Reengineering and Refactoring

Eva Blomqvist
eva.blomqvist@liu.se
Department of Computer and Information Science (IDA)
Linköpings universitet
Sweden

Slides partly by Aldo Gangemi, STLab, ISTC-CNR, Italy

Outline

- Introduction and motivation
- From any data source to the Semantic Web, e.g DB etc.
- Semion method and tool
- Writing Semion rules

Why do I need transformations?

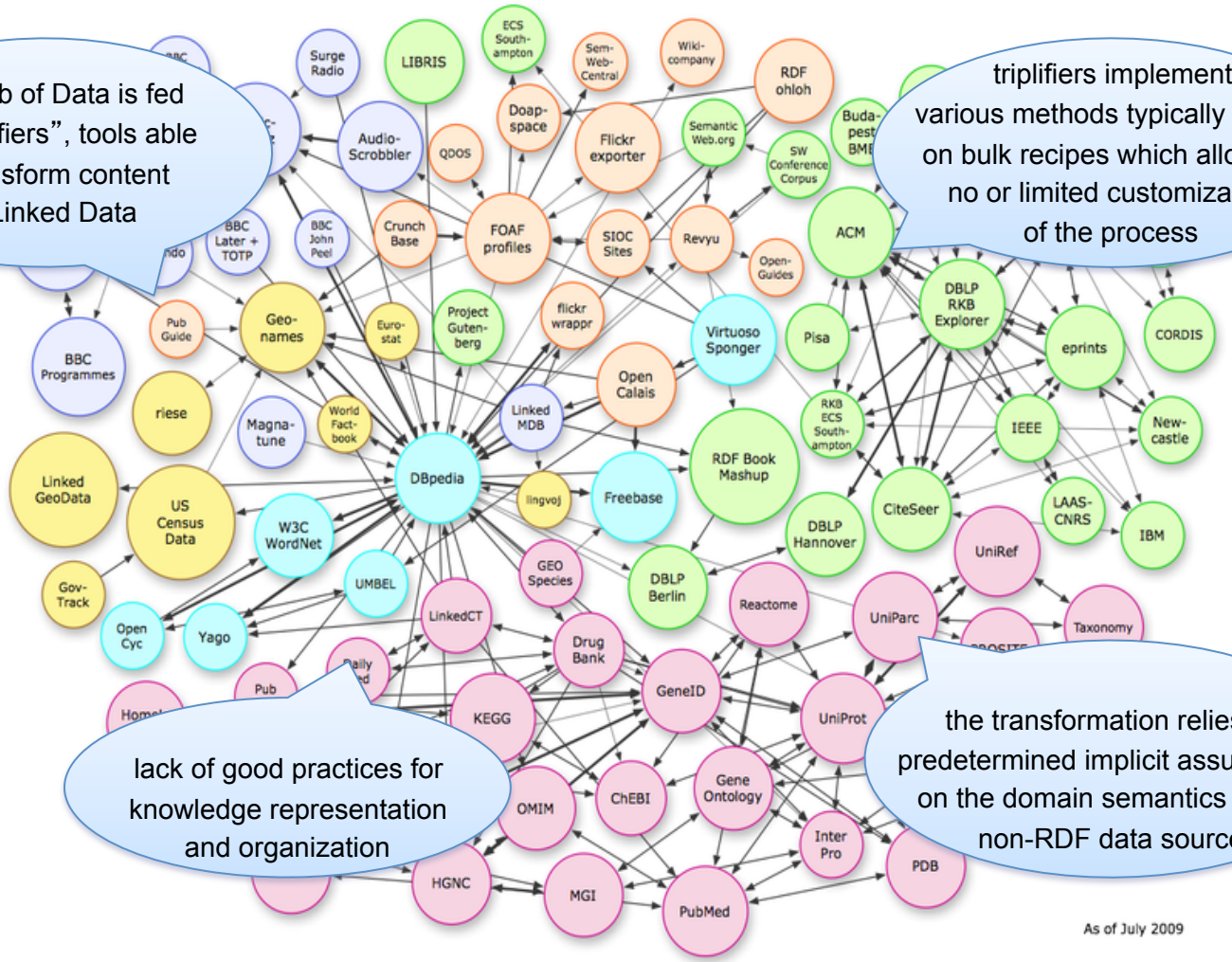
- A scenario
  - my system fetches knowledge from different sources in LOD
  - each of these sources uses its own ontology/vocabulary
- Another scenario
  - I have legacy data in a DB or in a custom XMI format
  - This data should be integrated with RDF data

How to arrive at a homogeneous representation of knowledge expressed with heterogeneous schemas/vocabularies?

Motivations

the Web of Data is fed by “triplifiers”, tools able to transform content to Linked Data

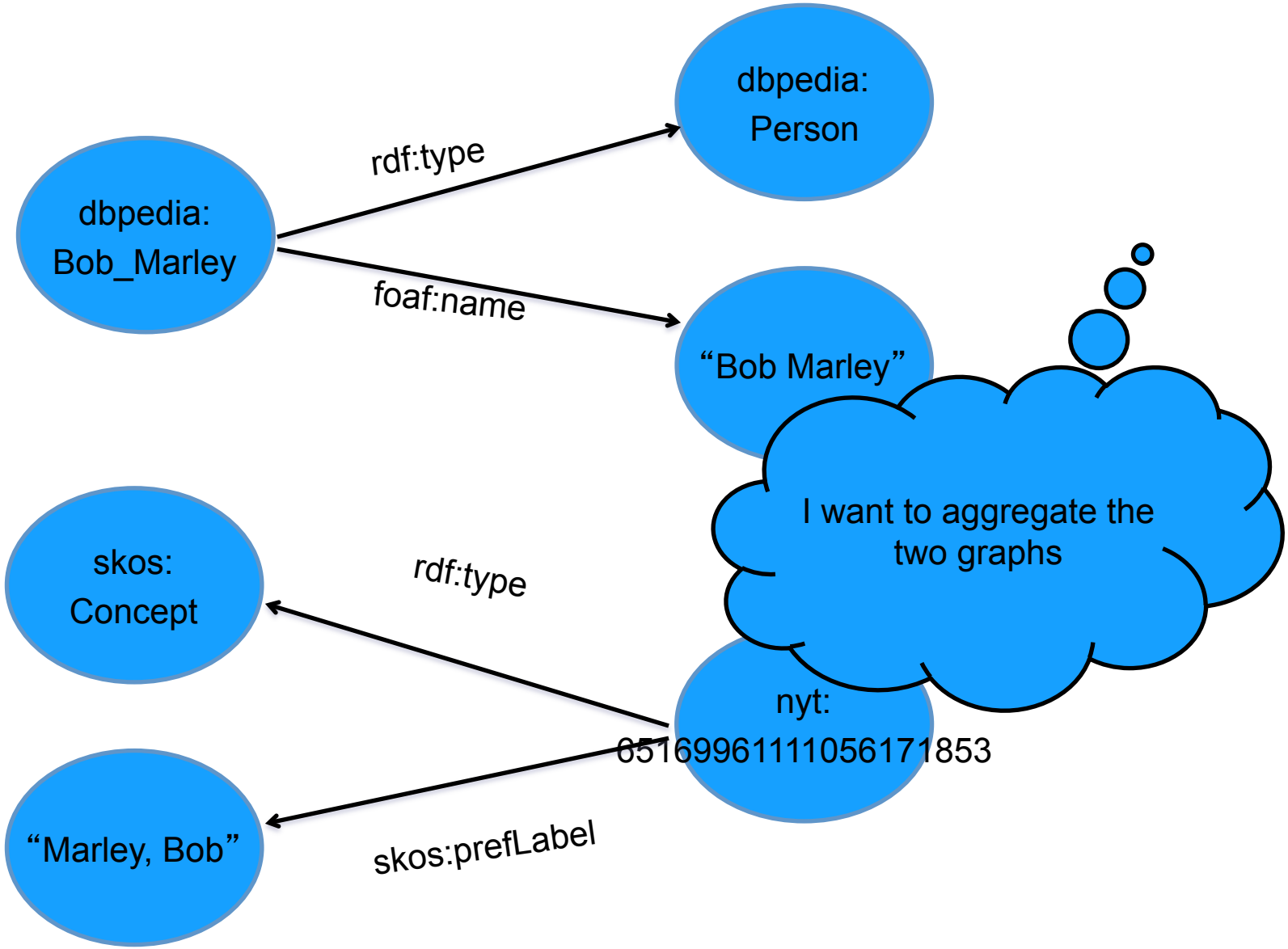
triplifiers implement various methods typically based on bulk recipes which allow for no or limited customization of the process



lack of good practices for knowledge representation and organization

the transformation relies on predetermined implicit assumptions on the domain semantics of the non-RDF data source

An Example


dbpedia:
Person
rdf:type
dbpedia:
Bob_Marley
foaf:name
“Bob Marley”
I want to aggregate the
two graphs
skos:
Concept
rdf:type
nyt:
65169961111056171853
“Marley, Bob”
skos:prefLabel

A DB stores data and answers queries

- Aldo is 48
- Aldo works in Rome
- Aldo does research on Semantic Web

nome	data di nascita	luogo_di_lavoro	temi
Aldo	08-16-962	Roma	Semantic Web
Alfio	….	….	….

Persons

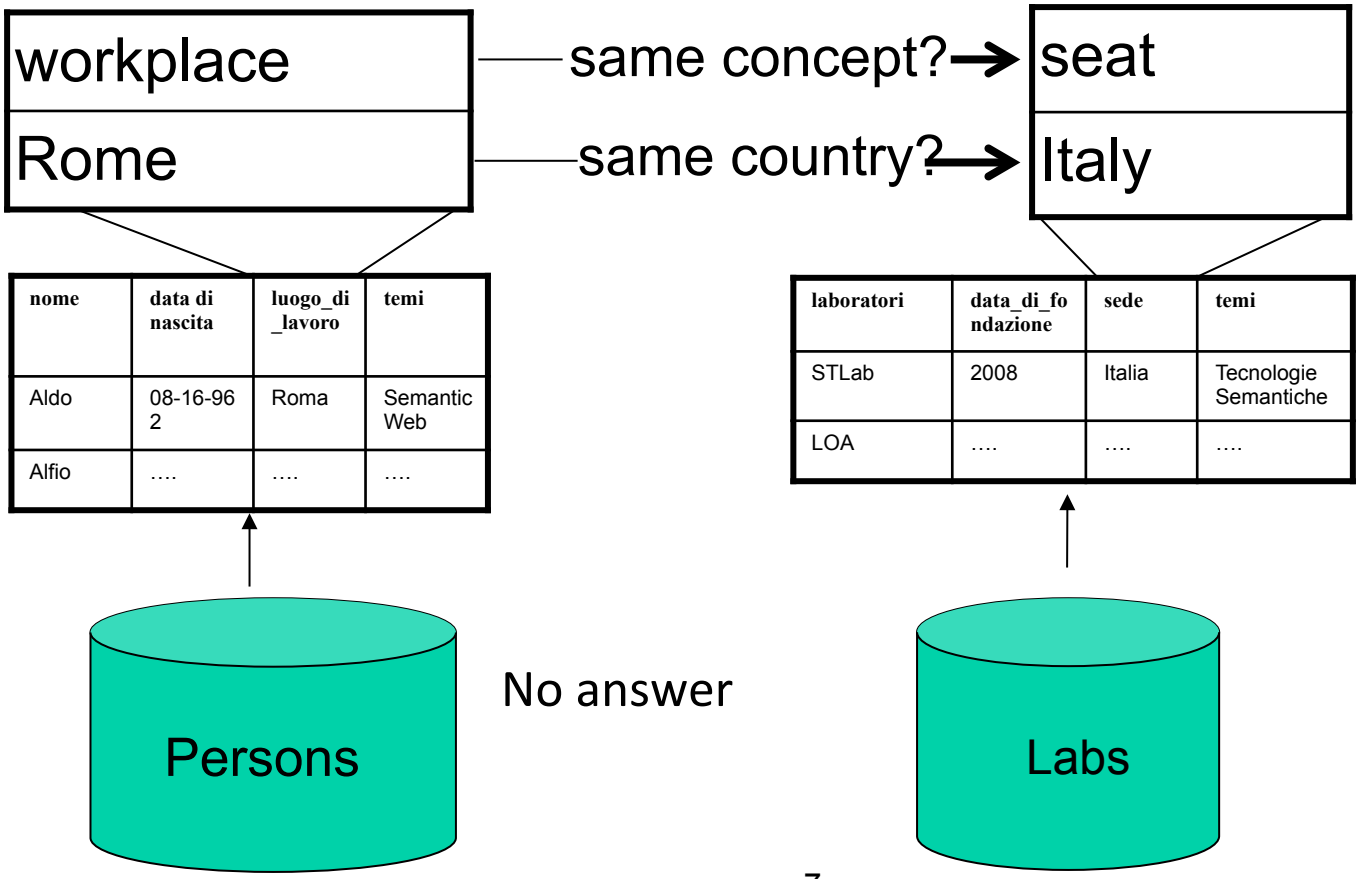
- STLab was founded in 2008
- STLab is in Italy
- STLab does research on semantic technologies

laboratori	data_di_fondazione	sede	temi
STLab	2008	Italia	Tecnologie Semantiche
LOA	….	….	….

Labs

Complex queries?

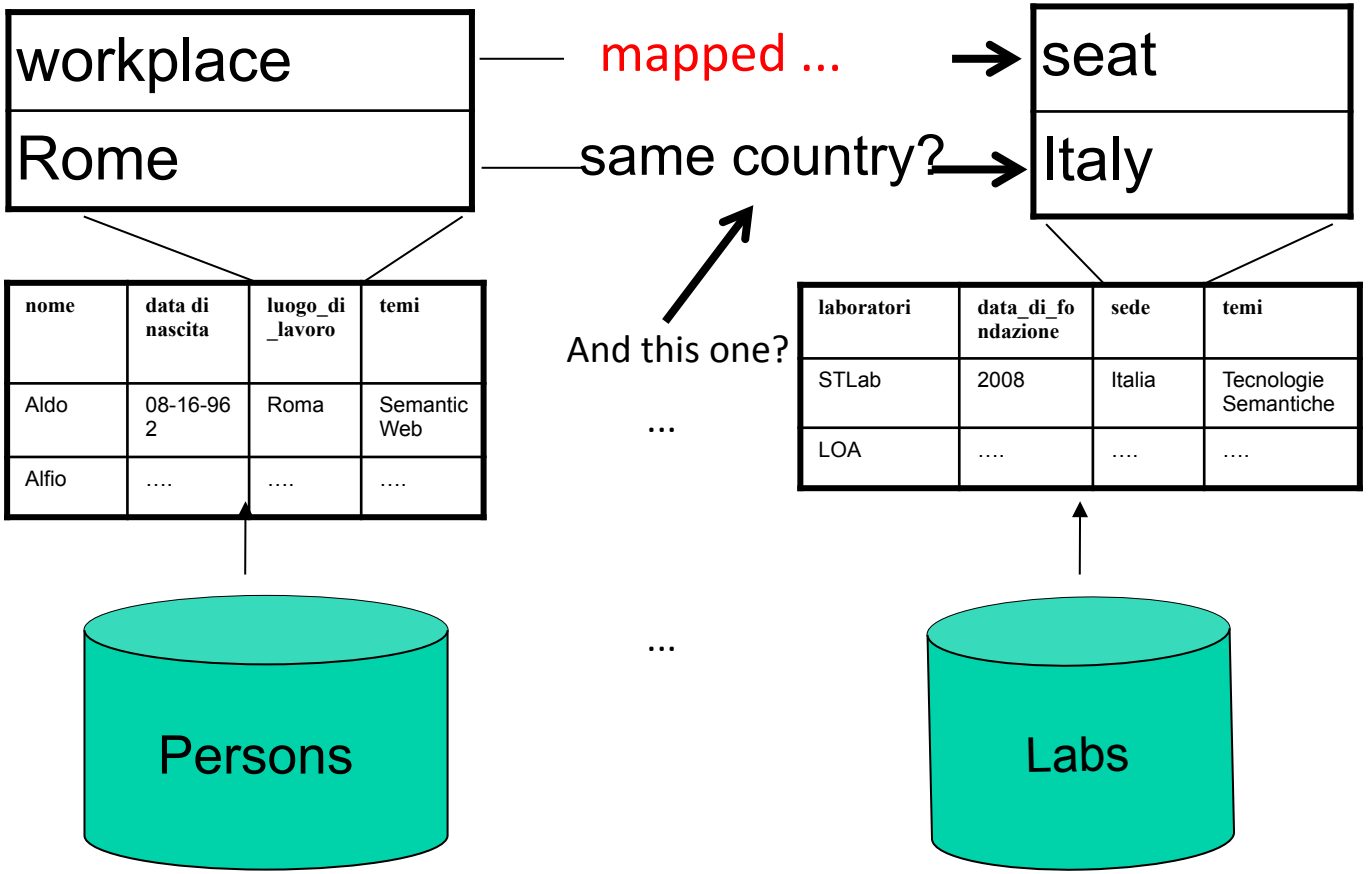
- *Who is interested in "Semantic Web" and is working in the same country as STLab is located?*



nome	data di nascita	luogo_di_lavoro	temi
Aldo	08-16-962	Roma	Semantic Web
Alfio	….	….	….

laboratori	data_di_fondazione	sede	temi
STLab	2008	Italia	Tecnologie Semantiche
LOA	….	….	….

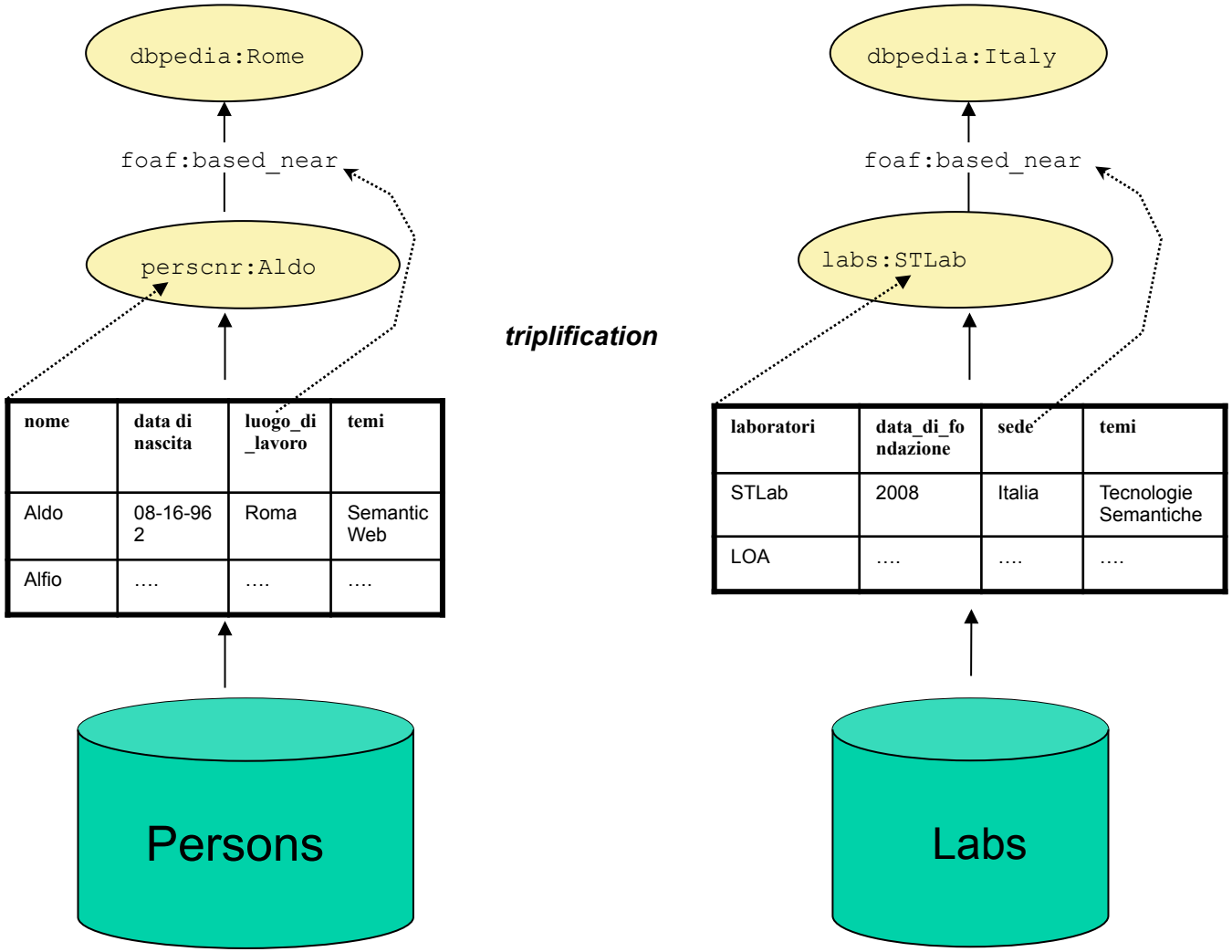
Is mapping enough?



nome	data di nascita	luogo_di _lavoro	temi
Aldo	08-16-962	Roma	Semantic Web
Alfio	....	....	....

laboratori	data_di_fondazione	sede	temi
STLab	2008	Italia	Tecnologie Semantiche
LOA	....	....	....

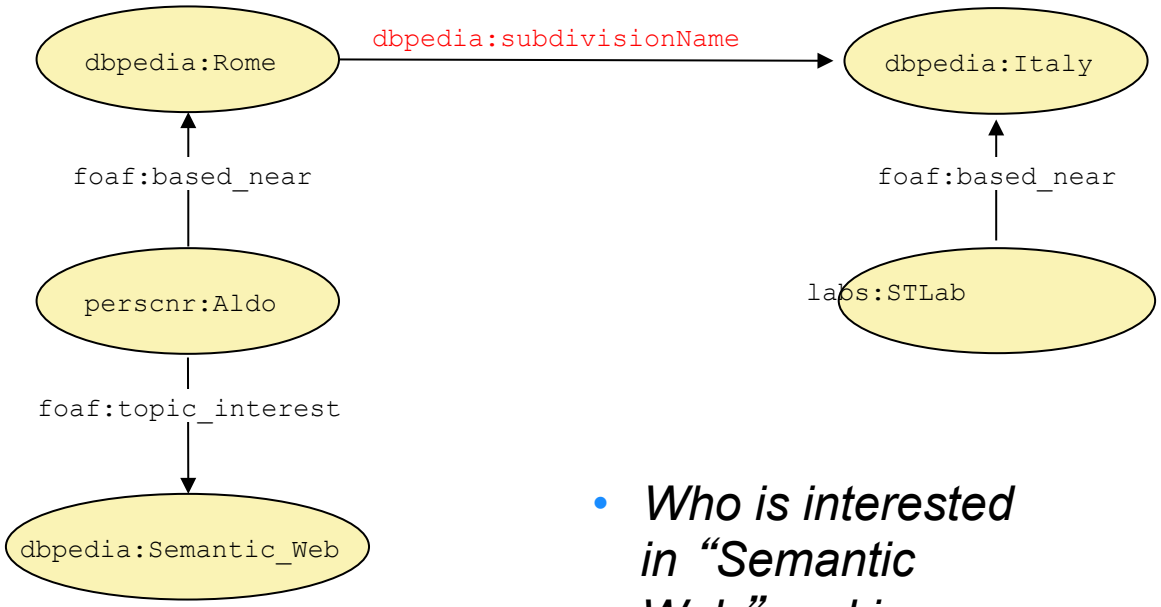
Publishing DB data as RDF on the Web



nome	data di nascita	luogo_di_lavoro	temi
Aldo	08-16-962	Roma	Semantic Web
Alfio	….	….	….

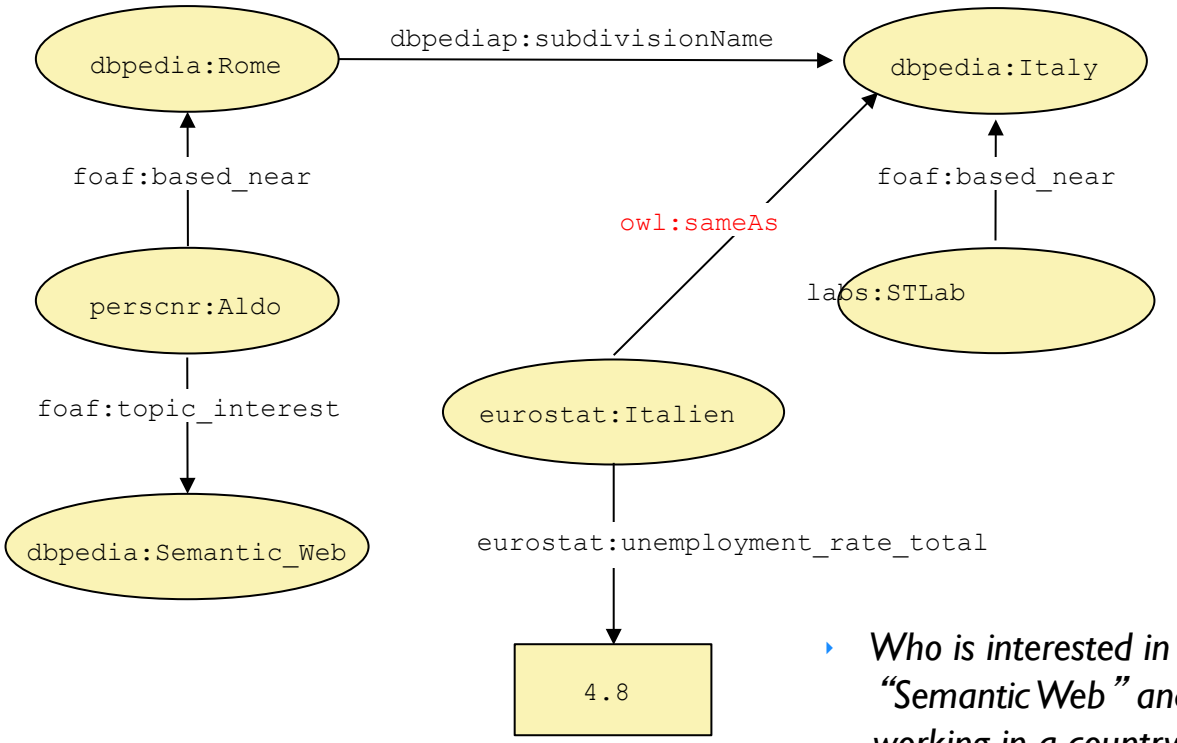
laboratori	data_di_fondazione	sede	temi
STLab	2008	Italia	Tecnologie Semantiche
LOA	….	….	….

Data linking and querying on the web of data



- *Who is interested in "Semantic Web" and is working in the same country as STLab is located?*

How to answer really tough queries?



- *Who is interested in "Semantic Web" and is working in a country where the unemployment rate is lower than 5%?*

Knowledge transformation issues

- Syntactic interoperability bottleneck (platform + data model)
  - e.g. *rdb, eav, xml, text, prolog, N3*
    - e.g. kos rdb with *adjacency list, path enumeration*
- Semantic interoperability bottleneck (logical + conceptual level)
  - e.g. rdb with: *lexical, statistical, formal* data
    - e.g. two different databases on the same topic
- Social, pragmatic interoperability bottleneck (privacy, sustainability, policy)
  - e.g. different requirements, organizational contexts, etc.

12

Dealing with web semantics: current state

- Much enthusiasm, a lot of nice, different ideas
- Much confusion and mutual misunderstanding between “scruffies” and “neats”
  - Pushing formal semantics beyond its limits (e.g. the “owl:sameAs” dispute)
  - Doing ad-hoc apps
  - Mixing up strings, classes, terms, concepts, topics, tags, etc.
  - Trivializing transformation from social to formal semantics (e.g. when translating a syntactic frame directly to an OWL construct)

Some techniques for semantic data reuse

- Virtual linked data
  - Automatic RDB schema conversion to RDFS
  - RDB data browsing and on-demand automatic conversion to RDF
  - Sample tools: Sparql endpoint+*D2R*
  - Dataset example: *IMDB*
  - +Time to usage –Flexibility
- Ontology-based access with ad-hoc queries
  - (DL-Lite) ontology to be designed separately
  - Ad-hoc SQL query on RDB, “embedded” in class spec
  - *On demand* ontology-based navigation
  - Sample tools: *Mastro+Quonto*
  - +Complexity –Flexibility –Time to usage
- Physical linked data with custom ontologies
  - Custom conversion of RDB/XML to one or more OWL ontologies
  - Custom conversion of data to RDF-OWL datasets that can be published and queried
  - Sample tools: Sparql endpoint+*Semion*
  - Sample datasets: *DBpedia, data.cnr.it*
  - +Flexibility ±Time to usage ±Complexity
- Key aspects
  - *Mapping specification*
  - *Consumable RDF data semantics*

14

Transformation patterns

- Types of transformation patterns
  1. Direct structural morphism

Broader	
Narrow concept	Paris
Broad concept	France

:Broader rdf:type dbs:Table
:Narrow_concept rdf:type dbs:Column
:France rdf:type dbs:Datum

  2. Semantic interpretation

Broader	
Narrow concept	Paris
Broad concept	France

:Broader rdf:type owl:ObjectProperty
:Narrow_concept rdf:type owl:Class
:Paris :broader :France

15

Transformation patterns (cont.)

3. Re-interpretation, e.g. through alignment patterns
   - *:Narrow_concept* → *skos:Concept*
     - also as mediated semantic interpretation
     - also as revised semantic interpretation

4. Production: new entities, vocabulary/string manipulation
   - *:Narrow_concept* → *:Concept*
   - *"%p"* → *:hyponym*
   - *"plant flora plant_life"* → *:Plant, :Flora, :PlantLife*

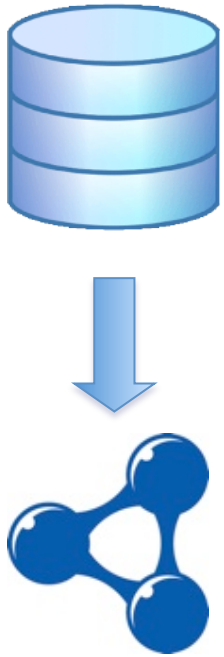
16

Semion - Method and Tool

17

A common recipe

- each table is a rdfs:Class
- each table record is an owl:Individual
- each table column is a rdf:Property



Example

Person	FirstName	LastName
	Aldo	Gangemi
	Valentina	Presutti
	Eva	Blomqvist

Class: Person

DatatypeProperty: firstName

DatatypeProperty: lastName

Individual: Person1
Type: Person
Facts: firstName “Aldo”
lastName “Gangemi”

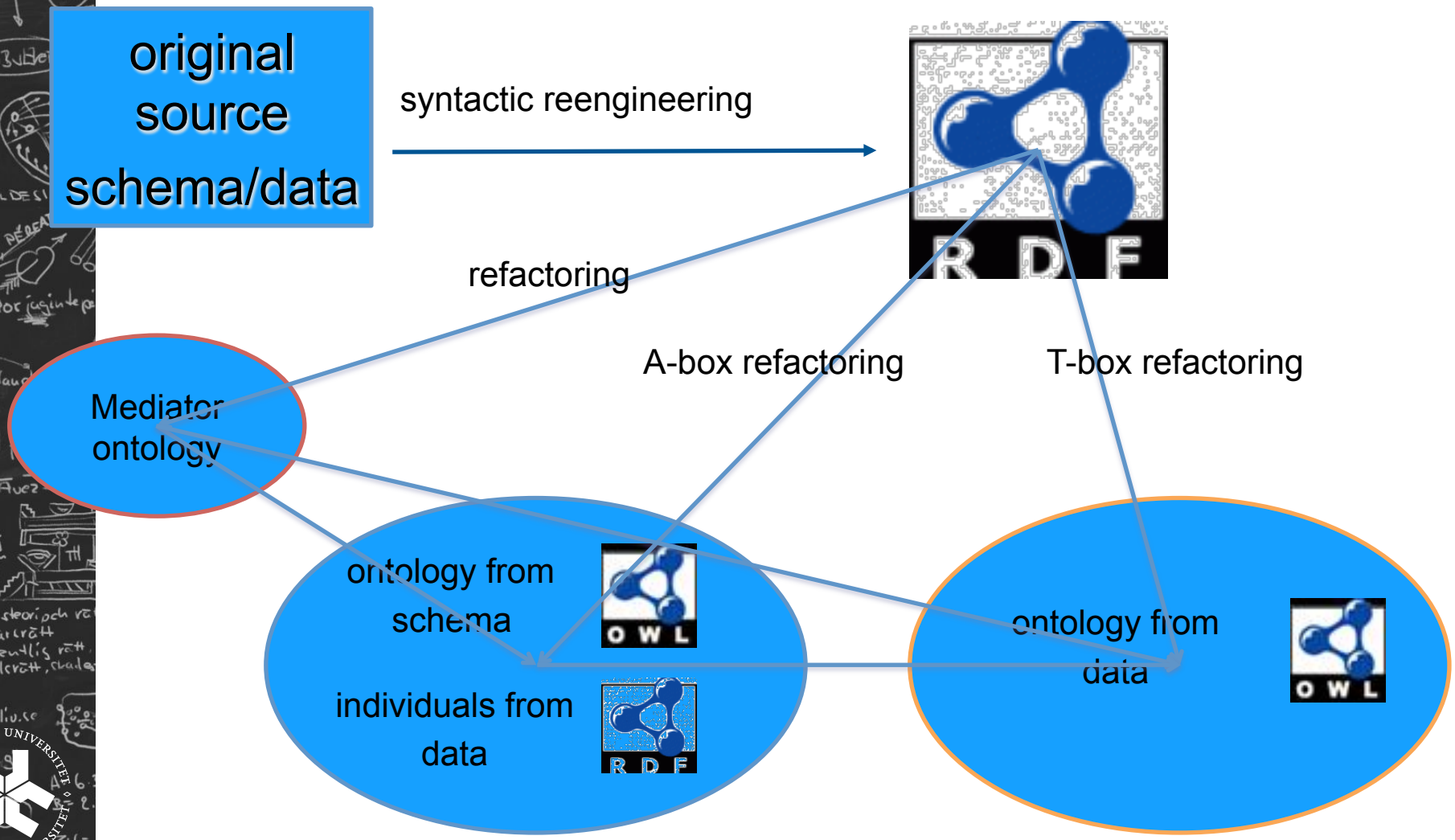
Individual: Person2
Type: Person
Facts: firstName “Valentina”
lastName “Presutti”

...

Implications

- Limited customization of the transformation process
- Difficulty in adopting good practices of knowledge reengineering and ontology design
- Limited exploitation of OWL expressivity for describing the domain

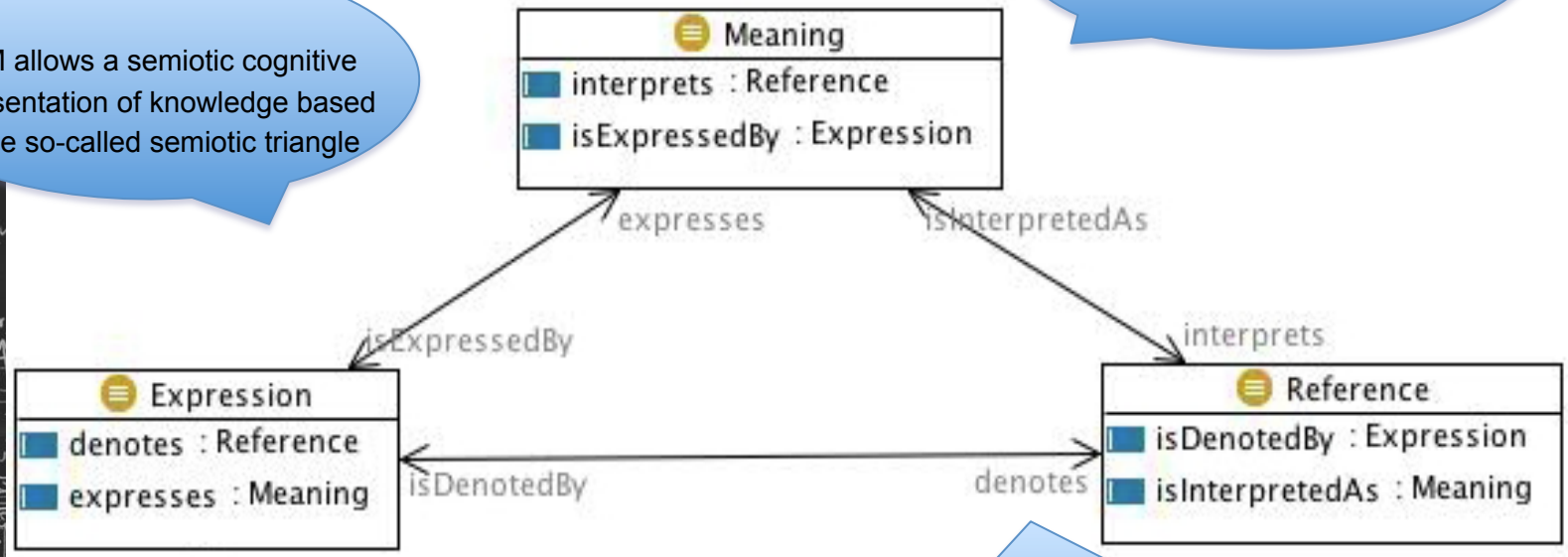
The Semion Approach



The Linguistic Meta-Model (LMM)

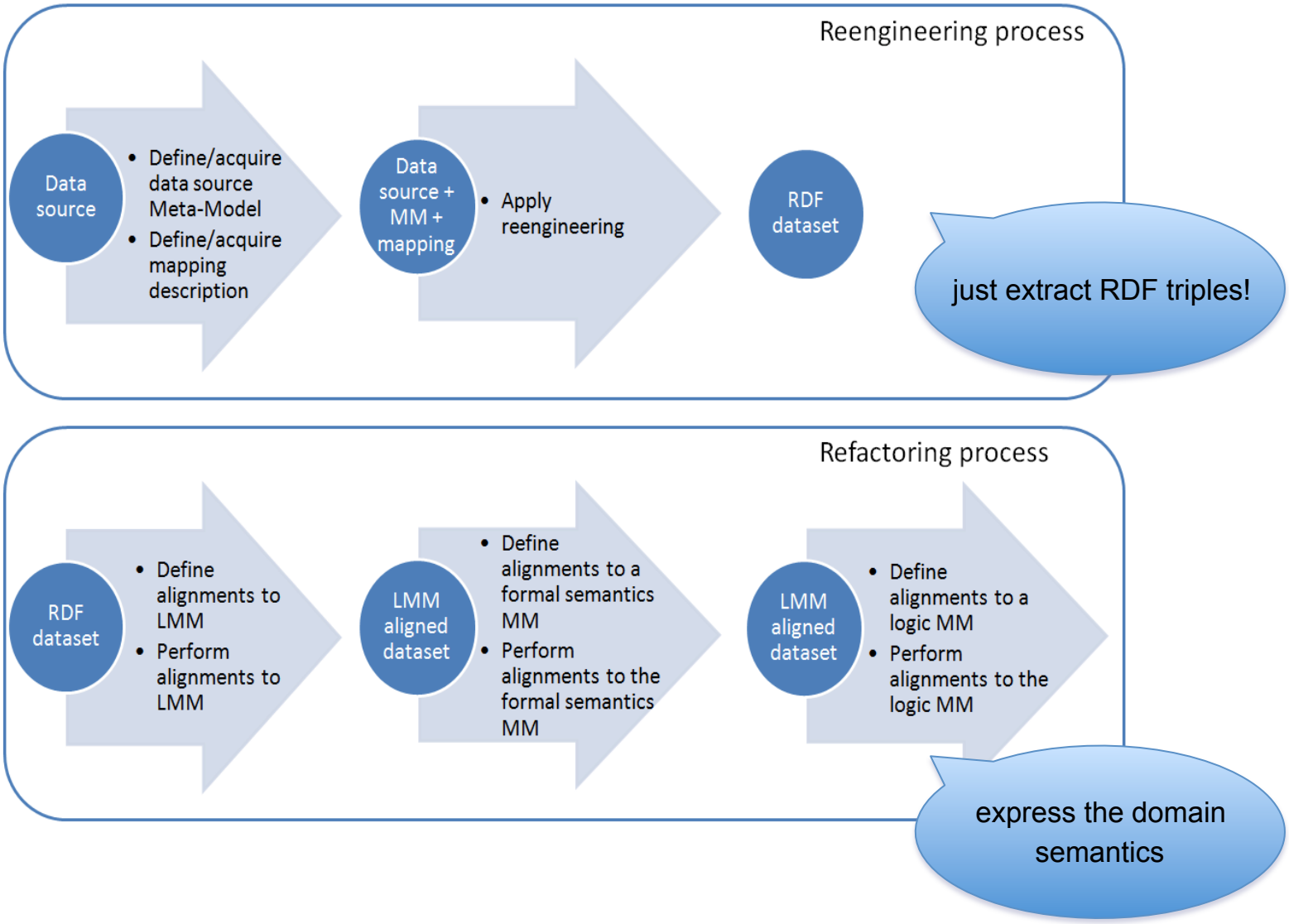
LMM plays the role of a mediator ontology

LMM allows a semiotic cognitive representation of knowledge based on the so-called semiotic triangle



most knowledge representation schemata can be aligned to the semiotic triangle

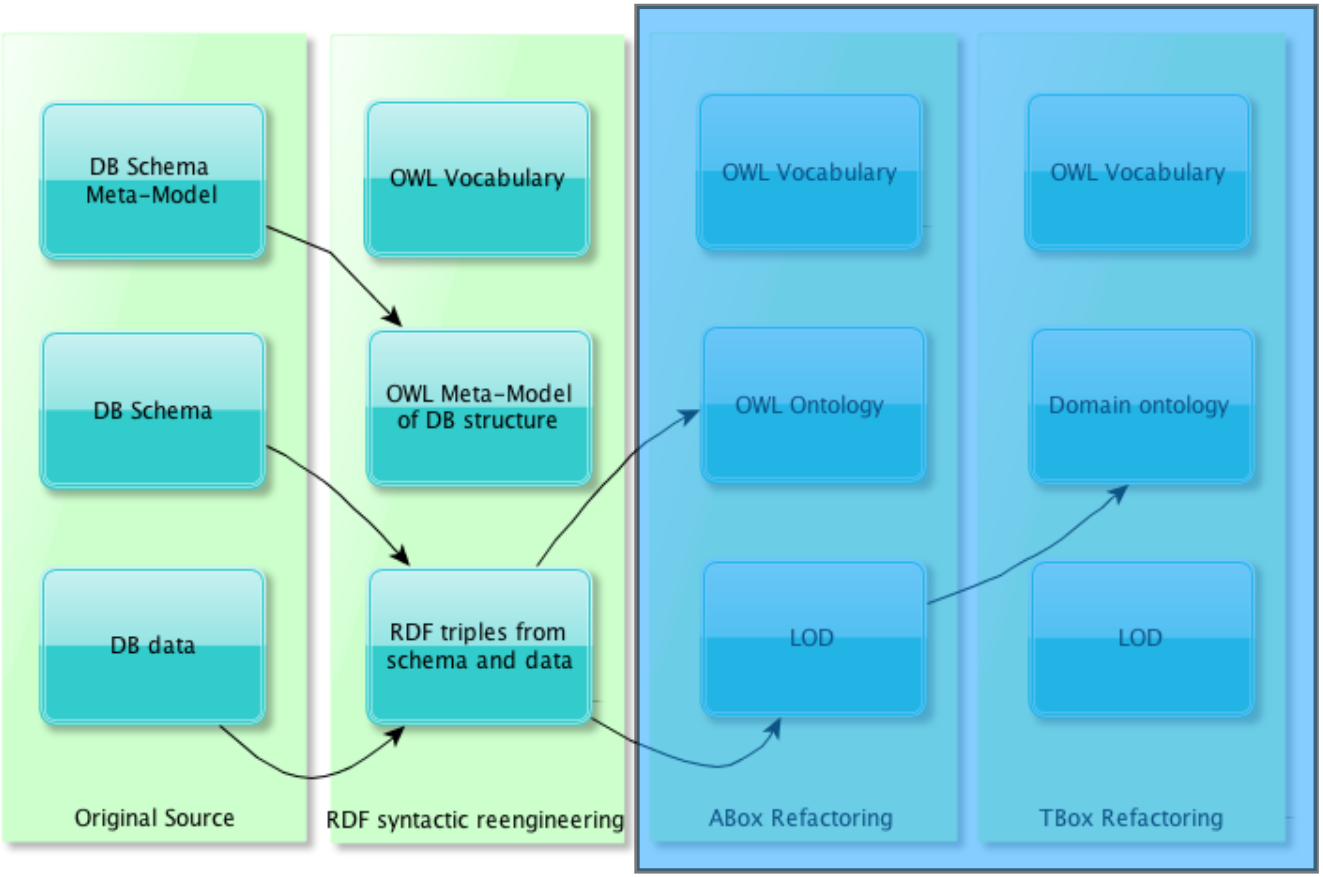
a mediator is an ontology that describes the entities used for expressing the in-formal semantics that we use for organizing our knowledge


Reengineering process
Data source
Define/acquire data source Meta-Model
Define/acquire mapping description
Data source + MM + mapping
Apply reengineering
RDF dataset
just extract RDF triples!
Refactoring process
RDF dataset
Define alignments to LMM
Perform alignments to LMM
LMM aligned dataset
Define alignments to a formal semantics MM
Perform alignments to the formal semantics MM
LMM aligned dataset
Define alignments to a logic MM
Perform alignments to the logic MM
express the domain semantics

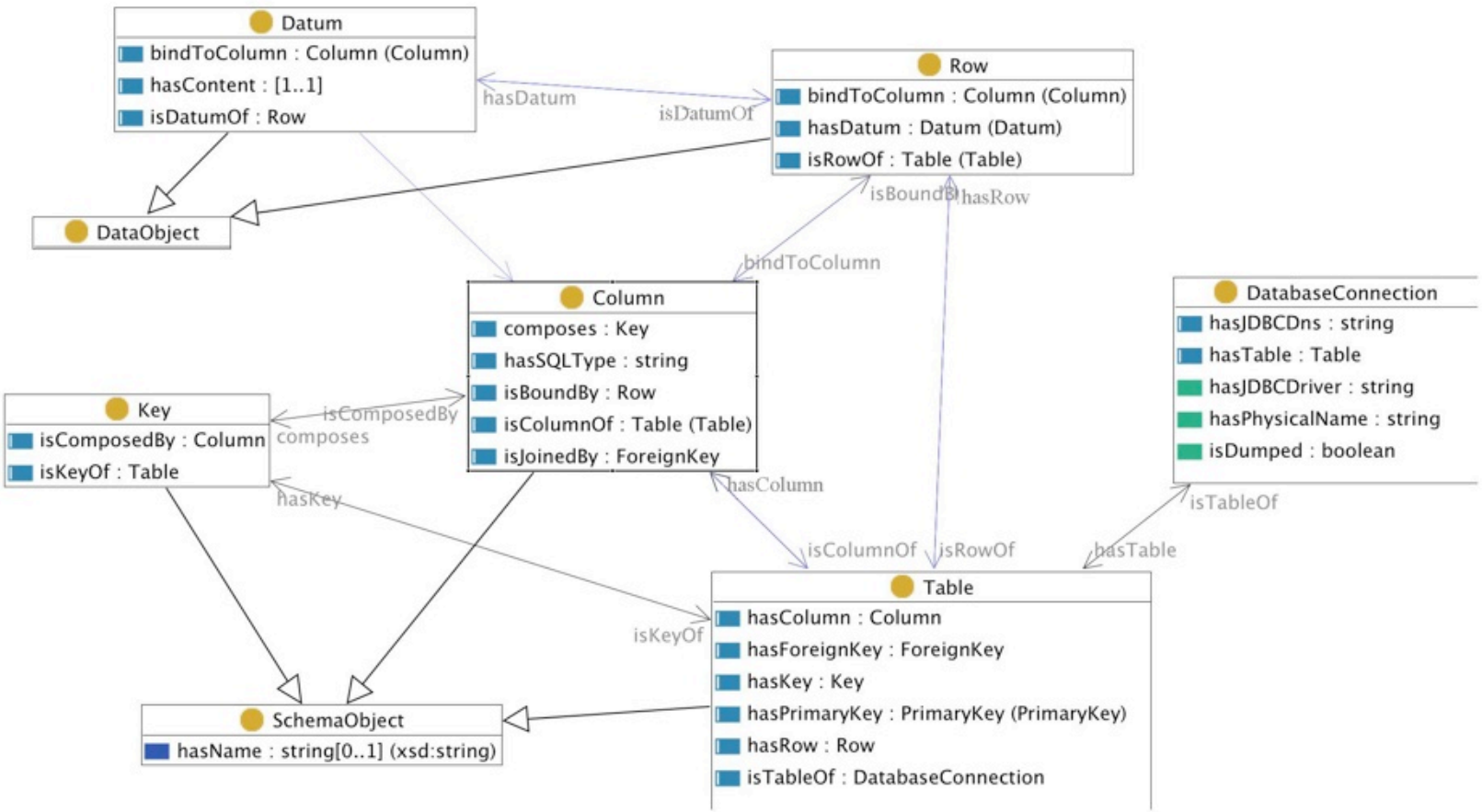
The Semion Reengineer

- It does not add any semantics, but just the RDF format
- Semion needs the meta-model of the structure of the source (and some code)
- Currently supports RDBMS and XML
- Supported sources can be extended by providing new reengineering services as an OSGi bundle (not available yet)

Basic idea


DB Schema Meta-Model
DB Schema
DB data
Original Source
OWL Vocabulary
OWL Meta-Model of DB structure
RDF triples from schema and data
RDF syntactic reengineering
OWL Vocabulary
OWL Ontology
LOD
ABox Refactoring
OWL Vocabulary
Domain ontology
LOD
TBox Refactoring

A meta-model for RDBs



Example of transformation of a DB

	FirstName	LastName
Person	Aldo	Gangemi
	Valentina	Presutti
	Eva	Blomqvist

Class: Table

ObjectProperty: hasRecord
Domain: Table
Range: Record
inverseOf: isRecordOf

ObjectProperty: isRecordOf
Domain: Table
Range: Record
inverseOf: hasRecord

Individual: Person
Type: Table
Facts: hasRecord AldoGangemi

Individual: AldoGangemi
Types: Record
Facts:
hasDatum AldoGangemiFirstName
hasDatum AldoGangemiLastName
isRecordOf Person

Reengineering Rules

- Table -> individual of dbs:Table
- Column -> individual of dbs:Column
- Record -> individual of dbs:Record
- Field -> individual of dbs:Datum

Primary Keys

- primary keys are used for URI generation

Person	FirstName	LastName	id (PK)
	Aldo	Gangemi	stlab.istc-cnr1

Individual: Person_stlab.istc-cnr1

Type: Record

Foreign Keys

- Foreign keys identifies relations between tables and are mapped to relations between individuals

Person	FirstName	LastName	id (PK)	affiliation
	Aldo	Gangemi	stlab.istc-cnr1	istc-cnr

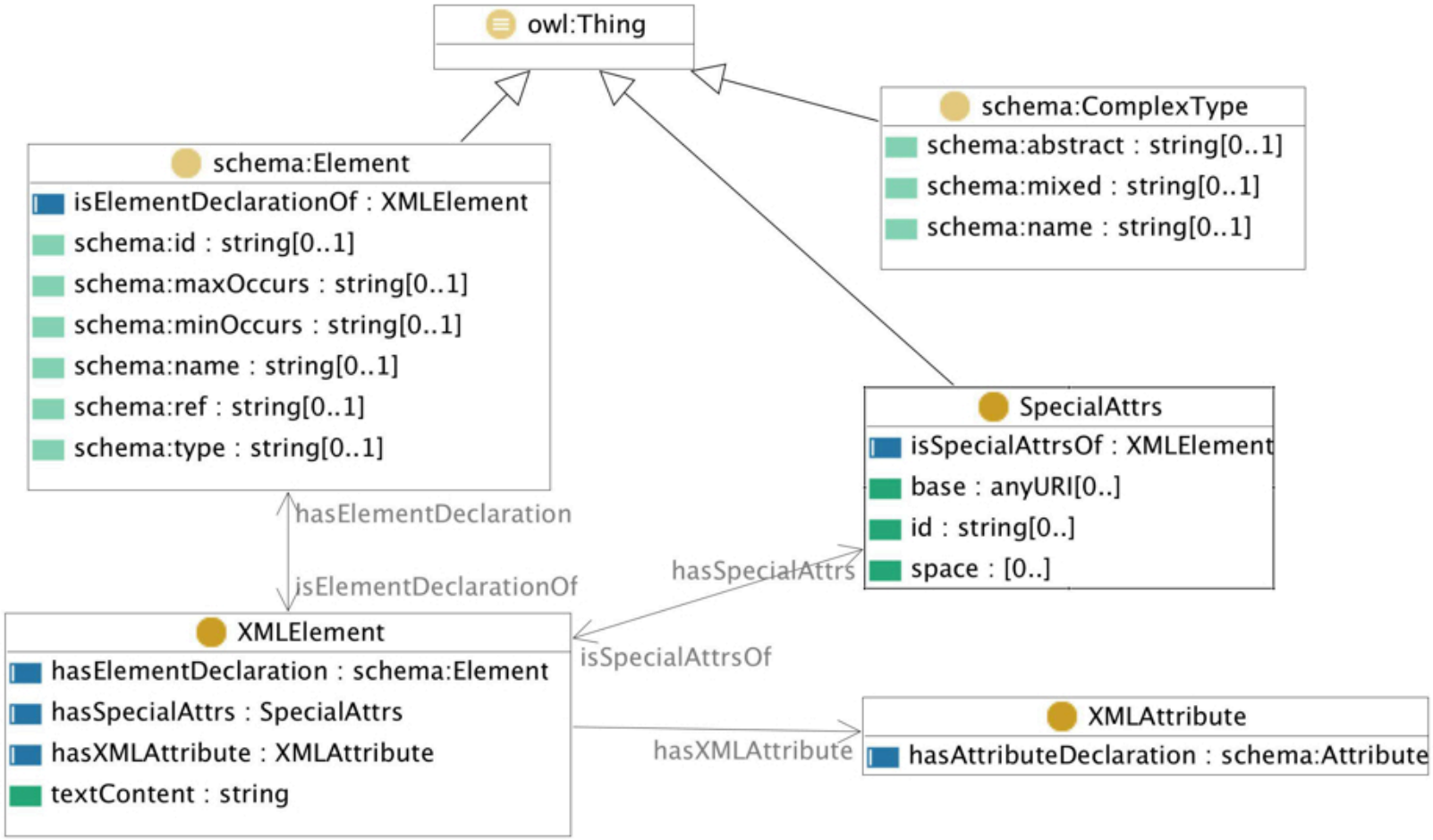
Institute	id (PFK)	name
	istc-cnr	Istituto di Scienze e Tecnologie della Cognizione - CNR

Individual: Person_stlab.istc-cnr1
Type: Record
Facts: hasDatum Person_stlab.istc-cnr1_affiliation

Individual: Person_stlab.istc-cnr1_affiliation
Type: Datum
Facts: hasContent Institute_istc-cnr

Individual: Institute_istc-cnr
Type: Record
Facts: hasDatum Institute_istc-cnr_name

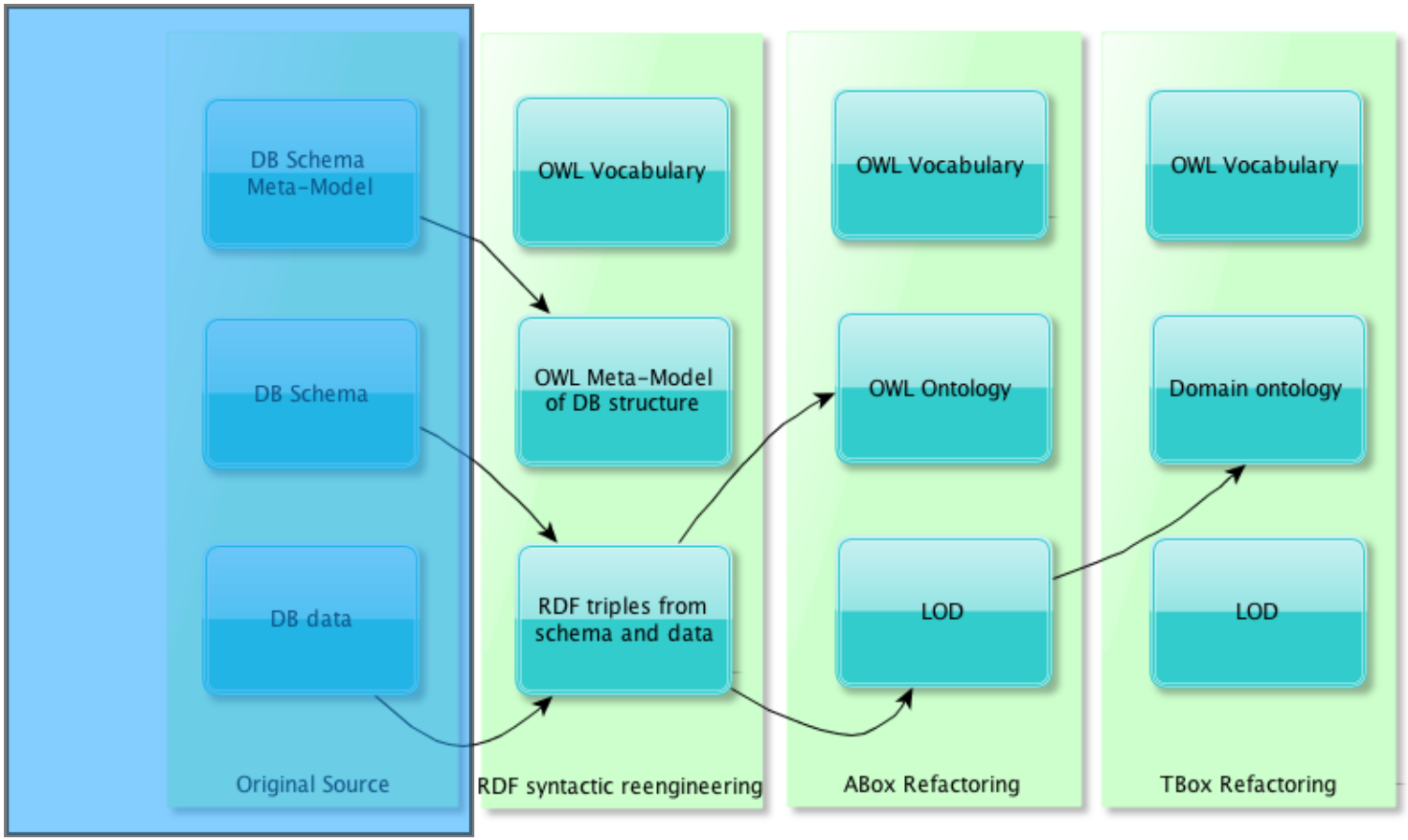
…and for XML


owl:Thing
schema:Element
isElementDeclarationOf : XMLElement
schema:id : string[0..1]
schema:maxOccurs : string[0..1]
schema:minOccurs : string[0..1]
schema:name : string[0..1]
schema:ref : string[0..1]
schema:type : string[0..1]
schema:ComplexType
schema:abstract : string[0..1]
schema:mixed : string[0..1]
schema:name : string[0..1]
SpecialAttrs
isSpecialAttrsOf : XMLElement
base : anyURI[0..]
id : string[0..]
space : [0..]
hasElementDeclaration
isElementDeclarationOf
hasSpecialAttrs
isSpecialAttrsOf
XMLElement
hasElementDeclaration : schema:Element
hasSpecialAttrs : SpecialAttrs
hasXMLAttribute : XMLAttribute
textContent : string
hasXMLAttribute
XMLAttribute
hasAttributeDeclaration : schema:Attribute

The Semion Refactorer

- Allows to align a data set expressed with a specific vocabulary/ontology to another vocabulary/ontology
- Is expressed as a set of rules
- Rules are expressed in a human readable syntax called SemionRule Syntax and can be transformed into
  - SWRL rules for reasoning
  - SPARQL CONSTRUCT for pure refactoring
- Rules realize recipes that can be saved (*refactoring patterns*)

Basic idea



TBox

ABox

The Reengineering View

Semion

Projects Explorer
- progetto
- goodPrj
- wordnet
- sensocomune
- crm
  - Data sources
    - crmRDF
- demo
- demo2

wordnetDB | align | crmRDF

Namespace Prefixes

Prefix	Namespace URI
dbs	http://andriry.altervista.org/tesiSpecialistica/dbs_l1.owl#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl2xml	http://www.w3.org/2006/12/owl2-xml#
dbSchema	http://www.wn.eu/
classification	http://www.ontologydesignpatterns.org/cp/owl/classification.owl#
owl	http://www.w3.org/2002/07/owl#
xsd	http://www.w3.org/2001/XMLSchema#
j.0	http://andriry.altervista.org/tesiSpecialistica/dbs.owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#

Add... Delete

Connection settings

Property	Value
hasTable	http://www.wn.eu/table_adjpositiontypes
hasJDBCDns	jdbc:mysql://localhost:3306/wordnet30
hasTable	http://www.wn.eu/table_samples
hasTable	http://www.wn.eu/table_adjpositions
hasPhysicalName	wordnet30
hasTable	http://www.wn.eu/table_sensesxsemlinksxsenses
hasTable	http://www.wn.eu/table_sensesxsynsets
hasName	database_wordnetDB^^http://www.w3.org/2001/XMLSchema#string
hasTable	http://www.wn.eu/table_samplesets

Add... Modify Delete

Database tables

Name
lexlinks
lexdomains
sensesxsynsets
words
adjpositiontypes
wordsxsensesxsynsets
vframes
casedwords

Add... Modify Delete

Extract data | Select all | Deselect all

Table columns

Name	Type	Primary I	Foreign I	Joins on table	Joins on Colu
vframes-colum	TINYINT UNSIGNED	☑	☐		
vframes-colum	VARCHAR	☐	☐		

Add... Modify Delete

Overview | Source Code | SPARQL

The Refactoring View

Semion

Projects Explorer

- progetto
- goodPrj
- demo3
- wordnet
- sensocomune
  - Data sources
    - sensocomuneFinal
      - Alignments
        - myDemoAlignment
        - myDemoAlignment2
    - newsc
    - sc2
    - newSensoComune
    - sensocomuneRDF
    - scomune
- crm
- demo
- demo2

sensocomuneFinal | myDemoAlignment | myDemoAlignment2

Rules

NAME	RULE
table2Meaning	dbs:Table(?x) -> LMM_L1:Meaning(?x)

Add | Edit | Delete | Run Inference

Source ontology namespaces

PREFIX	NAMESPACE
dbs	http://andriry.altervista.org/tesiSpecialistica/dbs_l1.owl#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl2xml	http://www.w3.org/2006/12/owl2-xml#
dbSchema	http://www.sc.it/
classification	http://www.ontologydesignpatterns.org/cp/owl/classification.o
xsd	http://www.w3.org/2001/XMLSchema#
owl	http://www.w3.org/2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
j.0	http://andriry.altervista.org/tesiSpecialistica/dbs.owl#
objectrole	http://www.ontologydesignpatterns.org/cp/owl/objectrole.owl#
collectionentity	http://www.ontologydesignpatterns.org/cp/owl/collectionentity

Target ontology namespaces

PREFIX	NAMESPACE
	http://www.ontologydesignpatterns.org/ont/lmm/LMM_L2.owl
rdfs	http://www.w3.org/2000/01/rdf-schema#
IOLite	http://www.ontologydesignpatterns.org/ont/dul/IOLite.owl#
owl2xml	http://www.w3.org/2006/12/owl2-xml#
DUL	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#
LMM_L1	http://www.ontologydesignpatterns.org/ont/lmm/LMM_L1.owl
LMM_L2	http://www.ontologydesignpatterns.org/ont/lmm/LMM_L2.owl
owl	http://www.w3.org/2002/07/owl#
xsd	http://www.w3.org/2001/XMLSchema#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#

Overview | Alignment Rules | SPARQL

SemionRule Syntax

```
dbs = <http://ontologydesignpatterns.org/ont/iks/dbs_l1.owl#> .
owl = <http://www.w3.org/2002/07/owl#> .


myRule[
    is(dbs:Table, ?x) . has(dbs:hasColumn, ?x, ?y)
                        ->
    is(owl:Class, ?x)
]
```

as a SPARQL CONSTRUCT

```
PREFIX dbs: <http://ontologydesignpatterns.org/ont/iks/dbs_l1.owl#> .
PREFIX owl: <http://www.w3.org/2002/07/owl#> .
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

CONSTRUCT{ ?x rdf:type owl:Class }
WHERE{
    ?x rdf:type dbs:Table .
    ?x dbs:hasColumn ?y
}
```

…and as a SWRL rule

```
<swrl:Variable rdf:ID="x"/>
<swrl:Variable rdf:ID="y"/>
<swrl:Imp>
    <swrl:body rdf:parseType="Collection">
                        <swrl:ClassAtom>
                                                <swrl:classPredicate rdf:resource="&dbs;Table"/>
                                                <swrl:argument1 rdf:resource="#x" />
                        </swrl:ClassAtom>
                        <swrl:IindividualPropertyAtom>
                                                <swrl:propertyPredicate
                                                                rdf:resource="&dbs;hasColumn"/>
                                    <swrl:argument1 rdf:resource="#x" />
                                    <swrl:argument2 rdf:resource="#y" />
                        </swrl:IindividualPropertyAtom>
    </swrl:body>
    <swrl:head rdf:parseType="Collection">
                        <swrl:ClassAtom>
                                                <swrl:classPredicate rdf:resource="&owl;Class"/>
                                                <swrl:argument1 rdf:resource="#x" />
                        </swrl:ClassAtom>
    </swrl:head>
</swrl:Imp>
```

Stanbol Rule Syntax

in Stanbol a rule is defined as

ruleName[body -> head]

where:

- The ruleName identifies the rule
- The body is a set of **atoms** that must be satisfied when evaluating the rule
- The head or consequent is a set of **atoms** that must be true if the condition is evaluated to be true
- Both body and head consist of a list of conjunctive atoms
  - *body = atom1 . atom2 . … . atomN*
  - *head = atom1 . atom2 . … . atomM*
- The conjunction ∧ in Stanbol Rules is expressed with the symbol " . "

Rule Atoms

- An atom is the smallest unit of a rule
  - e.g.: in predicate calculus

    Person(x) ⇒ hasFather(x, y)

    Person(•) and hasFather(•,•) are two atoms
- In Stanbol basic atoms are
  - Class assertion atom – is(foaf:Person, ?x)
  - Individual assertion atom – has(foaf:knows, ?x, ?y)
  - Data value assertion atom – values(foaf:firstName, ?x, ?fname)
  - …
- There are also comparison atoms, string and integer manipulation atoms
- The atoms may contain
  - constants: they consist of URIs or Literals (values)
    - e.g. http//dbpedia.org/resource/Bob_Marley is a constant, but "Bob Marley"^^xsd:string is a constant too
  - variables: any identifier preceded by **?**
    - e.g. ?x is a variable, but also ?y is a variable

Sample rule

Considering Stanbol Rules, the FOL formula

hasFather(x,y) ∧ hasBrother(y,z) ⇒ hasUncle(x,z)

becomes

myRule[has(<http//myont.org/hasFather>, ?x, ?y) .
has(<http/myont.org/hasBrother>, ?y, ?z)
->
has(<http//myont.org/hasUncle>, ?x, ?z)]

Namespace Prefixes

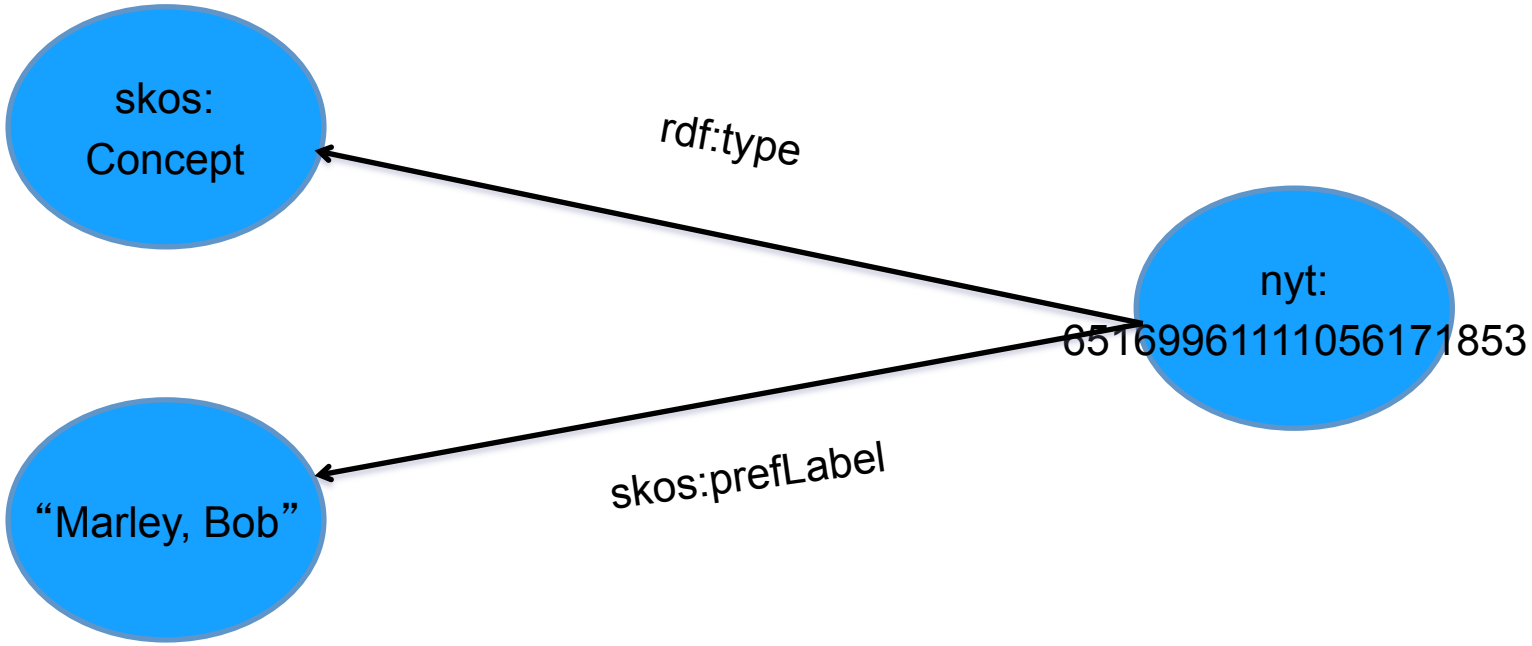
- URIs are useful, but sometime too long for humans
- We can use namespace prefixes instead of full URIs in rule atoms
- e.g:

myont = <http://myont.org/> .

myRule[has(myont:hasFather, ?x, ?y) .
has(myont:hasBrother, ?y, ?z)
->
has(myont:hasUncle, ?x, ?z)]

Define a refactoring recipe

we want to use the FOAF vocabulary instead of SKOS



Define a refactoring recipe

skos = <http://www.w3.org/2004/02/skos/core#> .

foaf = <http://xmlns.com/foaf/0.1/> .

conceptToPerson[is(skos:Concept, ?x) -> is(foaf:Person, ?x)] .

labelRule[values(skos:prefLabel, ?x, ?y) -> values(foaf:name, ?x, ?y)]

Exercise

- Download Semion and launch it as follows:
  - (Mac) java -jar -Xmx512m -XstartOnFirstThread / *LocalPathname*/it.cnr.istc.semion.tool-0.6-SNAPSHOT.one-jar.jar
  - (Win) java -jar -Xmx512m *LocalPathname* \it.cnr.istc.semion.tool-0.6-SNAPSHOT.one-jar.jar
- Connect to the indicated database and perform reengineering first, and alignment (refactoring) second