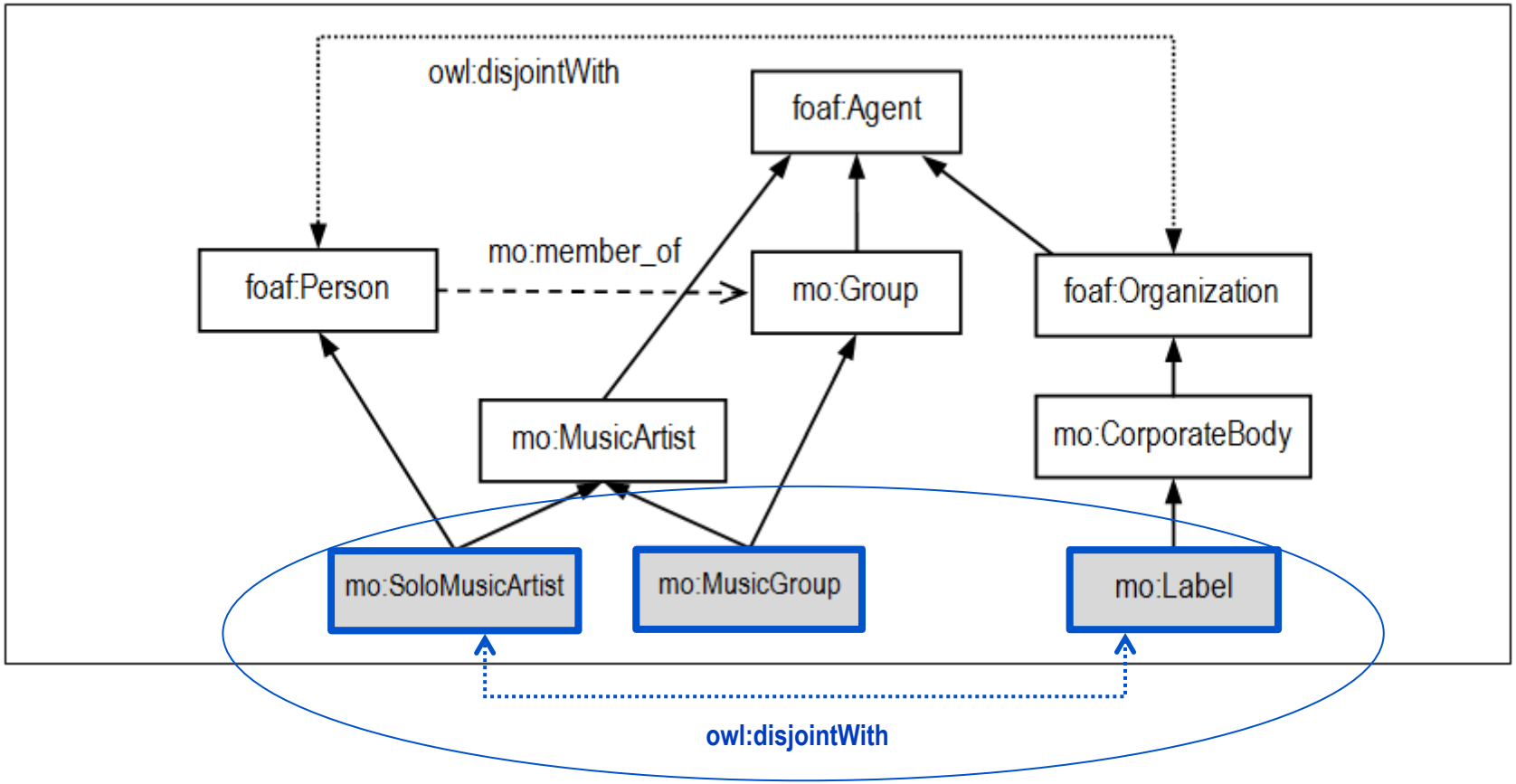
Operations over Ontologies

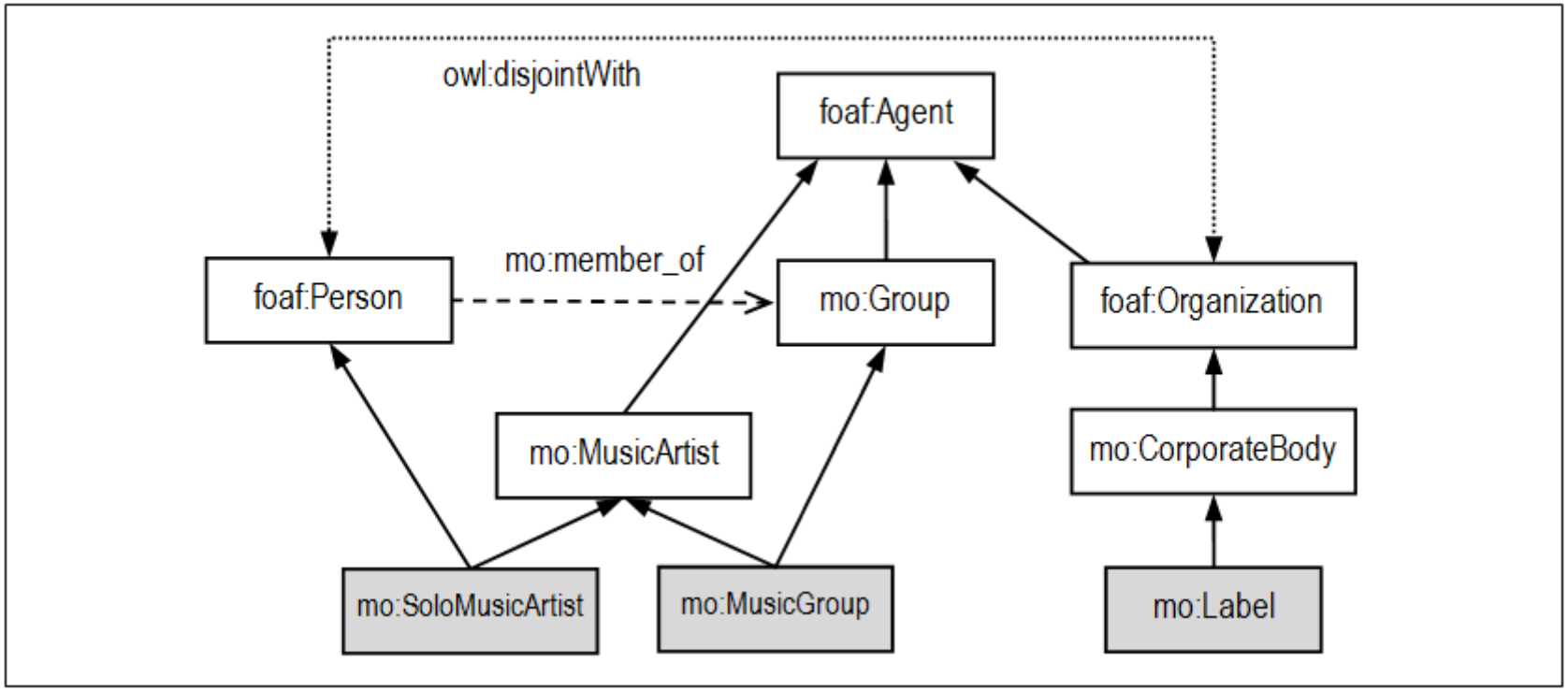
- **Example: a fragment of the Music Ontology**



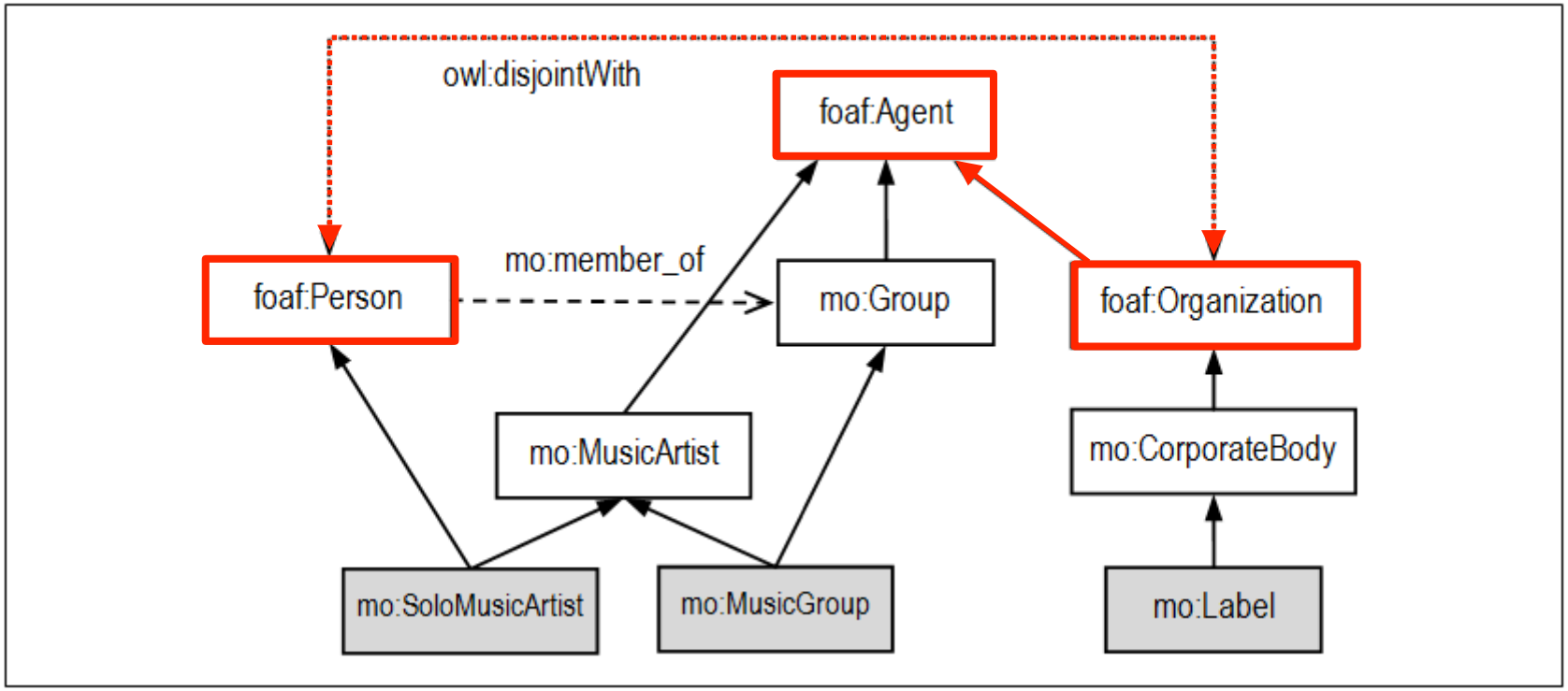
$O = \pi[W](MO)$ where W = { mo:SoloMusicArtist, mo:MusicGroup, mo:Label }

$O = (V,S)$ where $V = W$ and S = { mo:SoloMusicArtist $\sqsubseteq \neg$mo:Label }




owl:disjointWith
foaf:Agent
mo:member_of
foaf:Person
mo:Group
foaf:Organization
mo:MusicArtist
mo:CorporateBody
mo:SoloMusicArtist
mo:MusicGroup
mo:Label

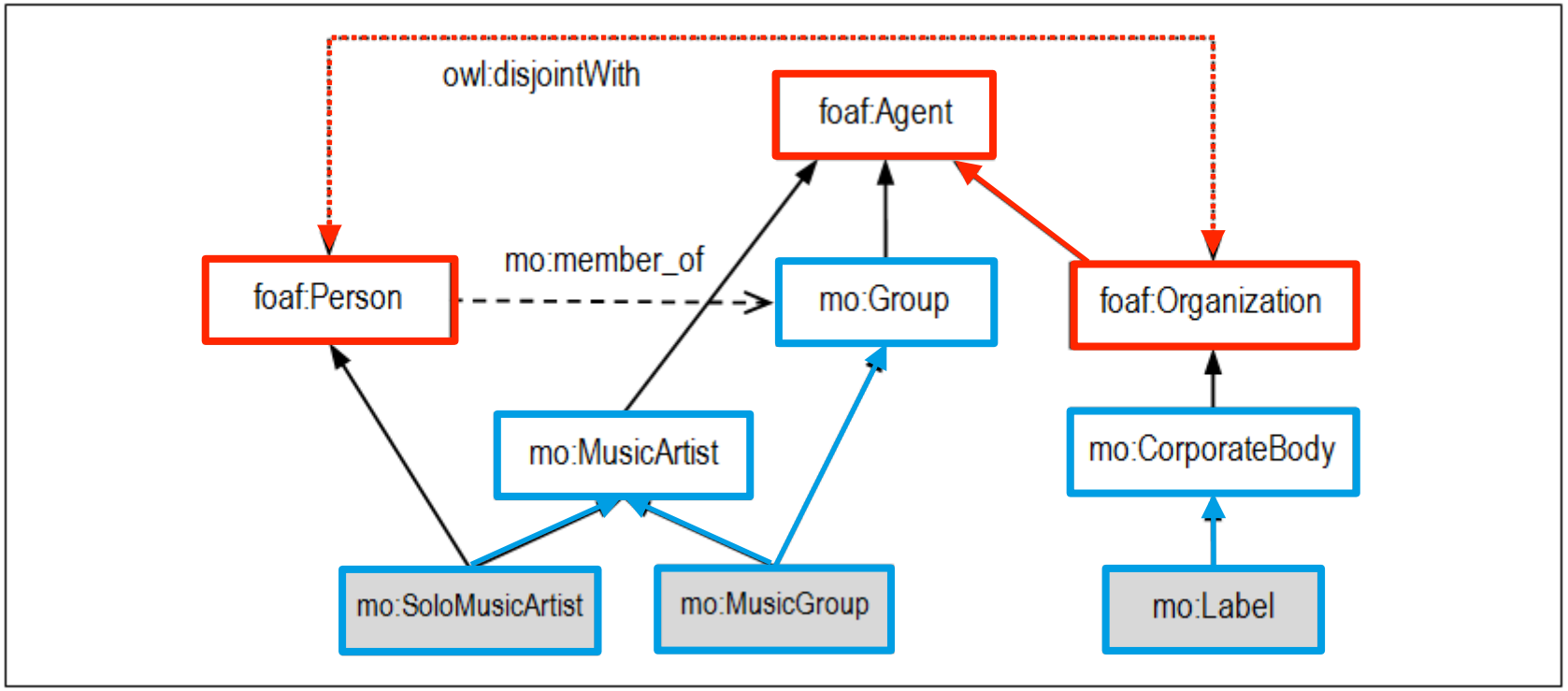
$O_1 = \pi[W](FOAF)$ where W = { foaf:person, foaf:Agent, foaf:Organization }



$O_2 = (V_2, S_2)$, where V_2 = { mo:Group, mo:MusicArtist, mo:CorporateBody, mo:SoloMusicArtist, mo:MusicGroup, mo:Label }

S_2 = {mo:SoloMusicArtist ⊑ mo:MusicArtist, mo:MusicGroup ⊑ mo:MusicArtist,… }

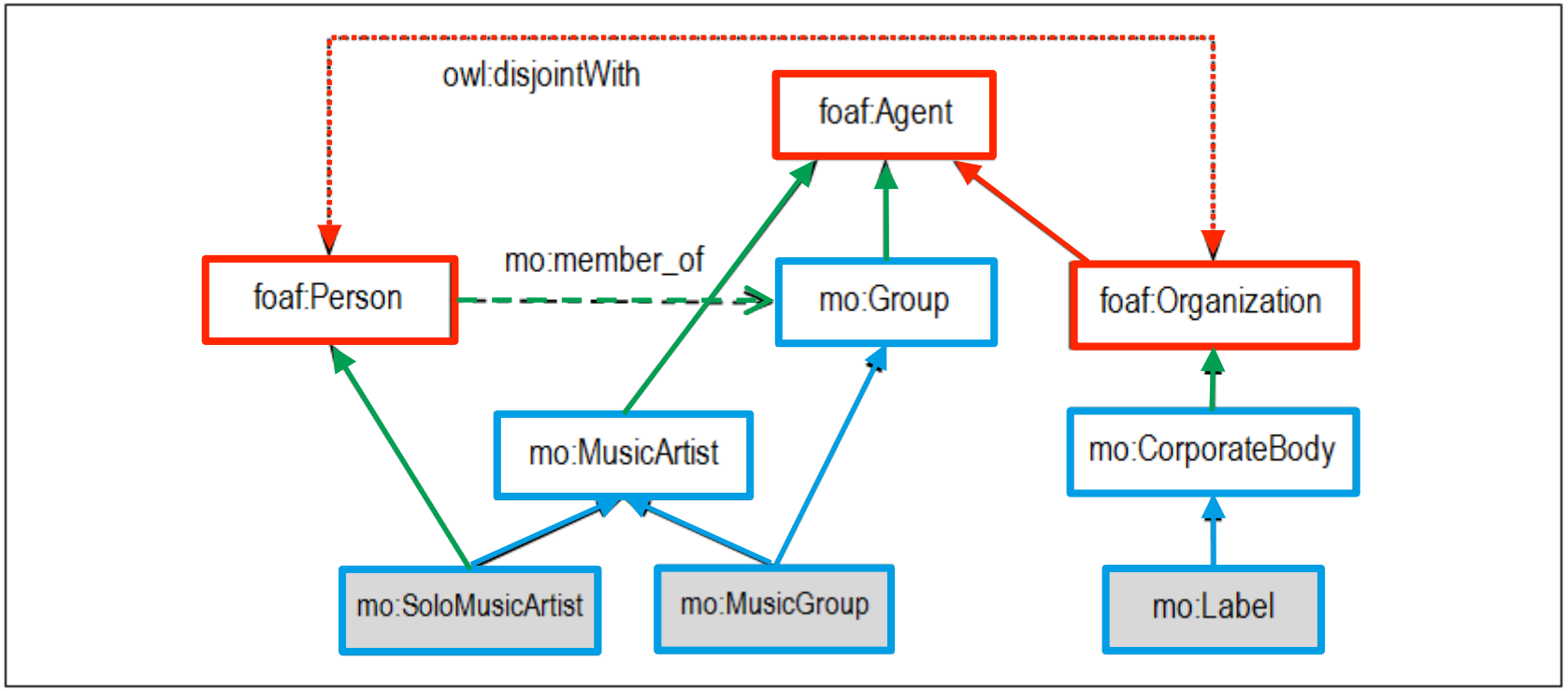
$O_1 = \pi[W](FOAF)$ where W = { foaf:person, foaf:Agent, foaf:Organization }



$O_3 = \sigma[F](\pi[W](FOAF) \cup O_2)$, where F = {mo:SoloMusicArtist ⊑ foaf:person,
mo:MusicArtist ⊑ foaf:Agent, mo:Group ⊑ foaf:Agent, mo:CorporateBody ⊑ foaf:Organization,
∃mo:member_of ⊑ foaf:person, ∃ mo:member_of⁻ ⊑ mo:Group }

$O_2 = (V_2, S_2)$, where V_2 = { mo:Group, mo:MusicArtist, mo:CorporateBody, mo:SoloMusicArtist,
mo:MusicGroup, mo:Label }
S_2 = {mo:SoloMusicArtist ⊑ mo:MusicArtist, mo:MusicGroup ⊑ mo:MusicArtist,… }

$O_1 = \pi[W](FOAF)$ where W = { foaf:person, foaf:Agent, foaf:Organization }



$O_3 = \sigma[F](\pi[W](FOAF) \cup O_2)$

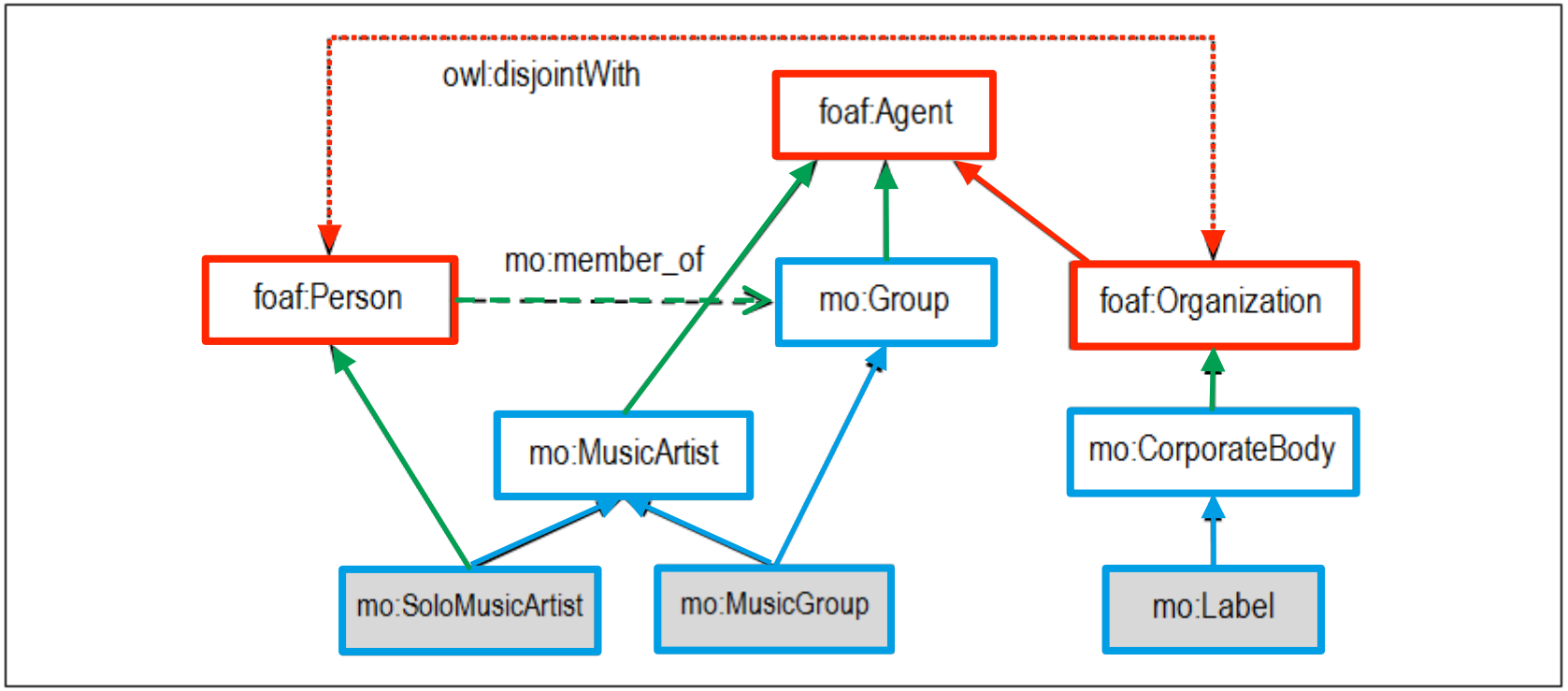
W = { foaf:person, foaf:Agent, foaf:Organization }

$O_2 = (V_2, S_2)$, where

V_2 = { mo:Group, mo:MusicArtist, mo:CorporateBody, mo:SoloMusicArtist, mo:MusicGroup, mo:Label }

S_2 = {mo:SoloMusicArtist ⊑ mo:MusicArtist, mo:MusicGroup ⊑ mo:MusicArtist,… }

F = {mo:SoloMusicArtist ⊑ foaf:person, mo:MusicArtist ⊑ foaf:Agent, mo:Group ⊑ foaf:Agent, mo:CorporateBody ⊑ foaf:Organization, ∃mo:member_of ⊑ foaf:person, ∃ mo:member_of⁻ ⊑ mo:Group }



Operations over Ontologies

- **Computing the operations**
  - Union
    - (must check if the new set of constraints implies $e \sqsubseteq \bot$, for some e)
  - Projection, Intersection
    - implemented as variants of IMPLIES
    - use the transitive closure of the constraint graphs
  - Difference
    - (To be further investigated)

Operations over Ontologies

- **Projection**

Input: an ontology $O_1 = (V_1, S_1)$ and a vocabulary $W \subseteq V_1$

Output: an ontology $O_P = (W, S_P)$, where
S_P is a set of constraints tautologically equivalent to
the set of constraints in $\tau[S_1]$ that use only symbols in W

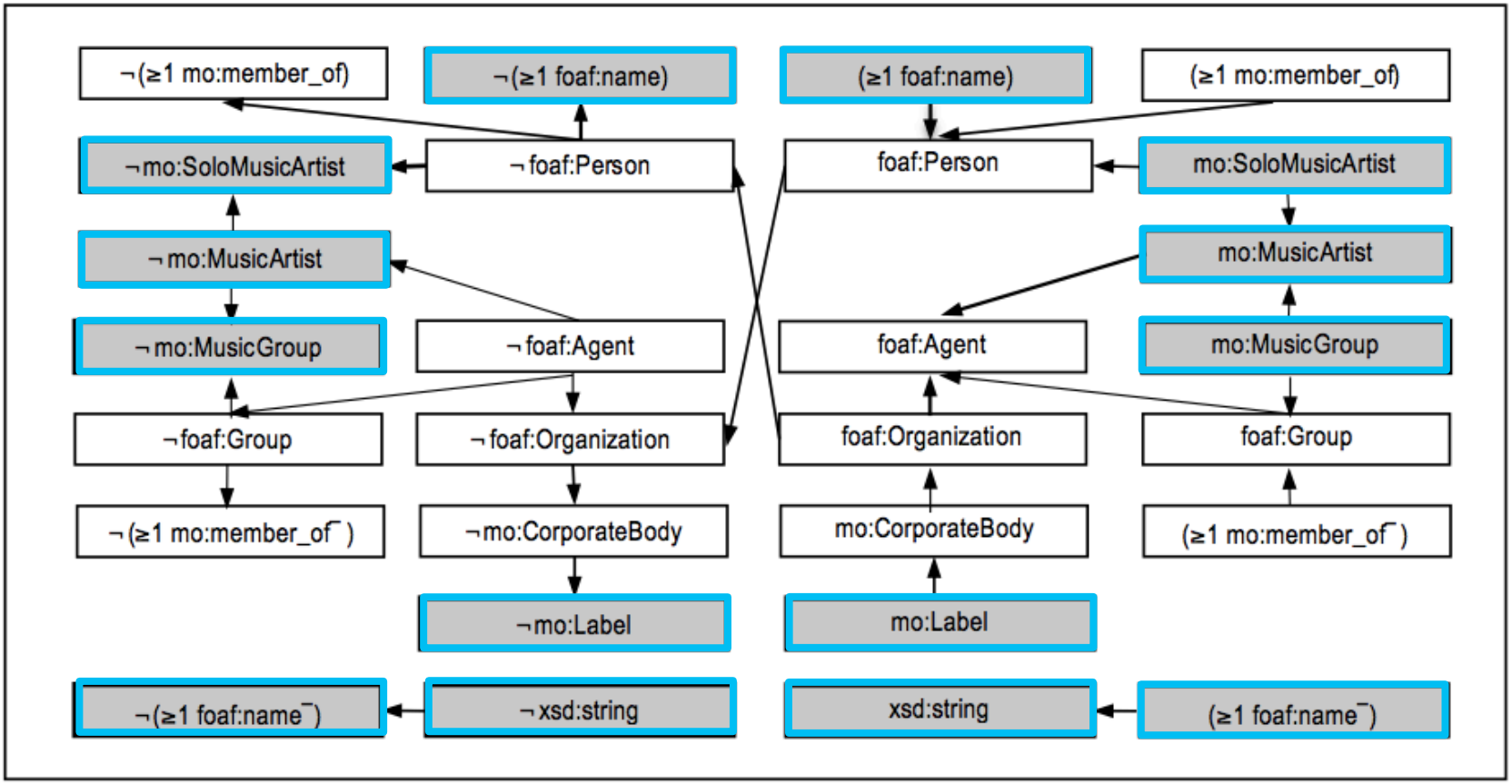
1. Generate $G(S_1)$, the constraint graph of S_1 .
2. Compute the transitive closure $G^*(S_1)$ of $G(S_1)$.
3. Mark all nodes of $G^*(S_1)$ that are labeled with expressions that use only symbols in W .
4. Generate a set of constraints S_P that correspond to:
   a) Arcs of $G^*(S_1)$ connecting marked nodes; and
   b) Expressions (in W) that label the same marked node .

Operations over Ontologies

Constraint	Informal specification
$(\geq 1\ \text{foaf:name}) \sqsubseteq \text{foaf:Person}$ $(\geq 1\ \text{foaf:name}^{-}) \sqsubseteq \text{xsd:string}$ $(\geq 1\ \text{mo:member\_of}) \sqsubseteq \text{foaf:Person}$ $(\geq 1\ \text{mo:member\_of}^{-}) \sqsubseteq \text{foaf:Group}$	The domain of foaf:name is foaf:Person The range of foaf:name is xsd:string The domain of mo:member_of is foaf:Person The range of mo:member_of is foaf:Group
$\text{mo:MusicArtist} \sqsubseteq \text{foaf:Agent}$ $\text{foaf:Group} \sqsubseteq \text{foaf:Agent}$ $\text{foaf:Organization} \sqsubseteq \text{foaf:Agent}$ $\text{mo:SoloMusicArtist} \sqsubseteq \text{foaf:Person}$ $\text{mo:SoloMusicArtist} \sqsubseteq \text{mo:MusicArtist}$ $\text{mo:MusicGroup} \sqsubseteq \text{mo:MusicArtist}$ $\text{mo:MusicGroup} \sqsubseteq \text{foaf:Group}$ $\text{mo:CorporateBody} \sqsubseteq \text{foaf:Organization}$ $\text{mo:Label} \sqsubseteq \text{mo:CorporateBody}$	mo:MusicArtist is a subset of foaf:Agent foaf:Group is a subset of foaf:Agent foaf:Organization is a subset of foaf:Agent mo:SoloMusicArtist is a subset of foaf:Person mo:SoloMusicArtist is a subset of mo:MusicArtist mo:MusicGroup is a subset of mo:MusicArtist mo:MusicGroup is a subset of foaf:Group mo:CorporateBody is a subset of foaf:Organization mo:Label is a subset of mo:CorporateBody
$\text{foaf:Person} \sqsubseteq \neg\text{foaf:Organization}$	foaf:Person and foaf:Organization are disjoint

Implementation of the Operations

$A \cap B = \emptyset$
iff
$A \subseteq \neg B$
iff
$B \subseteq \neg A$



Operations over Ontologies

Constraint	Informal specification
mo:Label ⊑ ¬(≥1 foaf:name)	$G(\Sigma_{APO})$ has a path from the node labeled with mo:Label to the node labeled with ¬(≥1 foaf:name) (which indicates that a label has no name)
(≥1 foaf:name⁻) ⊑ xsd:string	The range of foaf:name is xsd:string
mo:SoloMusicArtist ⊑ mo:MusicArtist mo:MusicGroup ⊑ mo:MusicArtist	mo:SoloMusicArtist is a subset of mo:MusicArtist mo:MusicGroup is a subset of mo:MusicArtist
mo:SoloMusicArtist ⊑ ¬mo:Label	mo:SoloMusicArtist and mo:Label are disjoint

Operations over Ontologies

- **Optimization**
  - Problem:
    - The transitive closure $G^*(S_1)$ contains redundancies !
  - Solution:
    - The problem is equivalent to finding the minimum equivalent graph (MEG) of a graph *G*, defined as the graph *G'* with the minimum set of edges such that the transitive closure of *G* and *G'* are equal
    - Finding the minimum equivalent graph has a polynomial solution for acyclic graphs and is NP-hard for strongly connected graphs

Topics

- Introduction
- A Formal Framework
- Operations over Ontologies
- **Implementation of the Operations**
- Conclusions

Implementation of the Operations

- **OntologyManagerTab**
  - Implemented as a Protégé Plugin
  - Works with (restricted) OWL ontologies
  - Offers a friendly user interface

- **Examples:**
  - A projection of the FOAF Ontology
  - The intersection of FOAF and the Music Ontology

OntologyManagerTab

newspaper Protégé 3.4.8 (file:/Applications/Protege_3.4.8/examples/newspaper/newspaper.pprj, Protégé Files (.pont and .pins))

File Edit Project Code Window Collaboration Tools Help

Classes | Slots | Forms | Instances | Queries | Ontology Manager Tab

Operation: Projection | Run

Ontology 1	
Agent	
Project	~Document
Project	~Person
OnlineAccount	
Document	~Project
Document	~Organization
SpatialThing	
Concept	
OnlineEcommerceAcco...	OnlineAccount
OnlineGamingAccount	OnlineAccount
Person	SpatialThing
Person	Agent
Person	~Project
Person	~Organization
Organization	Agent
Organization	~Document
Organization	~Person
Group	Agent
PersonalProfileDocument	Document

Load Ontology Projection

Vocabulary	Choice
Agent	☐
Concept	☐
Document	☐
Group	☐
Image	☐
OnlineAccount	☐
OnlineChatAccount	☐
OnlineEcommerceAcco...	☐
OnlineGamingAccount	☐
Organization	☐
Person	☐
PersonalProfileDocument	☐
Project	☐
SpatialThing	☐
account	☐
accountServiceHomepage	☐
aimChatID	☐
based_near	☐
currentProject	☐

Inclusion Constraints	

Save Results

Log:

```
Welcome to the Ontology Manager Tab!
Developed by Romulo de Carvalho Magalhaes

Loading: /Users/romulo/Ontologias/foafFull.rdf
Ontology successfully loaded as Ontology 1
```

OntologyManagerTab

newspaper Protégé 3.4.8 (file:/Applications/Protege_3.4.8/examples/newspaper/newspaper.pprj, Protégé Files (.pont and .pins))

File Edit Project Code Window Collaboration Tools Help

Classes | Slots | Forms | Instances | Queries | Ontology Manager Tab

Operation: Intersection — Run

Ontology 1	
Agent	
Project	~Document
Project	~Person
OnlineAccount	
Document	~Project
Document	~Organization
SpatialThing	
Concept	
OnlineEcommerceAcco...	OnlineAccount
OnlineGamingAccount	OnlineAccount
Person	SpatialThing
Person	Agent
Person	~Project
Person	~Organization
Organization	Agent
Organization	~Document
Organization	~Person
Group	Agent
PersonalProfileDocument	Document

Load Ontology 1

Ontology 2	
Item	
Agent	
Stream	Medium
ReleaseType	
ReleaseStatus	
MagneticTape	Medium
SpatialThing	
Record	MusicalManifestation
MD	Medium
Expression	
DVDA	Medium
RecordingSession	Event
Vinyl	Medium
Arrangement	Event
AnalogSignal	Signal
AnalogSignal	~DigitalSignal
PublishedLibretto	MusicalManifestation
SoloMusicArtist	Person
SoloMusicArtist	MusicArtist

Load Ontology 2

Inclusion Constraints	
Agent	
Document	
SpatialThing	
Person	
Organization	
Group	
Image	
depiction	
member	
isPrimaryTopicOf	
~Document	~(>=1 homepage Document)
(>=1 homepage Document)	Document
(>=1 member Agent)	Agent
~Person	
~SpatialThing	
~Agent	~(>=1 member Agent)
~Group	
~Image	
~isPrimaryTopicOf	

Save Results

Log:

```
Welcome to the Ontology Manager Tab!
Developed by Romulo de Carvalho Magalhaes

Loading: /Users/romulo/Ontologias/foafFull.rdf
Ontology successfully loaded as Ontology 1
Loading: /Users/romulo/Ontologias/musicontology.owl
Ontology successfully loaded as Ontology 2
Running Intersection over /Users/romulo/Ontologias/foafFull.rdf and /Users/romulo/Ontologias/musicontology.owl
Intersection done!
```

Topics

- Introduction
- A Formal Framework
- Operations over Ontologies
- Implementation of the Operations
- **Conclusions**

Conclusions

- **Question…**
  - How to design databases to be published on the Web (so that software agents can understand the data)?

Conclusions

- **Suggested answer…**
  - An external schema should be
    a combination of fragments
    of one or more domain ontologies

    so that matching
    the application schema
    with the external schema
    should become a *non-problem*
    (or it will remain an intractable problem)

Bruegel, Pieter. The Tower of Babel. c 1563

Oil on panel. 60 × 74.5 cm

Museum Boijmans Van Beuningen, Rotterdam, Netherlands

Conclusions

- **New question…**
  - What is the *meaning* of "a combination of fragments of one or more domain ontologies"?

    (So that matching the application schema with the external schema becomes a ***non-problem***)

Conclusions

- **Suggested answer…**
  - Operations over ontologies that create new ontologies, including their constraints, out of other ontologies

Conclusions

- **Implementation of the operations**
  - Based on a structural proof procedure, i.e., a procedure that explores the structure of sets of constraints
  - Works for lightweight constraints

Conclusions

- **Complexity**
  - "Easy",
    when the domain ontology contains
    class hierarchies and cardinality constraints (and keys)
  - "Difficult, but still possible",
    when the domain ontology contains
    class hierarchies, property hierarchies and
    cardinality constraints (and well-behaved keys)
  - "Difficult, but still possible – under certain assumptions",
    when, in addition,
    property domains and ranges are defined by unions

Acknowledgements

- **INCT for Web Science**
  - UFC: *Vânia Vidal, José Antonio Macedo, Ângela Pinheiro*
  - PUC-Rio: *Antônio Furtado, Rômulo Magalhães*

Thank You!