Ontologies in data integration

Prof. Letizia Tanca

Technologies for Information Systems

The new application context (recall)

- A (possibly large) number of data sources
- Time-variant data (e.g. WEB)
- Heterogeneous data sources
- Mobile, transient data sources
- Different levels of data structure
  - Databases (relational, OO…)
  - Semi-structured data sources (XML, HTML, more markups …)
  - Unstructured data (text, multimedia etc…)
- Different terminologies and different operational contexts

Ontologies

- A formal and shared definition of a vocabulary of terms and their inter-relationships
- Predefined relations:
  - *synonimy*
  - *omonimy*
  - *hyponimy*
  - *etc..*
- More complex, designer-defined relationships, whose semantics depends on the domain

  → e.g. enrolled(*student,course)*

A philosophical concept…

- Introduced by Aristoteles
- The science of being, i.e. the science of what is
- Ontology, as a philosophical discipline, studies the answers to questions like:
  - ***What does “being” mean?***
  - ***What are the features common to all beings?***

Definitions

- Ontology = formal specification of a conceptualization of a shared knowledge domain.
- An ontology is a controlled vocabulary that describes objects and the relationships between them in a formal way
- It has a grammar for using the terms to express something meaningful within a specified domain of interest.
- The vocabulary is used to express queries and assertions.
- Ontological commitments are agreements to use the vocabulary in a consistent way for knowledge sharing

semantic interoperability → semantic Web

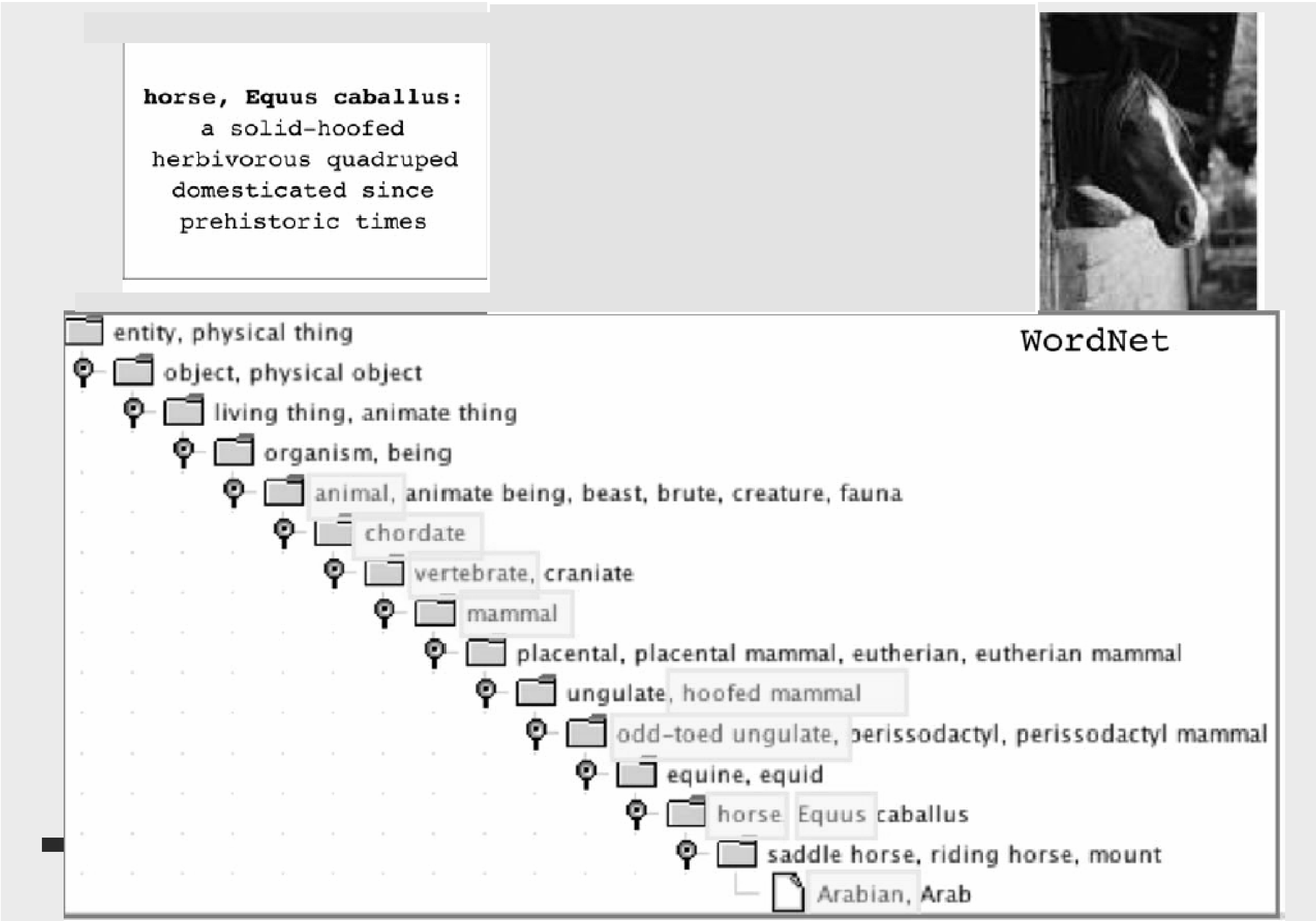
Aims…

- A formal specification allows for use of a common vocabulary for *automatic knowledge sharing*
- Formally specifying a *conceptualization* means giving a unique meaning to the terms that define the knowledge about a given domain
- *Shared:* an ontology captures knowledge which is common, thus *over which there is a consensus* (objectivity is not an issue here)

Ontology types

- **Taxonomic ontologies**
  - Definition of concepts through terms, their hierarchical organization, and additional relationships (synonymy,composition,…)
  - To provide a reference vocabulary
- **Descriptive ontologies**
  - Definition of concepts through data structures and their interrelationships
  - Provide information for “aligning” existing data structures or to design new, specialized ontologies (***domain ontologies***)
  - Closer to the database area techniques

Wordnet


horse, Equus caballus: a solid-hoofed herbivorous quadruped domesticated since prehistoric times
WordNet
entity, physical thing
object, physical object
living thing, animate thing
organism, being
animal, animate being, beast, brute, creature, fauna
chordate
vertebrate, craniate
mammal
placental, placental mammal, eutherian, eutherian mammal
ungulate, hoofed mammal
odd-toed ungulate, perissodactyl, perissodactyl mammal
equine, equid
horse, Equus caballus
saddle horse, riding horse, mount
Arabian, Arab

An ontology consists of…

- Concepts:
  - Generic concepts, they express general world categories
  - Specific concepts, they describe a particular application domain (***domain ontologies***)
- Concept Definition
  - Via a formal language
  - In natural language
- Relations between concepts:
  - Taxonomies (IS_A),
  - Meronymies (PART_OF),
  - User-defined associations,
  - Synonymies, homonymies, ...

Formal Definitions

O = (C, R, I, A)

O ontology, *C* concepts, *R* relations, *A* axioms

- Specified in some logic-based language
- Organized in a ISA hierarchy
- *I*= instance collection, stored in the information source
- Composed by a *T-Box* (theory) and an *A-box* (instances)

Formal Definitions

An ontology is (part of) a knowledge base, composed by:

- a ***T-Box****:* contains all the concept and role definitions, and also contains all the axioms of our logical theory (e.g. “A father is a Man with a Child”).
- an ***A-box***: contains all the basic assertions (also known as ground facts) of the logical theory (e.g. “Tom is a father” is represented as Father(Tom)).

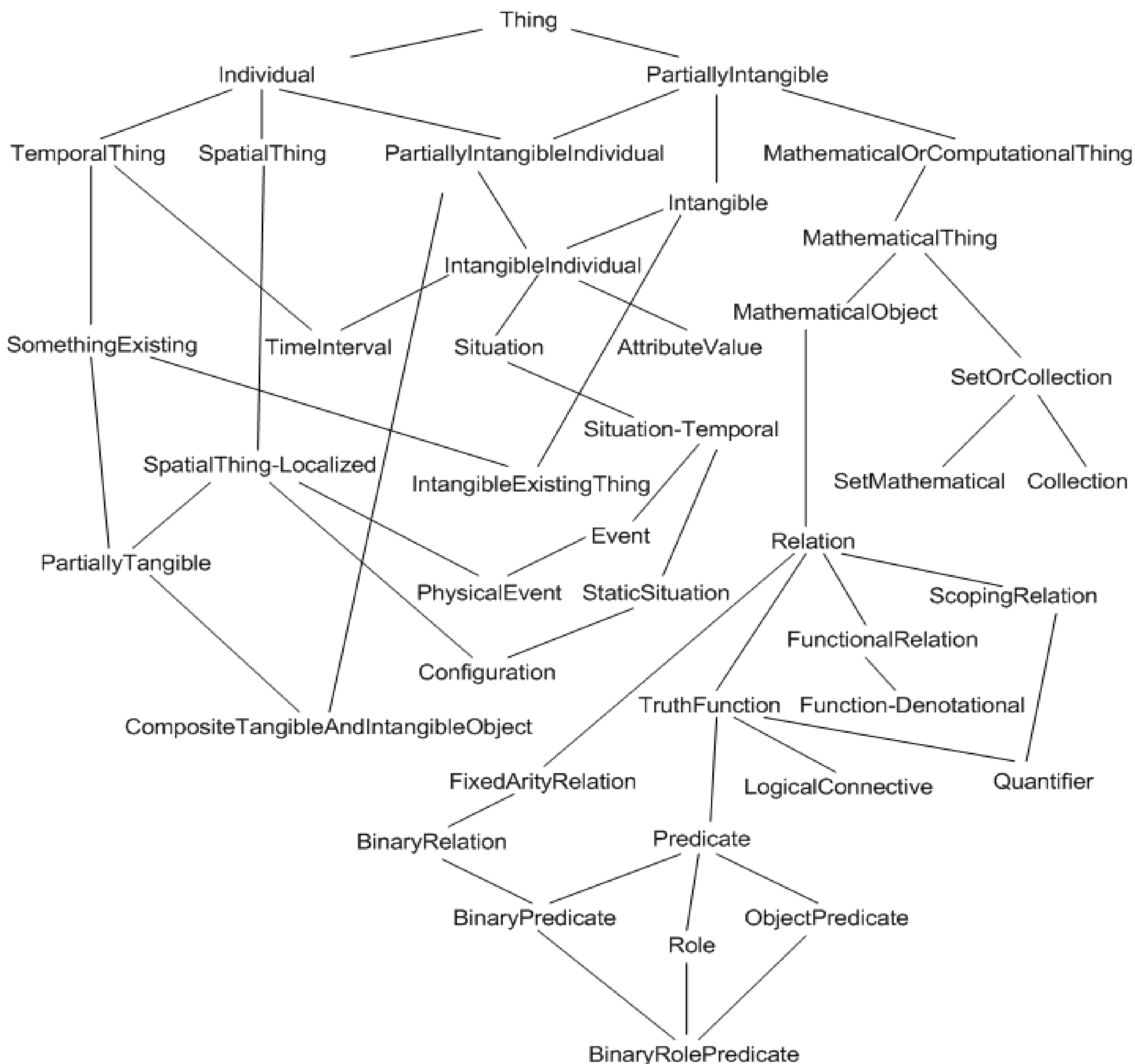
OpenCyc

- The open source version of the Cyc technology
- The world's largest and most complete general knowledge base and commonsense reasoning engine
- The Cyc project was born in 1984 and is still continuing http://www.cyc.com/opencyc

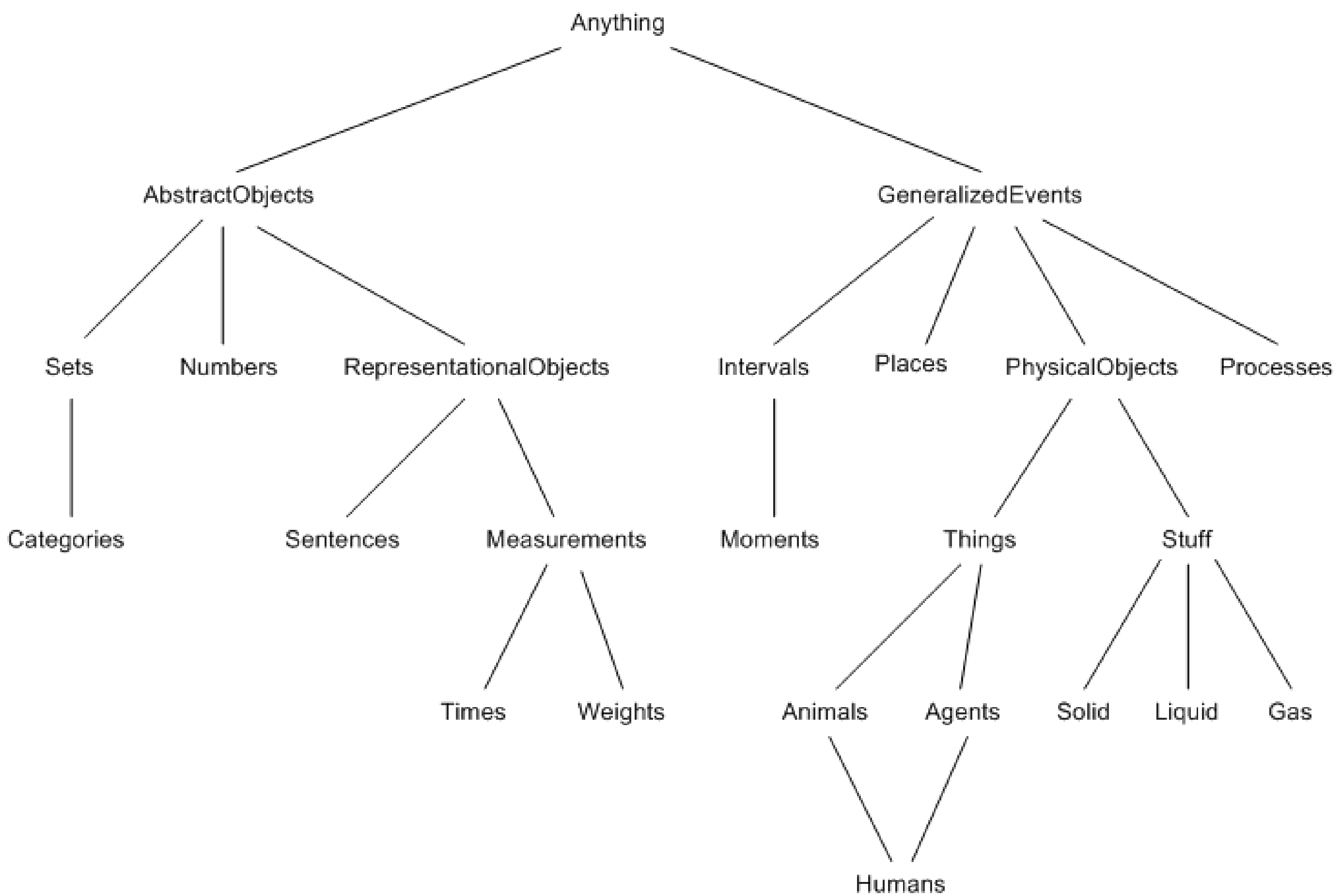
Release 1.0 of OpenCyc

- **6,000 concepts: an upper ontology for all of human consensus reality.**
- **60,000 assertions about the 6,000 concepts, interrelating them, constraining them, in effect (partially) defining them.**
- **A compiled version of the Cyc Inference Engine and the Cyc Knowledge Base Browser.**
- **A suite of tools for rapidly extracting knowledge from a domain expert, such as a physician or an oil drilling specialist.**
- **Documentation and self-paced learning materials to help users achieve a basic- to intermediate-level understanding of the issues of knowledge representation and application development using Cyc.**
- **A specification of CycL, the language in which Cyc (and hence OpenCyc) is written. There are CycL-to-Lisp, CycL-to-C, etc. translators.**
- **A specification of the Cyc API, by calling which a programmer can build an OpenCyc application with very little familiarity with CycL or with the OpenCyc KB.**
- **The ability to import and export CycML files.**
- **Sample programs that demonstrate use of the Cyc API for application development.**

Top level concepts of Cyc



Top level concepts of the Russel and Norvig ontology



The Semantic Web

- a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.
- will build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data.
- The first level above RDF: OWL, an ontology language what can formally describe the meaning of terminology used in Web documents → beyond the basic semantics of RDF Schema.

A fragment of an RDF (XML) document, describing an ontology.
The language is OWL
http://www.w3.org/TR/owl-ref/

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns="http://eng.it/ontology/tourism#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://eng.it/ontology/tourism">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Church">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Definition: Edificio sacro in cui si svolgono pubblicamente gli atti
di culto delle religioni cristiane.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#PlaceOfWorship"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Theatre">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Definition: a building where theatrical performances or motion-
picture shows can be presented.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#SocialAttraction"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="DailyCityTransportationTicket">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#CityTransportationTicket"/>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Definition: Biglietto che consente di usufruire di un numero
illimitato di viaggi sui mezzi pubblici (autobus e metropolitana)
all’interno del centro urbano (o della regione, con un costo maggiore)
per un periodo di 24 ore.</rdfs:comment>
  </owl:Class>
```

OWL

- The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans.
- OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics.
- OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

OWL

- Designed to meet the need for a Web Ontology Language. OWL is part of the growing stack of W3C recommendations related to the Semantic Web.
- XML provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents.
- XML Schema is a language for restricting the structure of XML documents and also extends XML with data types.
- RDF is a data model for objects ("resources") and relations between them, provides a simple semantics for this data model, and can be represented in an XML syntax.
- RDF Schema is a vocabulary for describing properties and classes of RDF resources, with a semantics for generalization-hierarchies of such properties and classes.
- OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.

OWL SUBLANGUAGES: ***OWL Lite***

Supports users primarily needing a classification hierarchy and simple constraints.

- Cardinality constraints: it only permits cardinality values of 0 or 1.
- Has a lower formal complexity than OWL DL
- It is simpler to provide tool support for OWL Lite than for its more expressive relatives
- OWL Lite provides a quick migration path for thesauri and other taxonomies.

OWL SUBLANGUAGES: ***OWL DL***

*S*upports users who want maximum expressiveness while:

- all conclusions are guaranteed to be computable (computational completeness)
- all computations will finish in finite time (decidability)
- includes all OWL language constructs, but they can be used only under certain restrictions
  - for example, while a class may be a subclass of many classes, a class cannot be an instance of another class
  - so named due to its correspondence with *description logics*, the logics that form the formal foundation of OWL.

OWL SUBLANGUAGES: *OWL FULL*

Meant for users who want maximum expressiveness and the syntactic freedom of RDF

- no computational guarantees
  - For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right.
- OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary
- unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Further existing projects

- RACER : a description logic reasoning system which implements the SHIQ Logic.
- KAON : an ontology and semantic web framework allowing the design and management of ontologies
- DOGMA : an ontology engineering framework based on the ORM (Object-Role-Modeling) conceptual model
- MADS : a spatio-temporal conceptual model (complex objects, n-ary relationships with attributes, generalization hierarchies, spatio/temporal and contextual features)

References

- RACER: http://www.sts.tu-harburg.de/~r.f.moeller/racer/
- KAON: http://kaon.semanticweb.org/
- DOGMA: M Jarrar, J Demey, R Meersman *"On Using Conceptual Data Modeling for Ontology Engineering",* Journal on Data Semantics, 2003 – Springer Verlag
- MADS: Christine Parent , Stefano Spaccapietra ,Esteban Zimányi, "*Spatio-temporal conceptual models: data structures + space + time*", Proc. 7th ACM international Symp. on Advances in Geographic Information Systems, Kansas City, USA,1999

Reasoning services

Services for the Tbox

- Subsumption: verifies if a concept C is subsumed by (is a subconcept of) another concept D
- Consistency: verifies that there exists at least one interpretation I for a given Tbox
- Local Satisfiability: verifies, for a given concept C, that there exist at least one interpretation in which C is true.

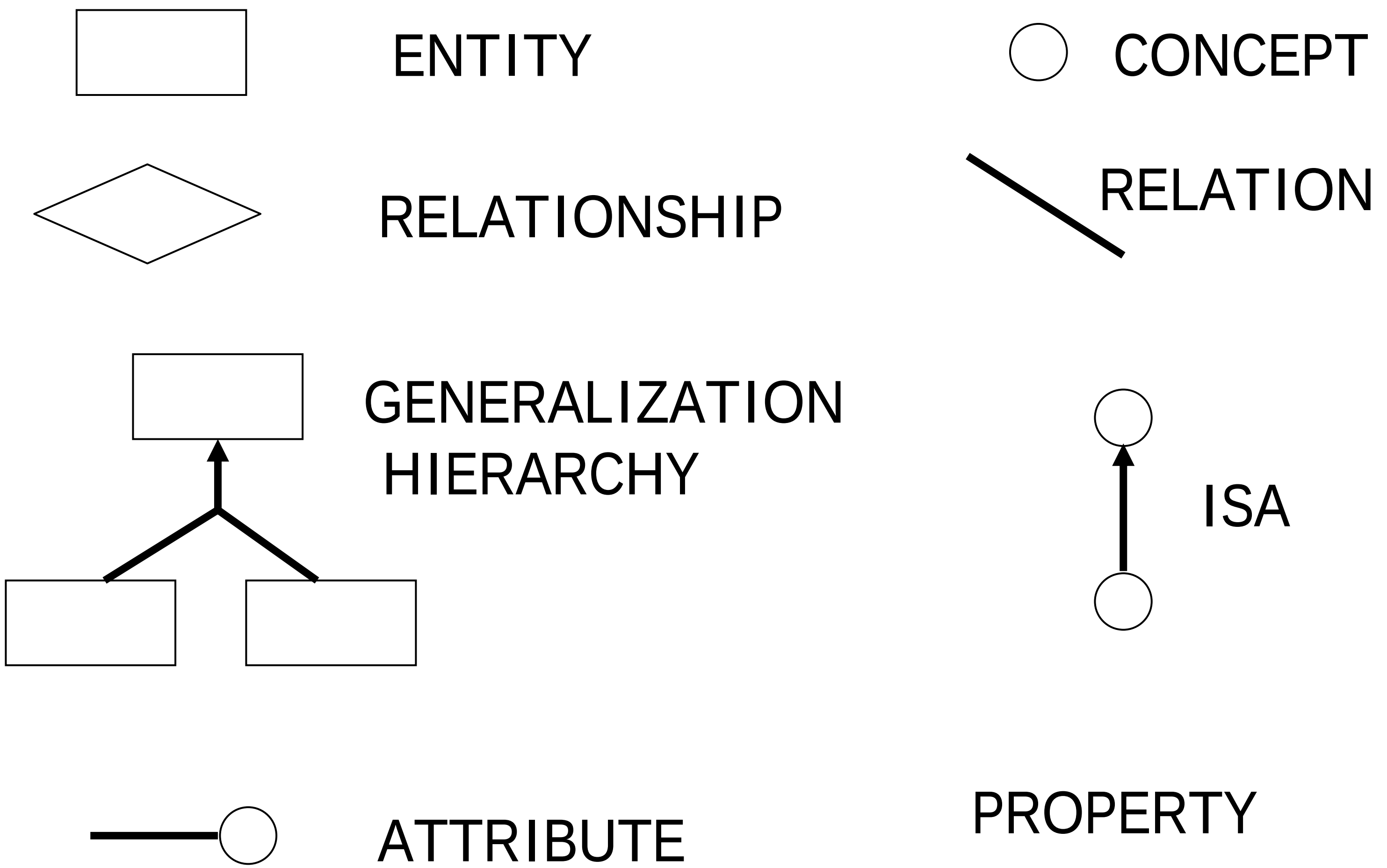
Services for the Abox

- Consistency: verifies that an Abox is consistent with respect to a given Tbox
- Instance Checking: verifies if a given individual x belongs to a particular concept C
- Instance Retrieval: returns the extension of a given concept C, that is, the set of individuals belonging to C.

Comparison

- analysis of the features of a descriptive ontology (data structures, instance management, constraint definition, queries)
- compare these features with the functionality provided by current representation approaches from the database world

e.g. ER vs.ontology



Comparison

Descriptive ontologies require rich models to enable representations close to human perception

	DL	DB
Complex data structures	**No**	**yes**
Generalization/specialization hierarchies	**yes**	**yes**
Defined concepts	**yes**	**no**

DB versus ontologies

How to improve database conceptual models to fulfill ontology requirements ?

- Supporting defined concepts and adding the necessary reasoning mechanisms
- Managing missing and incomplete information: <u>semantic differences</u> between the two assumptions made w.r.t. missing information (*Closed World Assumption* vs. *Open World Assumption*)

How can ontologies support integration?

- An ontology instead of a global schema
- An ontology as a schema integration support tool
- An ontology as a support tool for content interpretation and wrapping (e.g. HTML pages)
- An ontology as a support tool for content inconsistency detection and resolution

Ontologies and integration problems

- Discovery of “equivalent” concepts (mapping)
  - What does equivalent mean?
- Formal representation of these mappings
  - How are these mappings represented?
- Reasoning on these mappings
  - How do we use the mappings within our reasoning and query-answering process?

Ontology *matching*

- The process of finding pairs of resources coming from different ontologies which can be considered equal in meaning–*matching operators*
- The similarity value is usually a number in the interval [0,1] (fuzzy)
- It is an input to the different approaches to integration, described below
- Mediation may be done without integrating the ontologies, but using the matchings in different ways

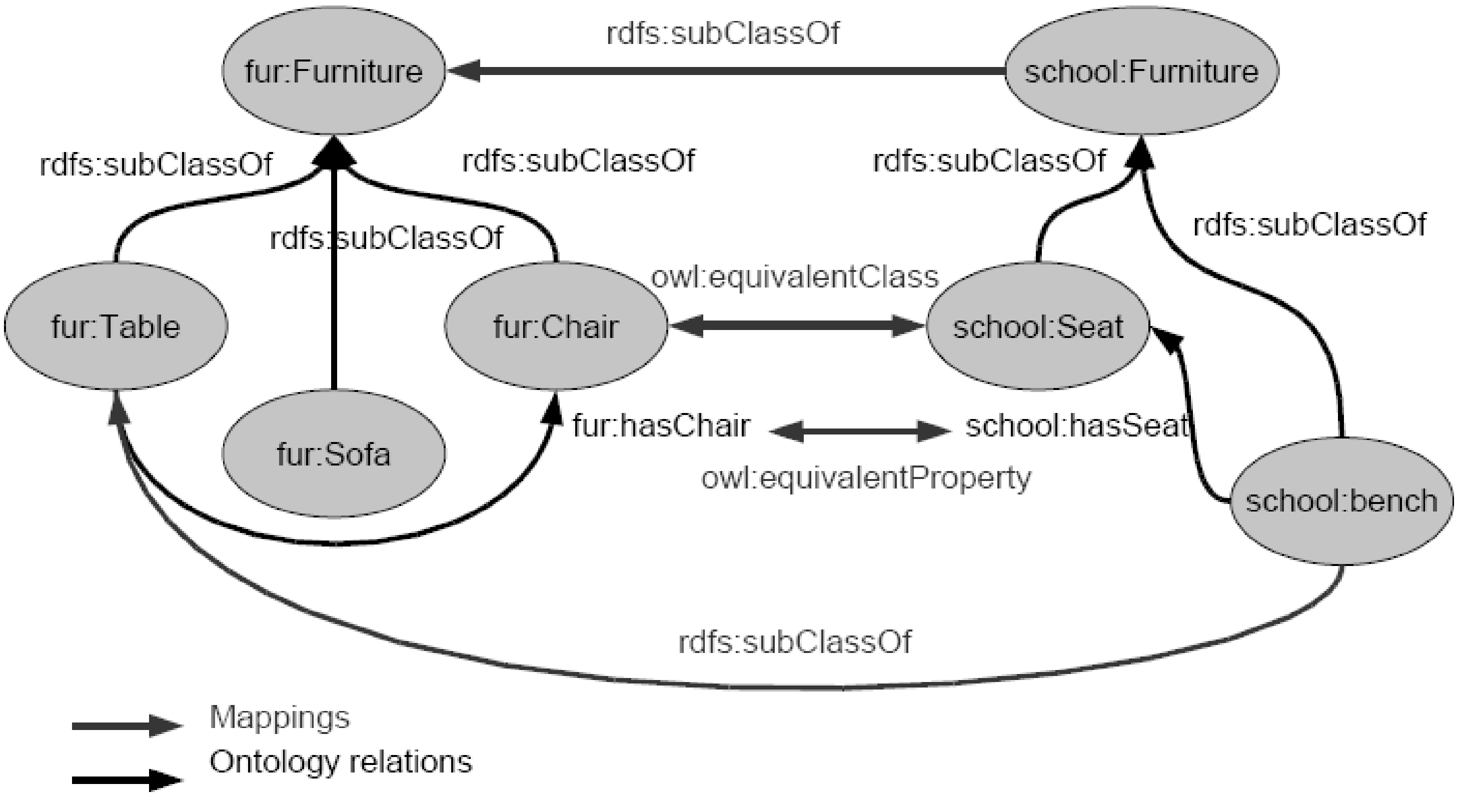
Similarity operator properties

- $sim(x,y) \in [0..1]$
- $sim(x,y) = 1 \leftrightarrow x = y$
- $sim(x,y) = 0 \leftrightarrow x \neq y$
- $sim(x,x) = 1$ (sim is reflexive)
- $sim(x,y) = sim(y,x)$ (sim is symmetric)
- $sim(x,z) \leq sim(x,y) + sim(y,z)$ (The triangular inequation holds)

Ontology mapping

- The process of relating similar concepts or relations of two or more information sources using equivalence relations or order relations.
- These relations are commonly implemented in inference and reasoning softwares, so we can use the output ontology to perform complex tasks on them without extra effort.

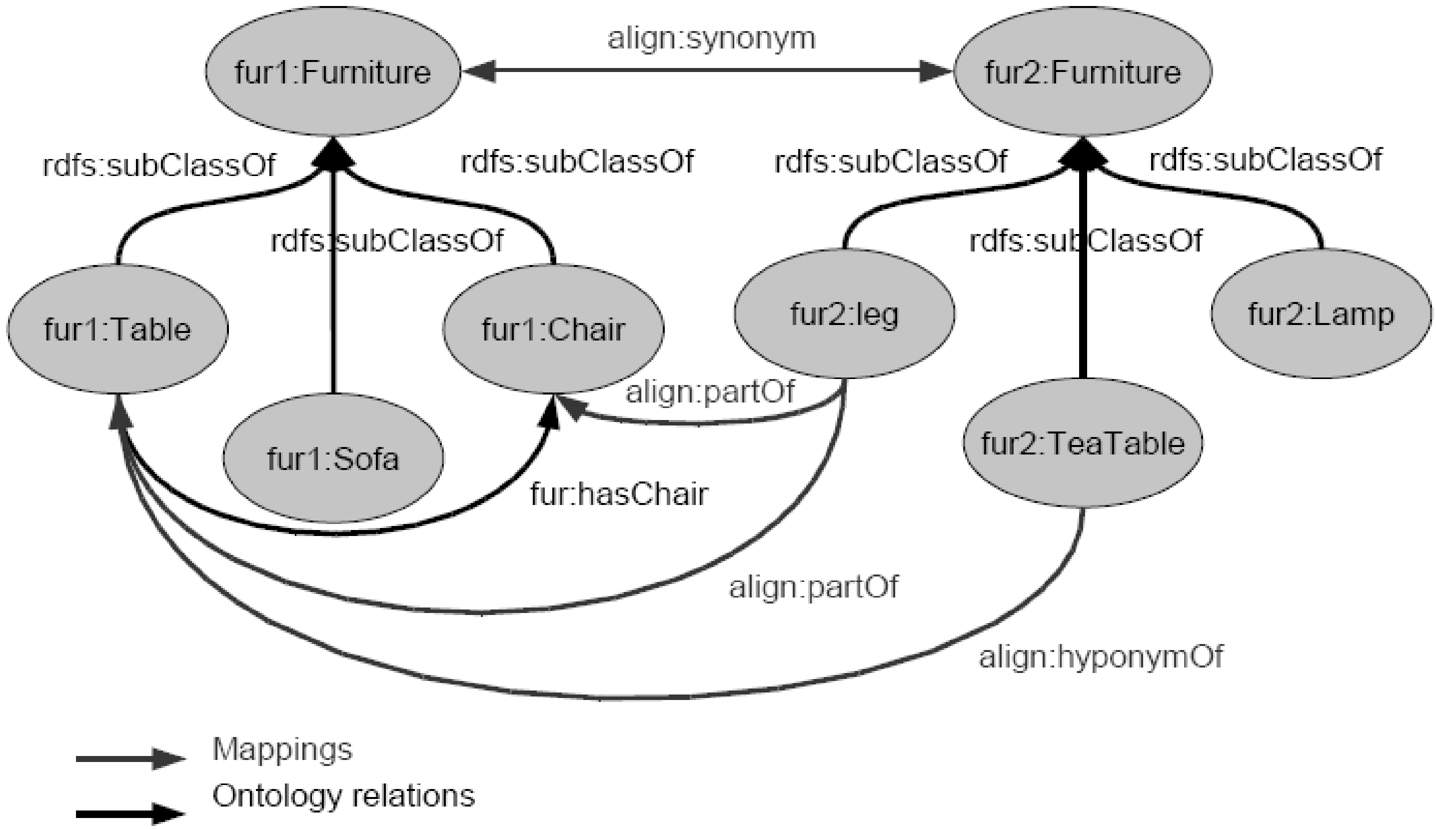
Ontology mapping



Ontology Aligning

- The process of bringing two or more ontologies into mutual agreement, making them consistent and coherent.
- Similar to ontology mapping but makes use of more expressive relations between ontologies concepts (partOf, subsumes, etc.).
- A great problem of this technique is that we must use an extended reasoner that can handle these relations, not commonly present in commercial softwares.

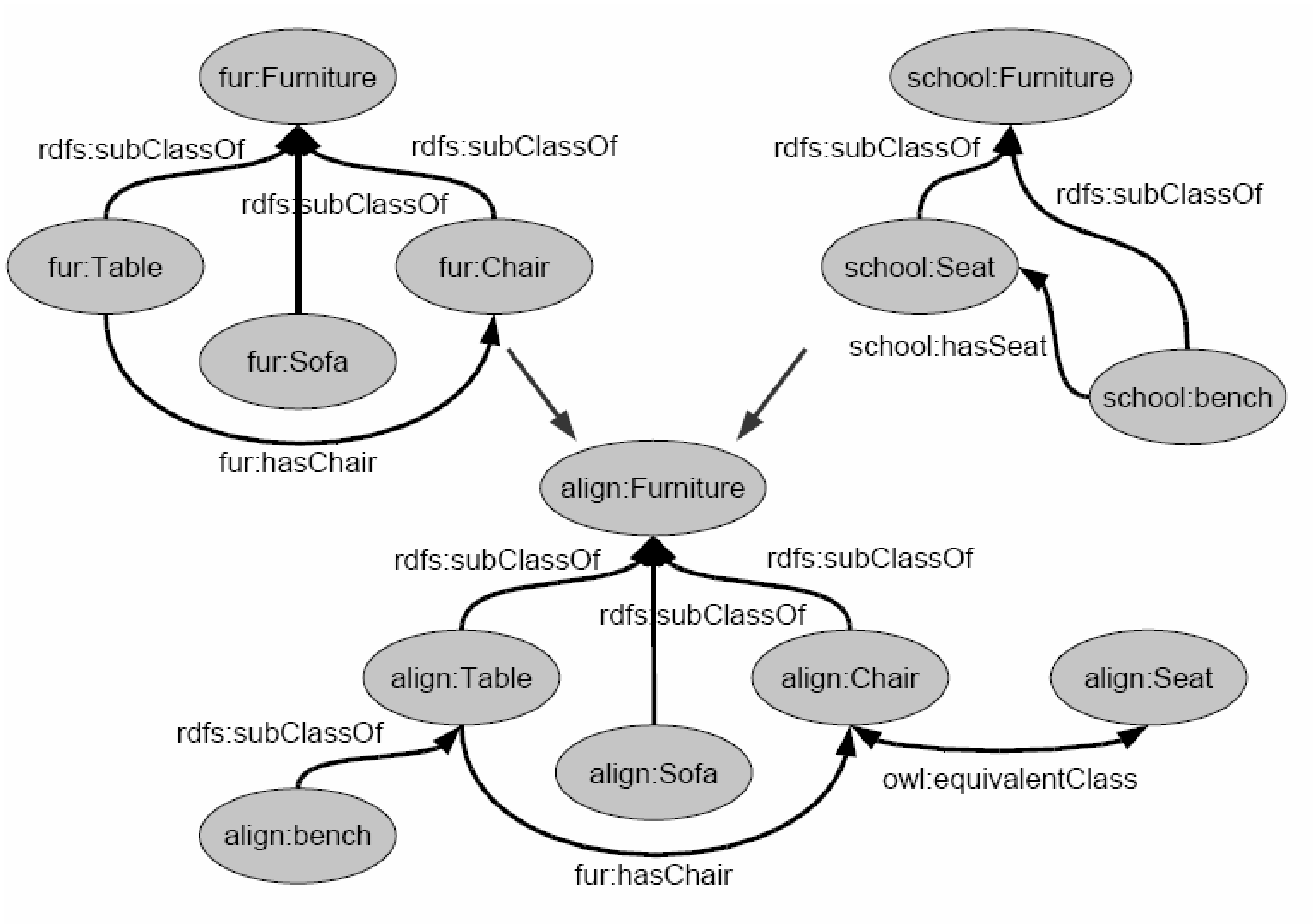
Ontology Aligning



Ontology merging

- The process of creating one ontology from two or more source ontologies with overlapping concepts or definitions.
- In the merging process the merged ontology is created from scratch, unifying all the source ontologies.
- In Ontology Merging there is no need for any reasoning software extensions because we reuse parts of sources ontologies without introducing new relations.

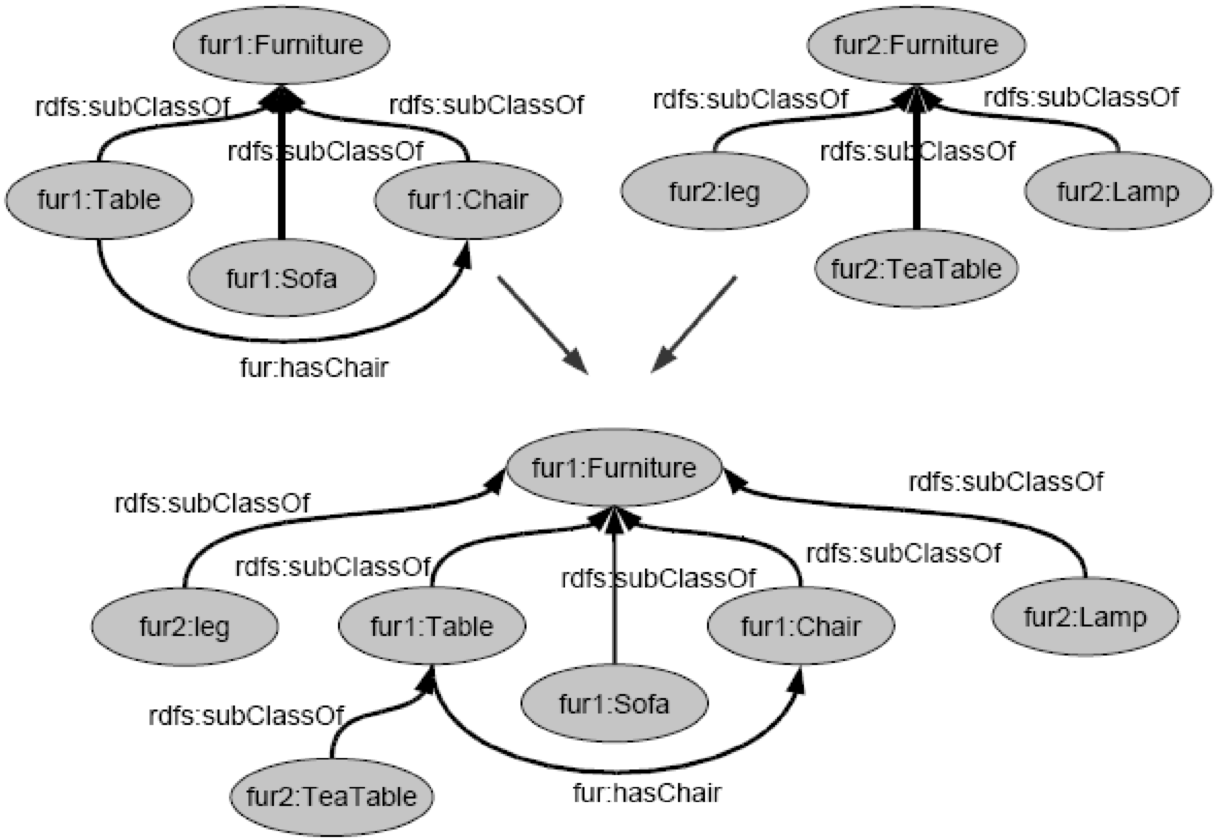
Ontology merging



Ontology integration

- Similar to Ontology Merging, but here the integrated ontology is created reusing parts of source ontologies as they are.
- A key task in Ontology Integration and Ontology Merging is the *consistency checking* that must ensure the absence of unforeseen or wrong implications into the merged ontology.

Ontology integration



Reasons for ontology mismatches

At the definition language level:

- Syntax
- Availability of different constructs (e.g. part-of, synonym, etc.)
- Linguistic primitives' semantics (e.g. union or intersection of multiple intervals)

→ Normalize by translating to the same language/ paradigm

Reasons for ontology mismatches

At the ontology level:

- **Scope: Two classes seem to represent the same concept, but do not have exactly the same instances**

- **Model coverage and granularity: a mismatch in the part of the domain that is covered by the ontology, or the level of detail to which that domain is modelled.**

- **Paradigm: Different paradigms can be used to represent concepts such as time. For example, one model might use temporal representations based on continuous intervals while another might use a representation based on discrete sets of time points.**

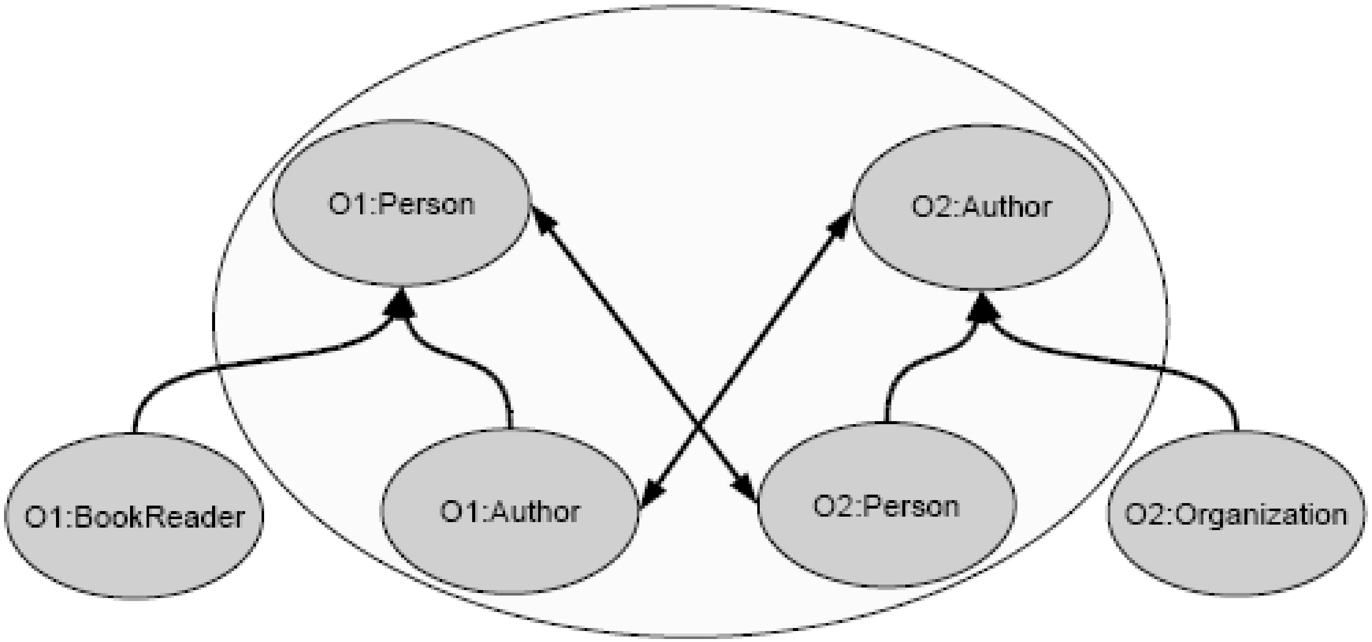
- **Encoding**

Reasons for ontology mismatches

At the ontology level:

- **Concept description: e.g. a distinctions between two classes can be modeled using a qualifying attribute or by introducing a separate class, or the way in which is-a hierarchy is built**
- **Homonymies**
- **Synonymies**

The bowtie inconsistency



How can ontologies support integration?

An ontology instead of a global schema:

- Intensional-level representation only in terms of ontologies
- Ontology mapping, merging, etc. insetad of schema integration
- Integrated ontology used as a schema for querying

How can ontologies support integration?

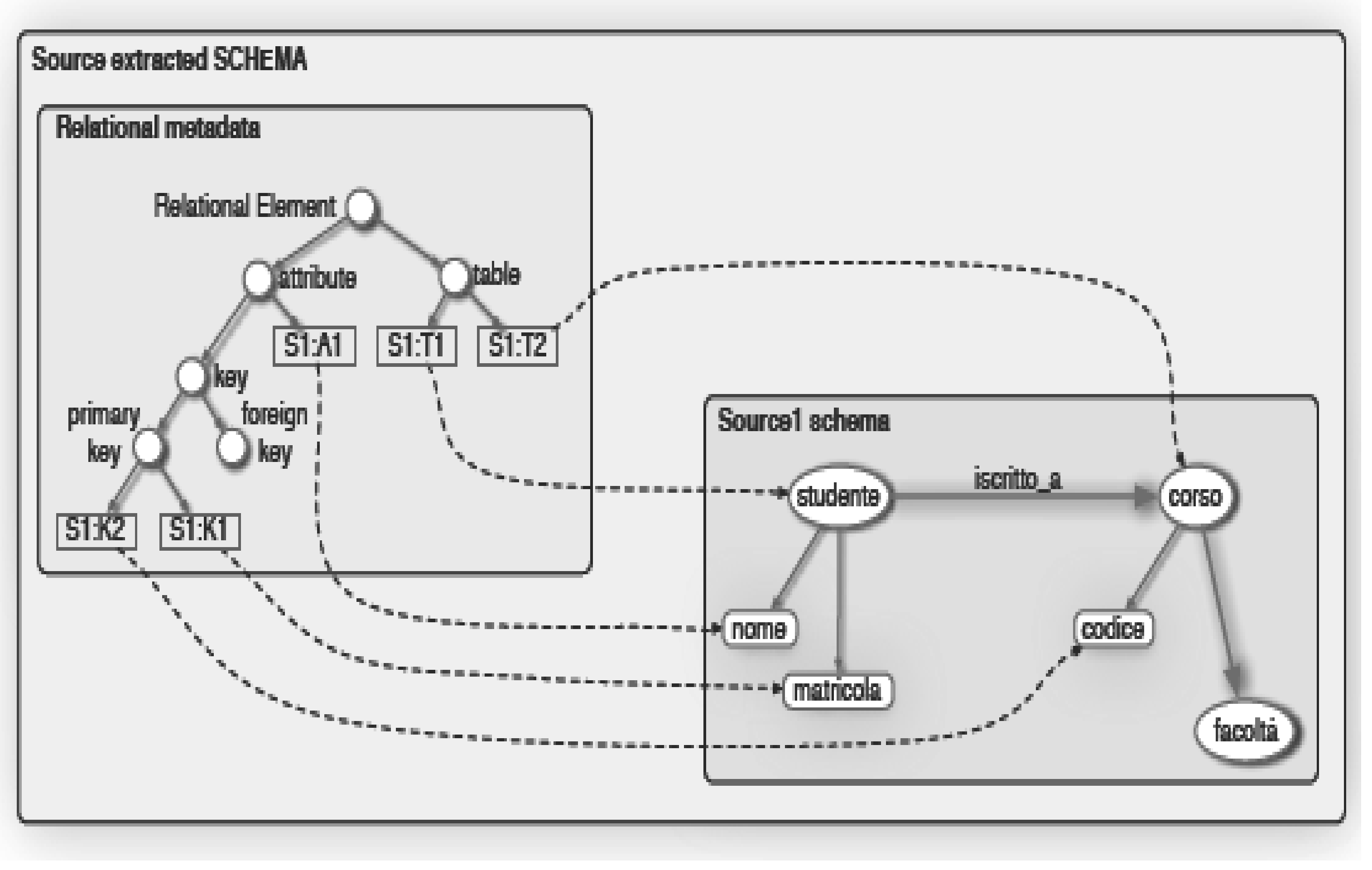
An ontology as a schema integration support tool

- Ontologies used to represent the semantics of schema elements (if the schema exists)
- Similarities between the source ontologies guide conflict resolution
  - At the schema level (if the schemata exist)
  - At the instance level

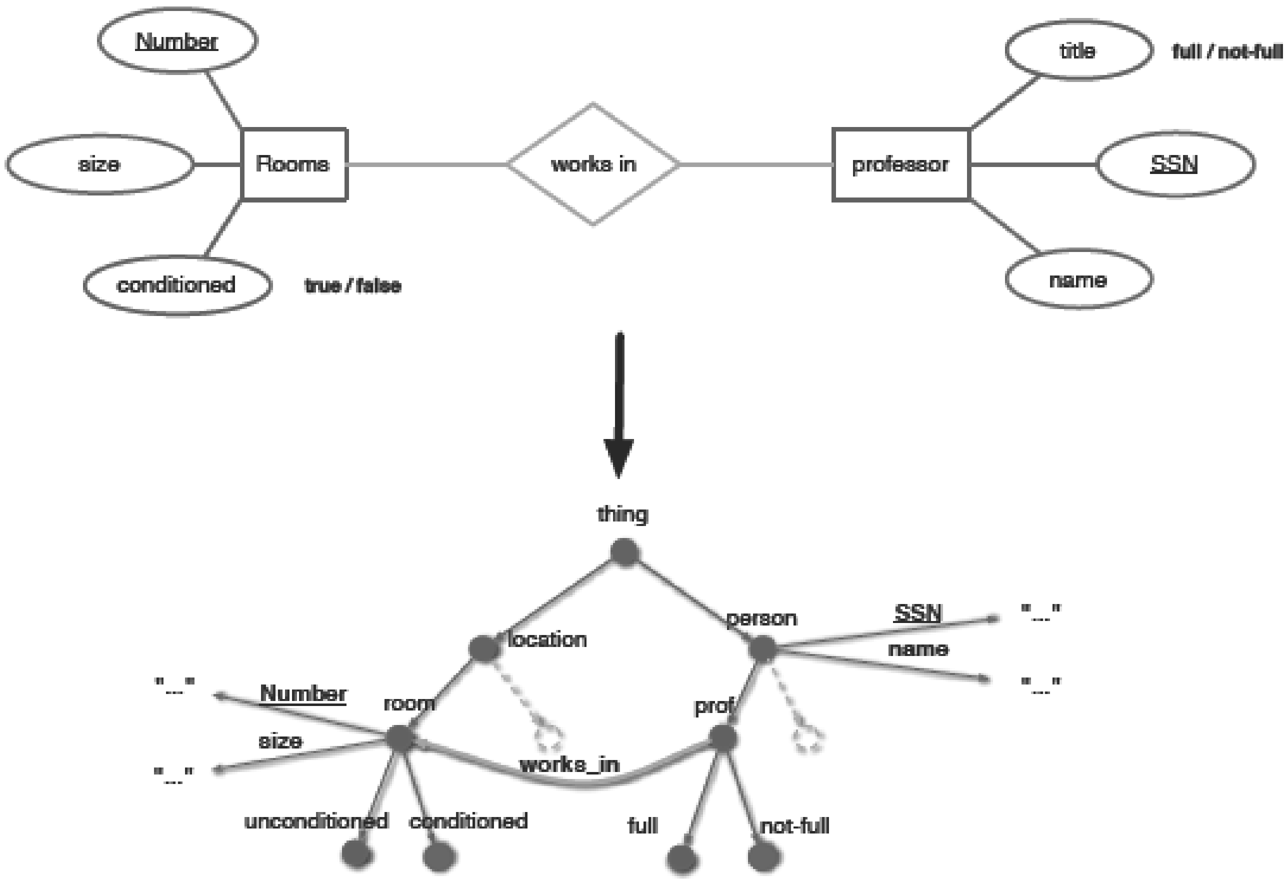
How can ontologies support integration?

- An ontology as a support tool for content interpretation and wrapping (e.g. HTML pages)
- An ontology as a support tool for content inconsistency detection and resolution

Ontology extraction from a relational schema



Ontology extraction from a ER schema



Query processing

Ontologies require query languages allowing

- Schema exploration
- Reasoning on the schema
- Instance querying (where does the instance sit?)

Query processing when instances are kept in a database

- Transformation of ontological query into the language of the datasource, and the other way round
- Different semantics (CWA versus OWA)
- What has to be processed where (e.g. push of the relational operators to the relational engine)