A method and guidelines for the cooperation of ontologies and relational databases in Semantic Web applications

Loris Bozzato[1] Stefano Braghin[2] Alberto Trombetta[3]

[1]DKM, Fondazione Bruno Kessler – Trento, Italy

[2]School of Computer Engineering, Nanyang Tech. Univ. – Singapore

[3]DISTA, Univ. degli Studi dell'Insubria – Varese, Italy

2nd Int. Workshop on Semantic Digital Archives (SDA 2012)
September 27, 2012 - Paphos, Cyprus

Outline

Outline

Introduction

Ontology based systems:

- Representation of complex and heterogeneous information
- Tools for reasoning over logically founded representations
- Possibility to share and reuse known representations of data

RDB based systems:

- Vast majority of data stored in RDB systems
- Mature optimizations for (query over) very large amounts of data

➔ Efforts for bridging the two representations

Motivation

Motivating application: Imm@base system

- Experience in application needing RDBS and OBS to cooperate (exchange informations in both directions)
- Based on legacy RDB and external classification ontologies
- Idea: use the possibilities of both representations

Problem

- No clear methodology for such cooperations
- No guidelines to decide what to represent and map in both directions

Outline

Motivating application: Imm@base system

Imm@base system

- Large repository of veterinary images in use at IZSLER institute
- Support for advanced classifications of images
- Use: records of veterinary cases data with subject photos

Motivating application: Imm@base system

Imm@base system

- Large repository of veterinary images in use at IZSLER institute
- Support for advanced classifications of images
- Use: records of veterinary cases data with subject photos

IZSLER (`http://www.izsler.it`)

Istituto Zooprofilattico Sperimentale della Lombardia e dell'Emilia Romagna

- A major italian veterinary institute
- Large user base, over more than 15 sites in northern Italy
- Services in:
  - animal disease diagnostics
  - human and animal food control
  - epidemics control
  - research in livestock hygiene

Motivating application: Imm@base system

Imm@base system: components

- Images database: annotated with semantic information from biomedical taxonomies (ICTVdb, NCBI)
- DARWIN system: classification of images with medical cases
- Ontology database:
  domain ontology, external taxonomies, DARWIN mapping

Constraint

DARWIN can not be modified

(e.g. used to quantify refunds in case of epidemics)

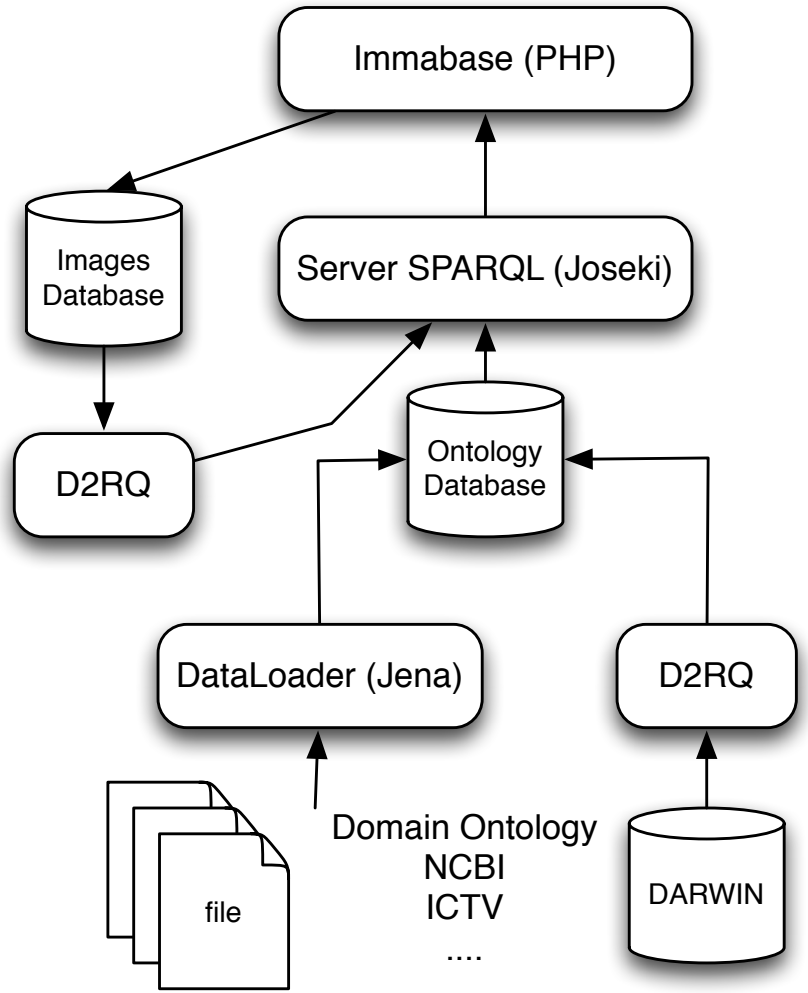
Imm@base architecture

Architecture

- Images DB and Ontology mapped to SPARQL endpoint
- Ontology PostgreSQL DB based on Jena API, Mapping trough D2RQ
- PHP interface for user access trough SPARQL

User interaction

- image upload and annotation from the ontology database (DARWIN + ontology)
- image and medical cases retrieval (ontology based)



Cooperation problem

Problem

- the architecture asks for a clear policy of cooperation between DB and ontology repositories
- How to use the existing mapping tools and languages to let DB and ontology to cooperate?

In particular...

- treatment of consistent references between the two schemas
- integration in an existing repository of an external data source
- identification of static and changing data
- decision on where to store schema instances

Related approaches

Several works about generation of Semantic Web content from data in traditional databases, for example:

- Mapping database model to ontologies: [Sahoo et al., 2009]
  access to content of DB as a semantic repository
  (Mapping languages: D2RQ, R2O...)
  ➔ Problem: only map from DB to Ontology Schema
  (e.g. how to use ontology instances in DB?)
  ➔ Problem: read-only mappings to publish data as Semantic Web data
  (e.g. what about updates in DB?)

Related approaches

Several works about generation of Semantic Web content from data in traditional databases, for example:

- Mapping database model to ontologies: [Sahoo et al., 2009]
  access to content of DB as a semantic repository
  (Mapping languages: D2RQ, R2O...)
  ➔ Problem: only map from DB to Ontology Schema
  (e.g. how to use ontology instances in DB?)
  ➔ Problem: read-only mappings to publish data as Semantic Web data
  (e.g. what about updates in DB?)
- Generation of ontology from DB relational model [Stojanovic et al., 2002]
  with migration of DB data as Ontology instances
  ➔ Problem: the ontology already exists (and different from DB schema)

Related approaches

Several works about generation of Semantic Web content from data in traditional databases, for example:

- Mapping database model to ontologies: [Sahoo et al., 2009]
  access to content of DB as a semantic repository
  (Mapping languages: D2RQ, R2O...)
  ➔ Problem: only map from DB to Ontology Schema
  (e.g. how to use ontology instances in DB?)
  ➔ Problem: read-only mappings to publish data as Semantic Web data
  (e.g. what about updates in DB?)
- Generation of ontology from DB relational model [Stojanovic et al., 2002]
  with migration of DB data as Ontology instances
  ➔ Problem: the ontology already exists (and different from DB schema)
- Automatic schema matching and merging [Rahm and Bernstein, 2001]
  ➔ Problem: merging is only part of the problem
  (e.g. distribution of schema instances?)

Outline

Cooperation method: introduction

We generalize the approach we followed in the Imm@base system:

- General method for integration of DB and ontology based applications
- Formulated as a re-engineering method [Villazón-Terrazas, 2009]

Goals:

- guide separation of data between RDB and ontology instances
- define suitable mapping between the two repositories

Cooperation method: introduction

What the method is

A sequence of processes to

- decide how and what to map between the two schemas
- decide how to distribute data taking advantage of the possibilities of the two representations

What the method is not

- A new mapping language or tool
- A method for ontology engineering
- A method for DB and ontology schema merging

Cooperation method: preliminary notions

- The method operates over conceptual representations of the repositories
- The method does not assume existence of both sources
  if one schema exists, conceptual model is extracted;
  otherwise, model defined from requirements
- Some of its tasks are easily automatizable
  e.g. conceptual model extraction from existing knowledge bases

Cooperation method: mapping tables

Output of the method: entity table and properties table

- for each conceptual object, specification of where to store instances and how to refer to them
- tables can be used to define formal mapping between the sources (e.g. by mapping languages)

Cooperation method: mapping tables

Output of the method: entity table and properties table

- for each conceptual object, specification of where to store instances and how to refer to them
- tables can be used to define formal mapping between the sources (e.g. by mapping languages)

Entity table: for each conceptual entity

- Ontology class, DB table: where its instances are stored
- Mapping: logical mapping on classes and DB tables
- ID: property chosen as identifier
- Source: original source

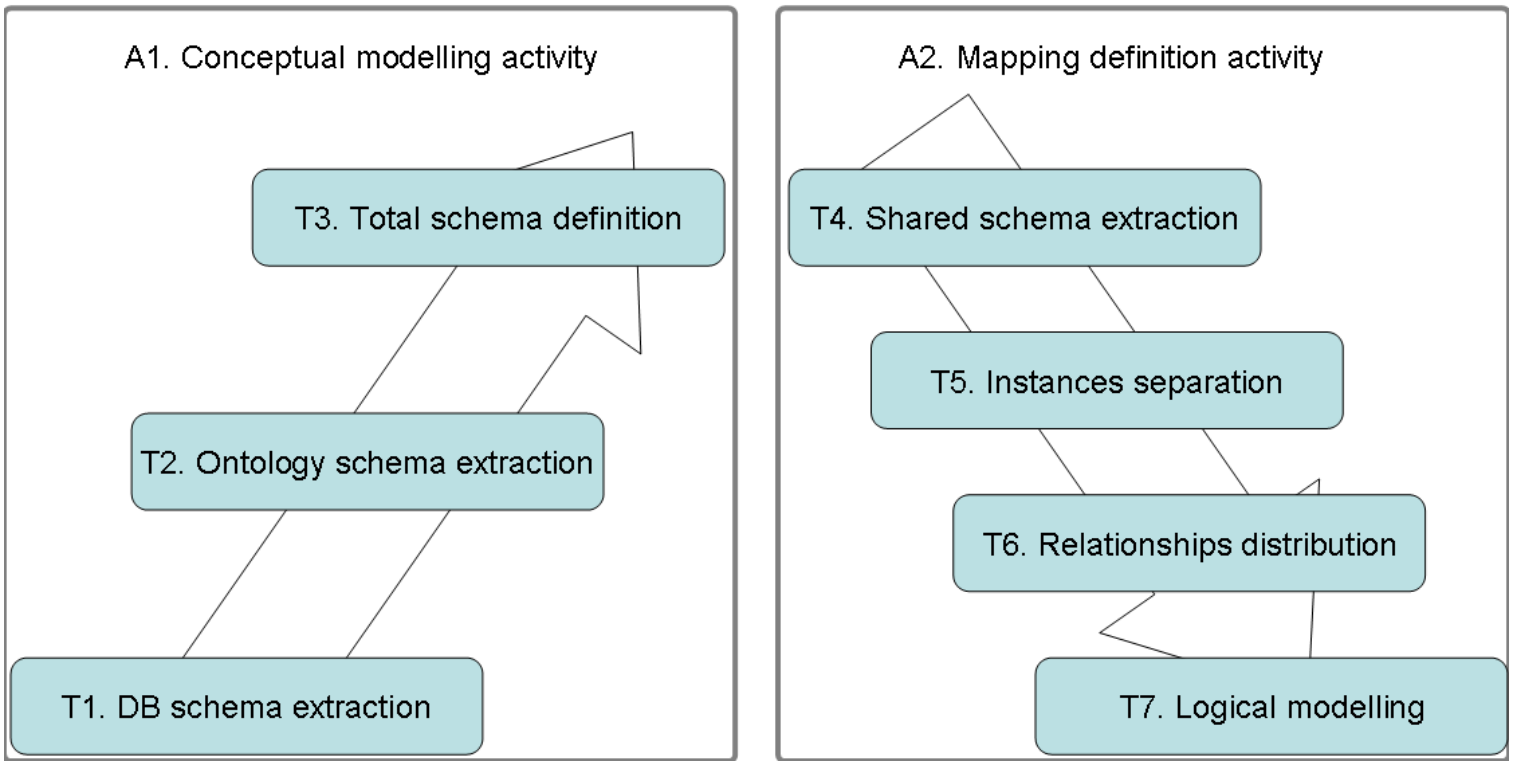
Properties table: for each relationship

- Ontology property, DB column: where its instances are stored,
- Mapping: mapping on ontology properties and DB columns,
- Domain and range: conceptual entities linked by the property.

Cooperation method: method specification

The method is divided in 2 Activities [Villazón-Terrazas, 2009]:

- Conceptual Modeling Activity:
  Reverse engineering on sources to extract complete conceptual schema
- Mapping Definition Activity:
  Forward engineering on extracted model for definition of mapping tables



Running example: introduction

Running example:

- a simplification of actual integration of Images DB schema and Ontology schema in Imm@base system
- DB contains data related to the medical images files
- ontology stores properties of images subjects

We assume to already have the ontology conceptual schema and to be able to extract the conceptual schema from DB tables

Method: reverse engineering activity A1

A1. Conceptual Modeling Activity
Extraction of conceptual representation for entire system
Input: DB and ontology, their conceptual models or requirements
Output: Total conceptual model

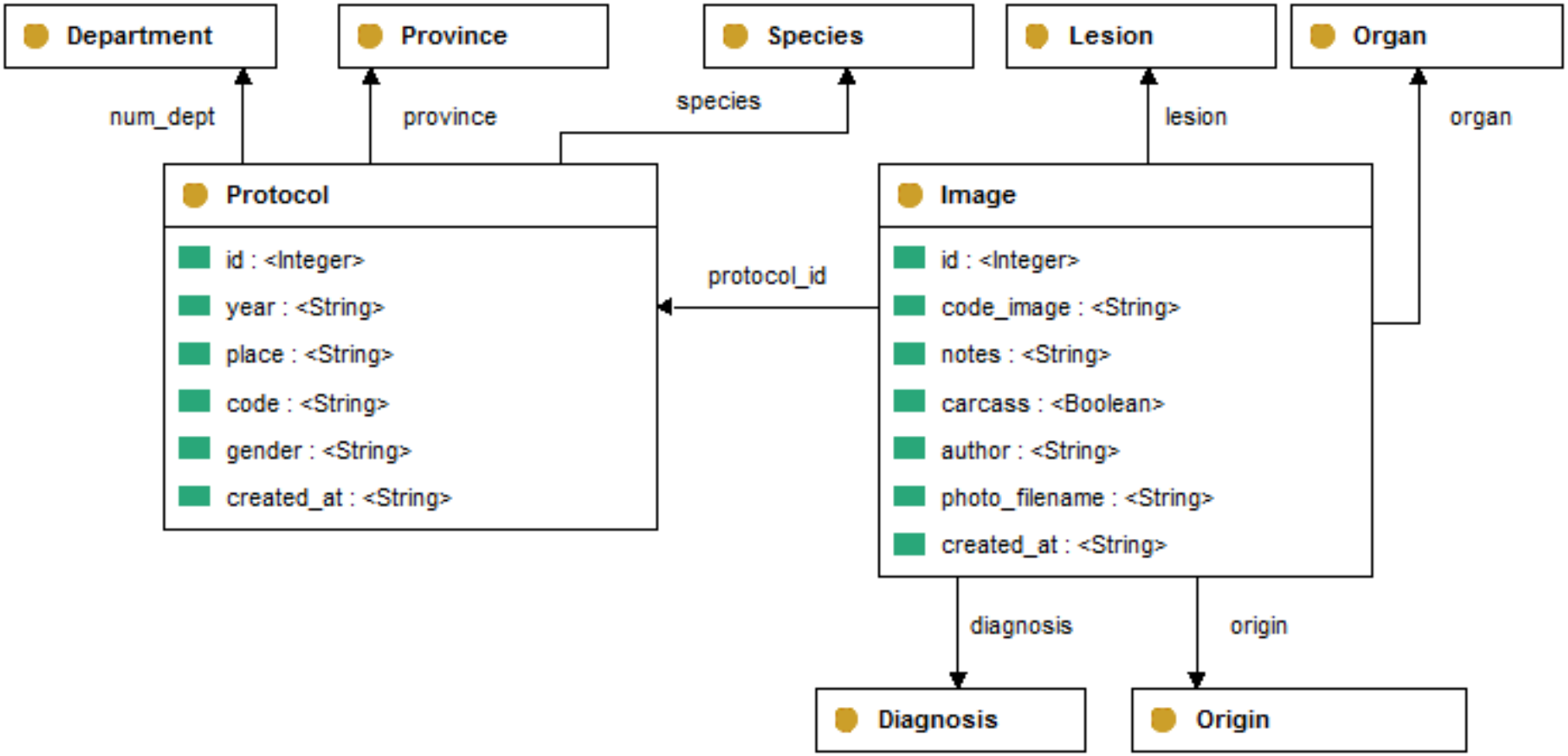
T1. DB schema extraction: extract conceptual schema from DB
Input: DB, requirements or original DB schema
Output: DB conceptual schema

T2. Ontology schema extraction: extract conceptual schema from ontology
Input: Ontology, requirements or original ontology schema
Output: Ontology conceptual schema

T3. Total schema definition:
merge previous schemas to obtain a complete system schema
Input: DB and ontology schemas
Output: Total conceptual model

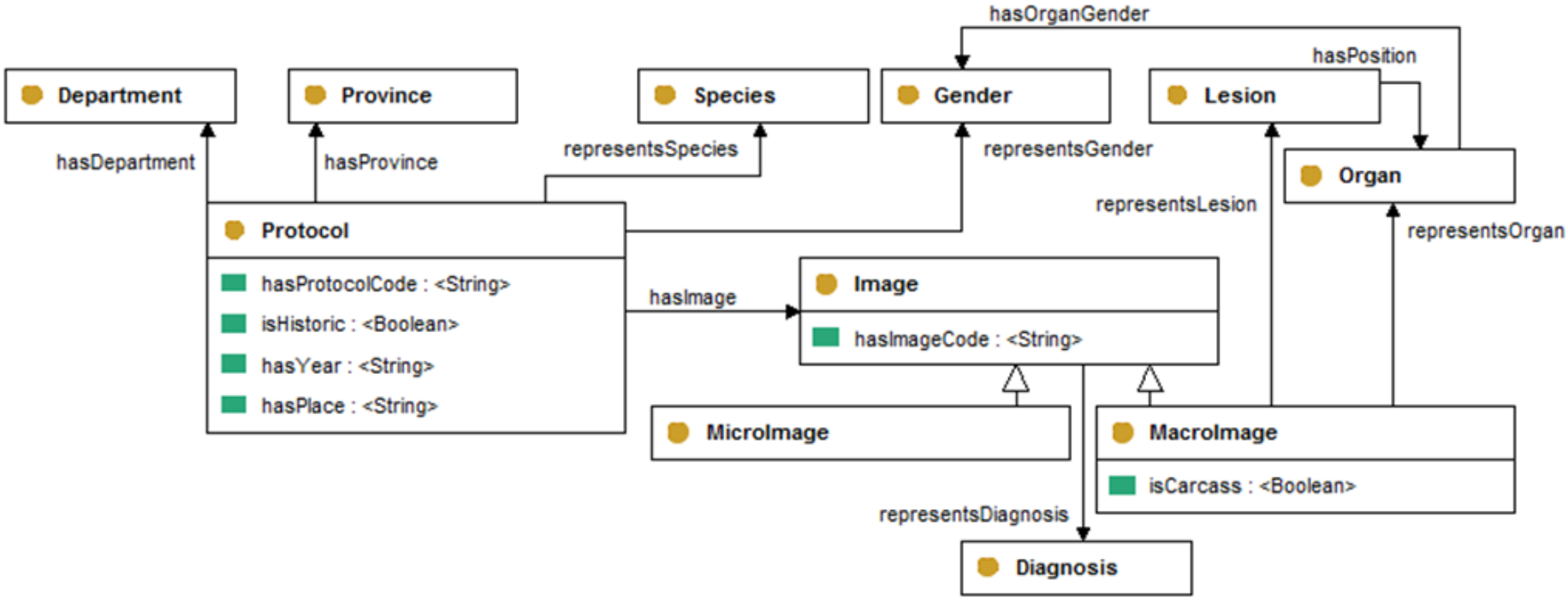
Example: reverse engineering activity A1

T1. DB schema extraction



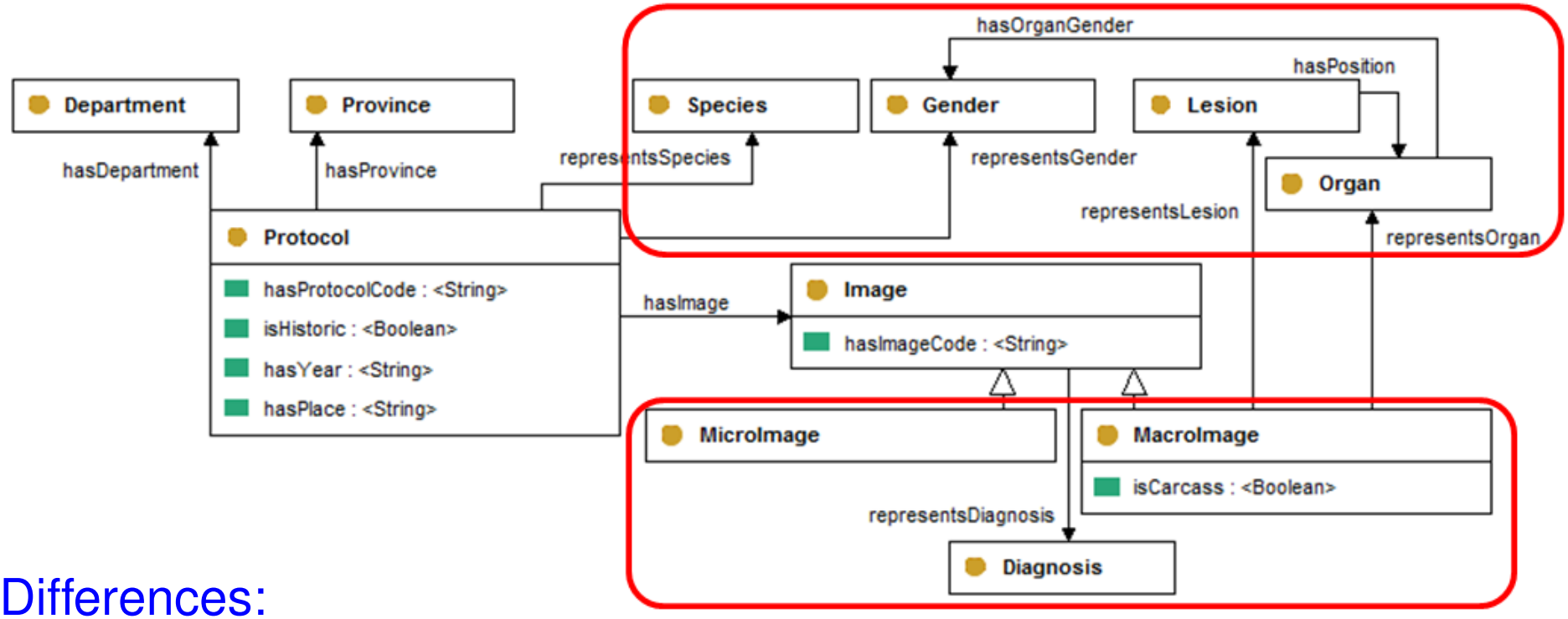
Example: reverse engineering activity A1

T2. Ontology schema extraction



Example: reverse engineering activity A1

T2. Ontology schema extraction

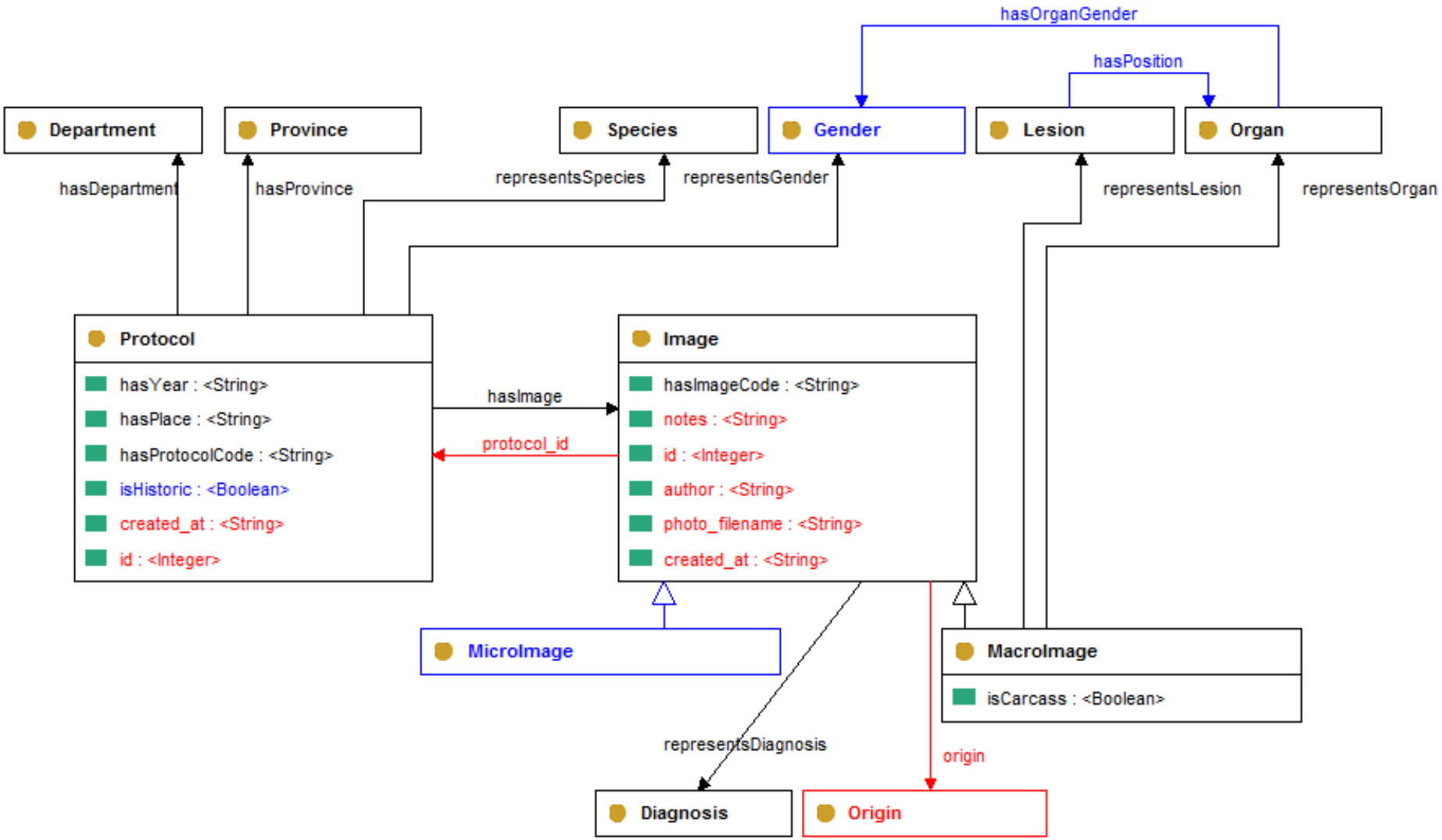


Differences:

- Origin only in DB schema
- Gender as relation, not attribute
- Image specialized in Micro/MacroImage
- Ontology contains relations between subject properties

Example: reverse engineering activity A1

T3. Total schema definition: merge of 2 schemas (e.g. gender...)



Method: forward engineering activity A2

A2. Mapping Definition Activity
Decision on separation of instance and mapping definition
Input: Total conceptual model
Output: Complete mapping tables

T4. Shared schema extraction:
identify conceptual objects to be shared between schemas
Input: Total conceptual model
Output: Shared conceptual schema

T5. Instances distribution:
for every entity, decide where to store its instances
Input: Shared conceptual schema
Output: Entities table

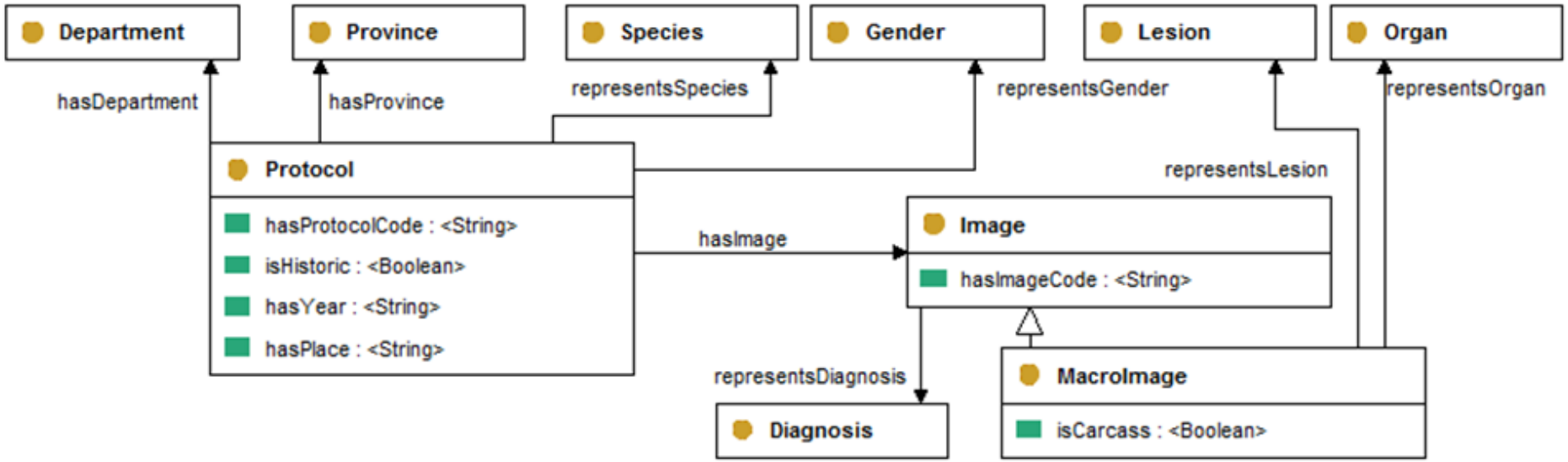
Method: forward engineering activity A2

T6. Relationships distribution:
for every relationship, decide where to store its instances
Input: Shared schema, entities table
Output: Properties table

T7. Logical modeling:
define actual classes and tables to be mapped
Input: Shared schema, mapping tables
Output: Complete mapping tables

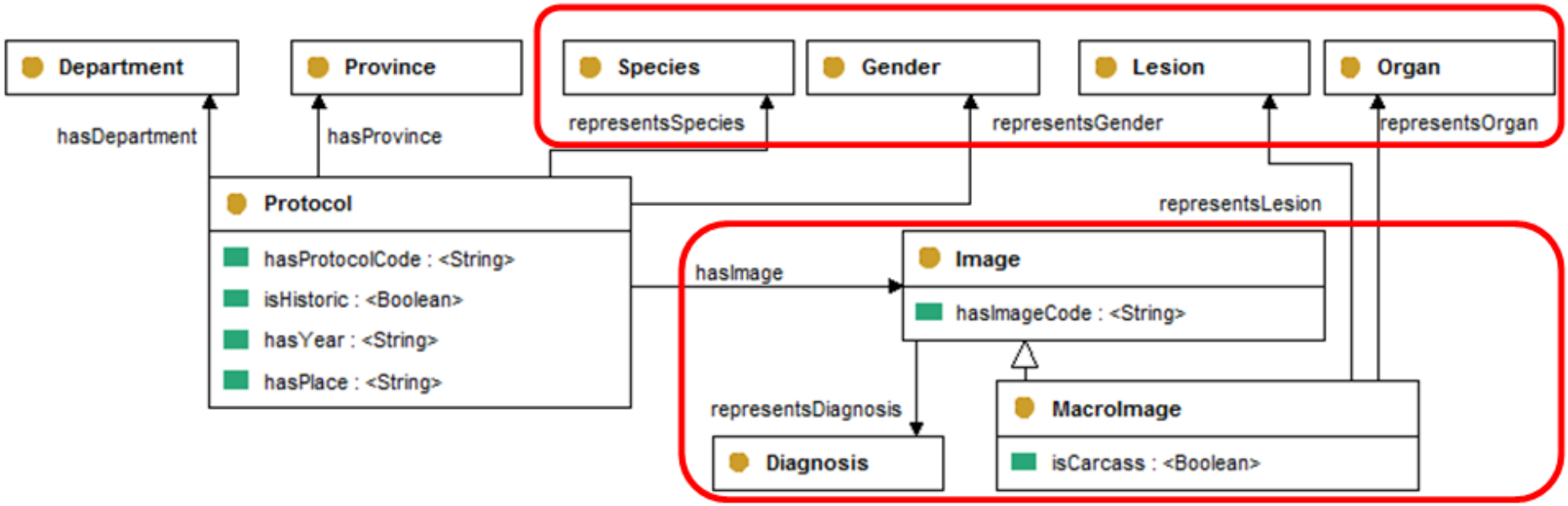
Example: forward engineering activity A2

T4. Shared schema extraction



Example: forward engineering activity A2

T4. Shared schema extraction



What is not shared:

- Only in DB: file related attributes (Origin, author, notes...)
- Only in Ontology: MicroImage, subject properties relations

Example: forward engineering activity A2

T5. Instances distribution

Entity	Mapping	Onto.Class	DB Table	ID	Source
Protocol	DB - O	-	-	-	O, DB
Image	DB - O	-	-	-	O, DB
MacroImage	DB - O	-	-	-	O
MicroImage	O	-	-	-	O
Species	O - DB	-	-	-	O, DB
Gender	O - DB	-	-	-	O
Organ	O - DB	-	-	-	O, DB
Origin	DB	-	-	-	DB

Example: forward engineering activity A2

T6. Relationships distribution

Domain	Property	Range	Type	Mapping	DB column
Image (DB)	representsDiagnosis origin hasImageCode notes author ...	Diagnosis (O) Origin (DB) <string> <string> <string> ...	object object datatype datatype datatype ...	DB - O DB DB - O DB DB ...	- - - - - ...
MacroImage (DB)	representsOrgan representsLesion isCarcass	Organ (O) Lesion (O) <boolean>	object object datatype	DB - O DB - O DB - O	- - -
Lesion (O)	hasPosition	Organ (O)	object	O	-

Example: forward engineering activity A2

T7. Logical modeling

Entity	Mapping	Onto.Class	DB Table	ID	Source
Protocol	DB - O	Protocol	Protocol	Protocol.ID	O, DB
Image	DB - O	Image	Image	Image.ID	O, DB
MacroImage	DB - O	MacroImage	Image	Image.ID	O
MicroImage	O	MicroImage	-	URI	O
Species	O - DB	Species	-	URI	O, DB
Gender	O - DB	Gender	-	URI	O
Organ	O - DB	Organ	-	URI	O, DB
Origin	DB	-	Origin	Origin.ID	DB

Example: forward engineering activity A2

T7. Logical modeling

Domain	Property	Range	Type	Mapping	DB column
Image (DB)	representsDiagnosis hasImageCode notes author ...	Diagnosis (O) <string> <string> <string> ...	object datatype datatype datatype ...	DB - O DB - O DB DB ...	Image.origin Image.code Image.notes Image.author ...
MacroImage (DB)	representsOrgan representsLesion isCarcass	Organ (O) Lesion (O) <boolean>	object object datatype	DB - O DB - O DB - O	Image.organ Image.lesion Image.carcass
Lesion (O)	hasPosition	Organ (O)	object	O	-

Outline

Guidelines: instance separation

Which schema refer to when storing instances of a conceptual object?

- Ontology instances
  - necessary to draw inferences from them
  - arranged in complex taxonomies or meronimies
  - need to verify consistency w.r.t. ontology logical constraints
  - need to comply with imported (standard) ontology
  - fixed and non-changing data
  - represent application metadata
- RDB instances
  - represent working application data
  - expected to be updated and changed
  - only simple queries (and no inferences)
  - large data volume and query processing

Guidelines: mapping direction

How to map/identify instances in the 2 directions?

- Instances in DB, class in ontology (mapping tools)
  - From DB to ontology: entity instances mapped to individuals of ontology class
  - Entities identified from ontology instances by mapping their primary key as `hasID` datatype value
- Instances in ontology, values in DB
  - Solution 1: use the URI of individual in DB tuples references
    Problem: check consistency of the references
  - Solution 2: DB table relating DB tuple to individual in ontology
    Plus: additional data for each instance can be stored in table
  - Solution 3: use URI of instance as ID of the counterpart tuple in DB
  - Solution 3': define a transformation from URI to ID in DB

Outline

Conclusions

Conclusions:

- We presented a method for collaboration of RDB and OBS
- Towards a fruitful distribution of data between them
- Guidelines to support decisions in method deployment
- Generalization from scenario of Imm@base system

Conclusions

Conclusions:

- We presented a method for collaboration of RDB and OBS
- Towards a fruitful distribution of data between them
- Guidelines to support decisions in method deployment
- Generalization from scenario of Imm@base system

Considerations:

- First step towards definition of a generally applicable re-engineering process
- Necessary to refine and evaluate with experiences on several real-world scenarios
- We want to draw the attention to some relevant aspects of the cooperation and possibly promote discussion

Conclusions

Conclusions:

- We presented a method for collaboration of RDB and OBS
- Towards a fruitful distribution of data between them
- Guidelines to support decisions in method deployment
- Generalization from scenario of Imm@base system

Considerations:

- First step towards definition of a generally applicable re-engineering process
- Necessary to refine and evaluate with experiences on several real-world scenarios
- We want to draw the attention to some relevant aspects of the cooperation and possibly promote discussion

Directions:

- Study of automatization possibilities
- Tasks implementation
- Study needs for consistency checks and DB transactions

Thank you for listening

A method and guidelines for the cooperation of ontologies and relational databases in Semantic Web applications

Loris Bozzato, Stefano Braghin, Alberto Trombetta

bozzato@fbk.eu
s.braghin@ntu.edu.sg
alberto.trombetta@uninsubria.it

References I

Rahm, E. and Bernstein, P. A. (2001).

A survey of approaches to automatic schema matching.
VLDB J., 10(4):334–350.

Sahoo, S., Halb, W., Hellmann, S., Idehen, K., Thibodeau, T., Auer, S., Sequeda, J., and Ezzat, A. (2009).

A Survey of Current Approaches for Mapping of Relational Databases to RDF.
RDB2RDF XG Report, W3C.
`http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf`.

Stojanovic, L., Stojanovic, N., and Volz, R. (2002).

Migrating data-intensive web sites into the semantic web.
In *Proceedings of SAC*, pages 1100–1107. ACM.

Villazón-Terrazas, B. (2009).

D2.2.2. Methods and Tools Supporting Re-engineering.
NeOn Project Deliverable D2.2.2/v2.0, NeOn.