POLITECNICO
MILANO 1863

Semistructured data integration

Cinzia Cappiello
A.A. 2023-2024

These slides are based on Prof. Tanca slides.

POLITECNICO
MILANO 1863

First part – mediators and wrappers

Again recall the new application context

- A (possibly large) number of data sources
- Heterogeneous data sources (use of wrappers)
- ➢ 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
- Time-variant data (e.g., WEB)
- Mobile, transient data sources

3

SEMISTRUCTURED DATA

FOR THIS DATA THERE IS SOME FORM OF STRUCTURE, BUT IT IS NOT AS

- PRESCRIPTIVE
- REGULAR
- COMPLETE

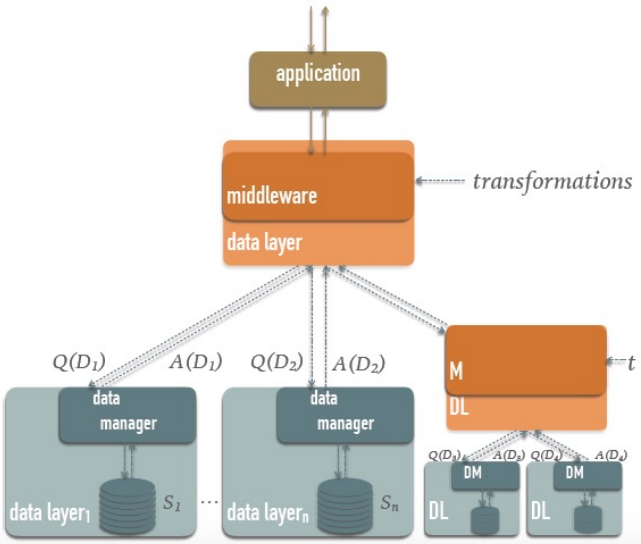
AS IN TRADITIONAL DBMSs

EXAMPLES

- WEB DATA
- XML DATA
- BUT ALSO DATA DERIVED FROM THE INTEGRATION OF HETEROGENEOUS DATASOURCES

4

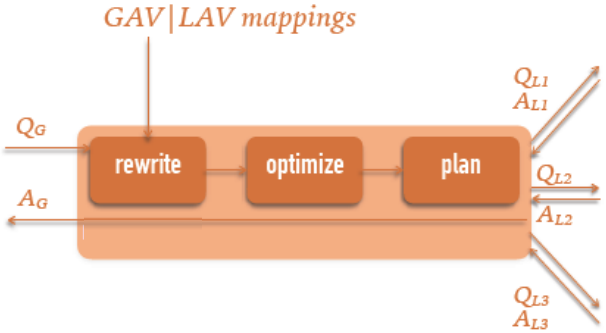
Our general framework for Data Integration



(Alvaro Fernandez, EDBT Summ. School 2017)

5

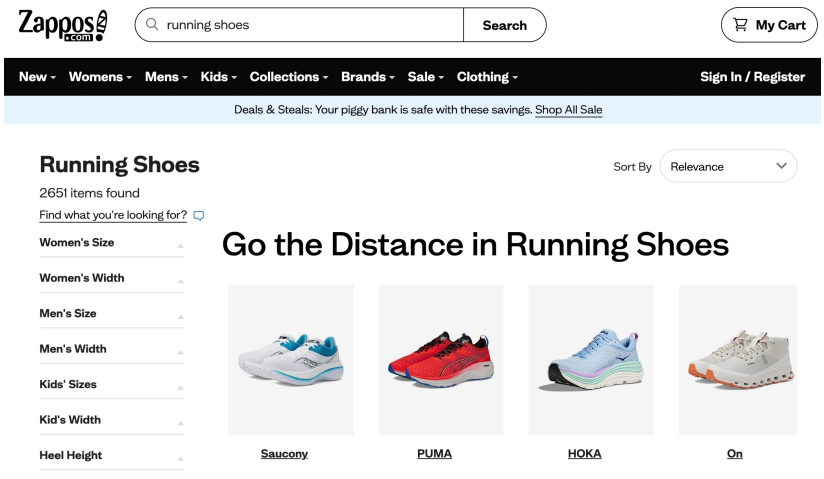
A closer look at the middleware



(Alvaro Fernandez, EDBT Summ. School 2017)

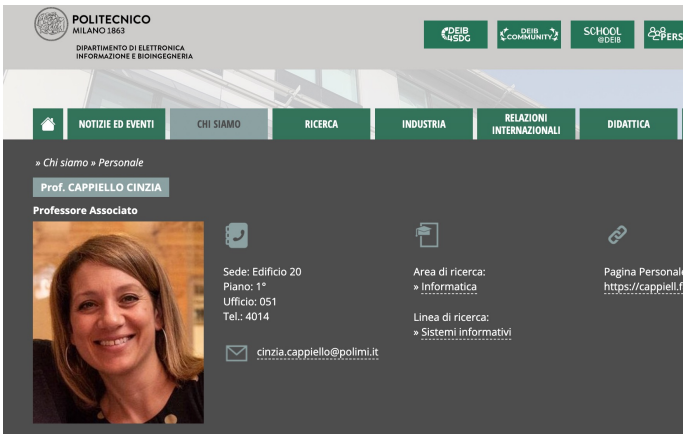
6

AN EXAMPLE OF SEMISTRUCTURED DATA



SEMISTRUCTURED DATA:

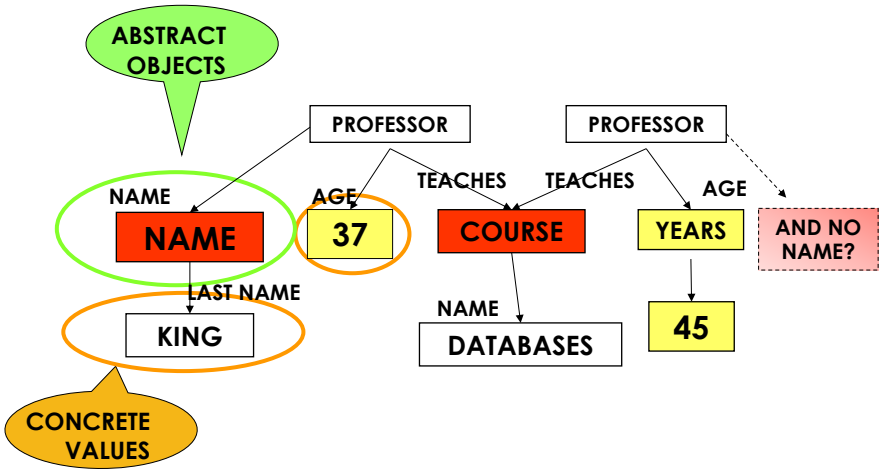
a page produced from a database



SEMISTRUCTURED DATA MODELS

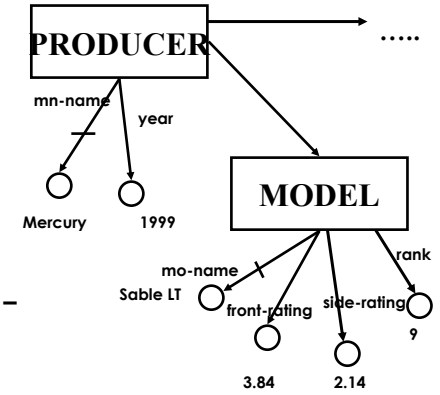
- BASED ON
  - TEXT
  - TREES
  - GRAPHS
    - LABELED NODES
    - LABELED ARCS
    - BOTH
- THEY ARE ALL DIFFERENT AND DO NOT LEND THEMSELVES TO EASY INTEGRATION

A GRAPH-BASED REPRESENTATION, WHERE THE IRREGULAR DATA STRUCTURE APPEARS VERY CLEARLY



A SIMPLE XML DOCUMENT WITH ITS GRAPH BASED REPRESENTATION

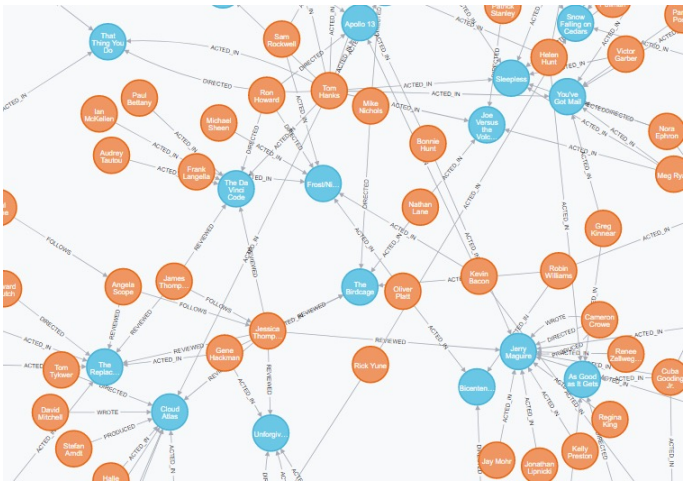
```
<producer>
 <mn-name>Mercury</mn-
  name>
 <year>1999</year>
 <model>
 <mo-name>Sable LT</mo-
  name>
 <front-
  rating>3.84</front-
  rating>
 <side-rating>2.14</side-
  rating>
 <rank>9</rank>
 </model>
 .........
</producer>
```



11

A Graph-based model:

Neo4j



12

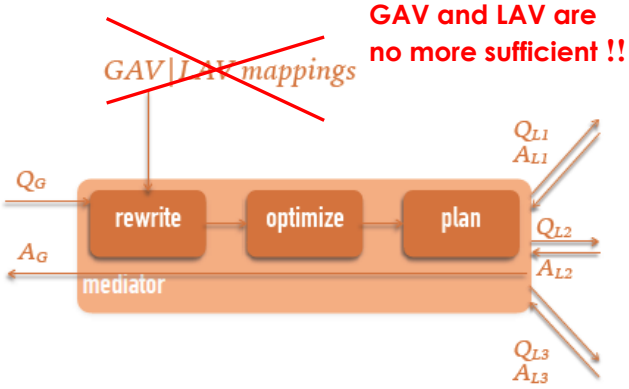
12

INFORMATION SEARCH IN SEMISTRUCTURED DATABASES

- WE WOULD LIKE TO:
  - INTEGRATE
  - QUERY
  - COMPARE

  DATA WITH DIFFERENT STRUCTURES ALSO WITH SEMISTRUCTURED DATA, JUST AS IF THEY WERE ALL STRUCTURED
- AN OVERALL DATA REPRESENTATION SHOULD BE PROGRESSIVELY BUILT, AS WE DISCOVER AND EXPLORE NEW INFORMATION SOURCES

MEDIATORS



A **mediator** must do the same as the integration systems seen up to now, but this time the problem is much more complex

(Alvaro Fernandez, EDBT Summ. School 2017)

MEDIATORS must do many different things

The term mediation includes:

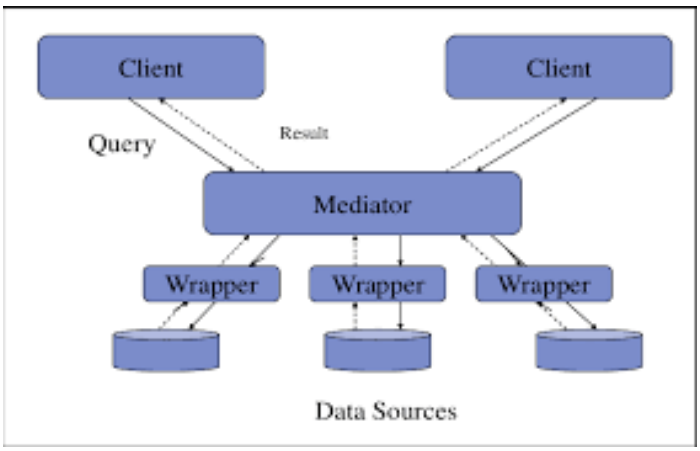
- the processing needed to make the interfaces work
- the knowledge structures that drive the transformations needed to transform data to information
- any intermediate storage that is needed (Wiederhold)

Problem:

each different domain needs a mediator appropriately designed to "understand" its semantics

15

Mediation-based systems

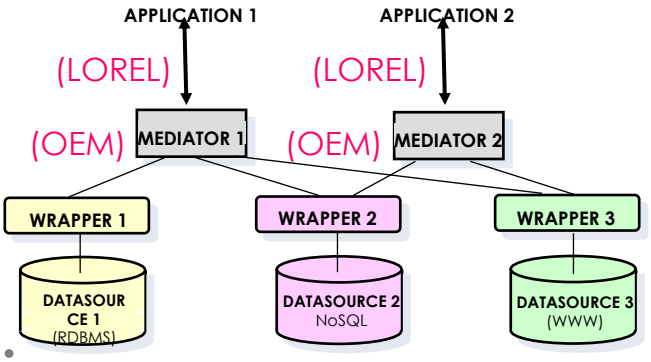


16

16

EXAMPLE: TSIMMIS

- first system based on the **mediator/wrapper paradigm**
- Proposed already in the 90's at Stanford university



Mediator-based approach

IN TSIMMIS:

- UNIQUE, GRAPH-BASED INTERNAL DATA MODEL: OEM (Object Exchange Model), MANAGED BY THE MEDIATOR
- WRAPPERS FOR THE MODEL-TO-MODEL TRANSLATIONS
- QUERY POSED TO THE MEDIATOR IN THE *LOREL (Lightweight Object REpository Language)* LANGUAGE
- MEDIATOR "KNOWS" THE SEMANTICS OF THE APPLICATION DOMAIN

OEM (Object Exchange Model) (TSIMMIS)

- Graph-based
- It does not represent the schema
- It directly represents the data : self-descriptive

```
<temp-in-farenheit,int,80>
```

19

Object structure in OEM

<(Object-id),label,type,value>

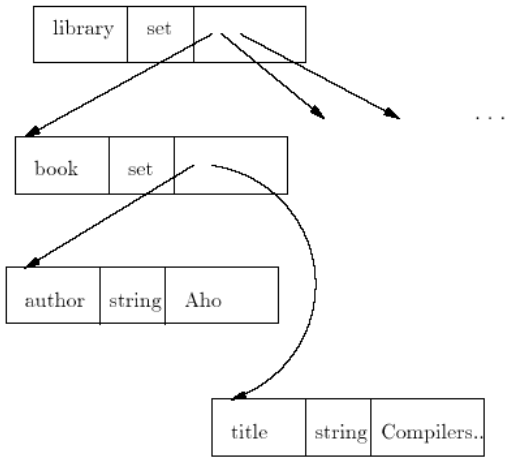
Nested structure

$\langle$set-of-temps, set, $\{cmp_1, cmp_2\}\rangle$
cmp_1: $\langle$temp-in-Fahrenheit, int, 80$\rangle$
cmp_2: $\langle$temp-in-Celsius, int, 20$\rangle$

20

20

OEM (Object Exchange Model) (TSIMMIS)



Typical complications when integrating semi- or un-structured data

- Each mediator is specialized into a certain domain (e.g. weather forecast), thus
- Each mediator must know domain metadata , which convey the data semantics
- On-line duplicate recognition, reconciliation and removal (no designer to solve conflicts at design time here)
- If data source changes a little, the wrapper has to be modified → automatic wrapper generation (later)

The language of TSIMMIS is *LOREL*

- *L*ightweight *O*bject *RE*pository *L*anguage
- Object-based
- Similar to object oriented query languages, with some modifications appropriate for semistructured data:

" Find books authored by Aho"

```
select library.book.title
where library.book.author = "Aho"
from library
```

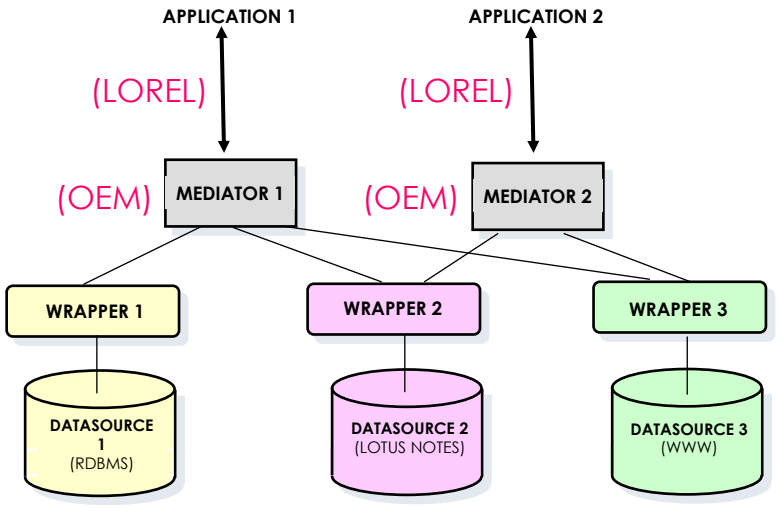

Query formulation in the Lorel language

```
select library.book.title
where library.book.author = "Aho"
from library
```
(if more than one root is available)

OK, but if this query must be produced at run-time and there is no schema, how does the user (or the system, if a transformation has to be applied) know that a node *library* exists, which contains nodes *book*, which in turn contain the fields *author* and *title ?*

- The TSIMMIS system introduced the *Dataguide*: a kind of a-posteriori schema, progressively built by the Mediator while exploring the data sources. Again, strictly bound to the application !!!

TSIMMIS SYSTEM



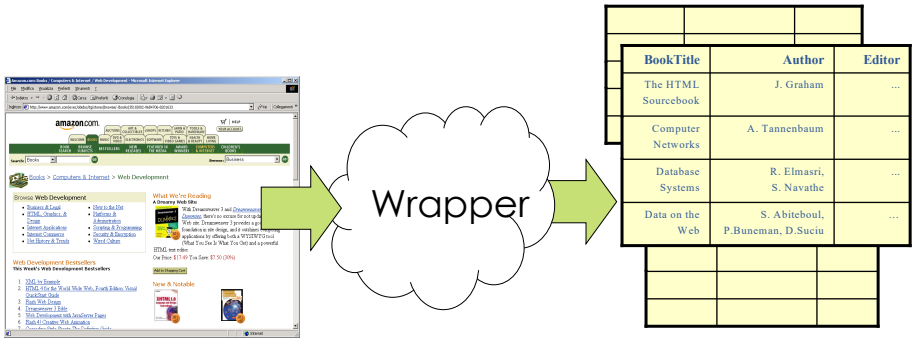
25

LET'S GO BACK TO THE CONCEPT OF WRAPPER

- Convert queries into queries/commands which are understandable for the specific data source
  - they can extend the query possibilities of a data source
- Convert query results from the source's format to a format which is understandable for the application

26

a WRAPPER for Web Pages



HTML page

database table(s) (or XML docs)

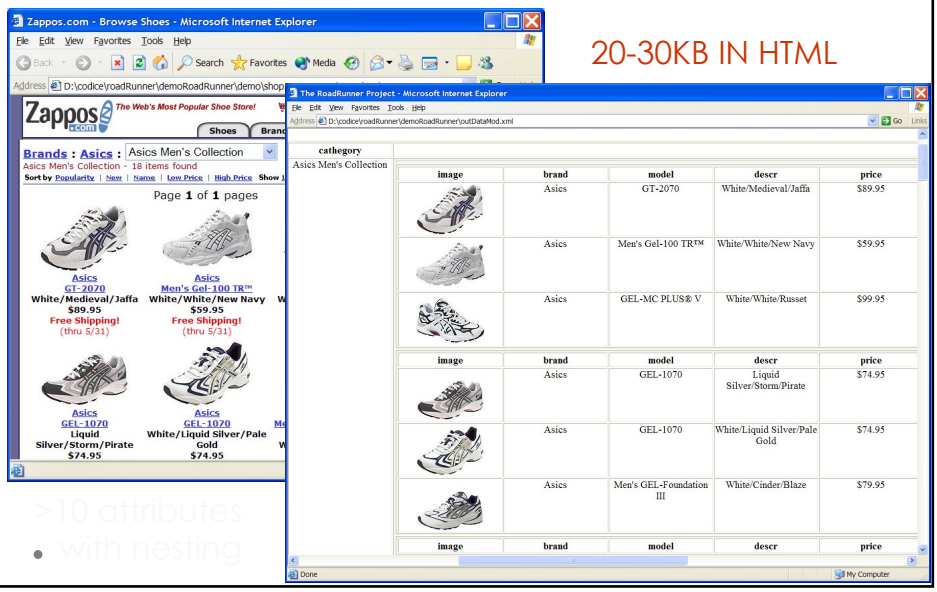
27

Estraction of information from HTML docs

- Information extraction
  - Source Format: plain text with HTML tags (no semantics)
  - Target Format: e.g., relational table (possibly nested, NF^2) or XML, JSON, etc.(we add ***structure***, i.e. ***semantics***)
  - Much easier if the underlying page structure is derived from a DB
- Wrapper
  - Software module that performs an ***extraction step***
  - Intuition: use extraction rules which exploit the ***marking tags***

28

A complex extraction process



20-30KB IN HTML

- >10 attributes
- with nesting

29

Problems

- Web sites change very frequently
- A layout change may affect the extraction rules
- Human-based maintenance of an ad-hoc wrapper is very expensive
- Better: ***automatic wrapper generation***

30

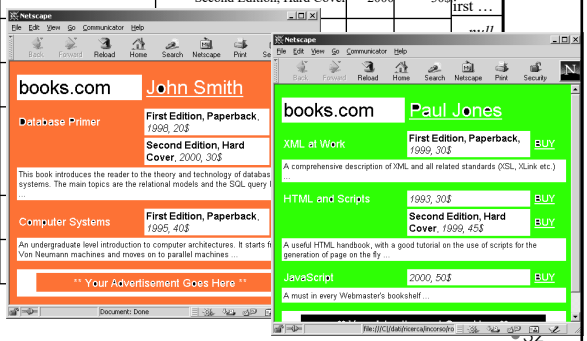
Automatic wrapper generation

- We can only use it when pages are *regular* to some extent
- OK when:
  - Many pages sharing the same structure
  - e.g. pages are dynamically generated from a DB

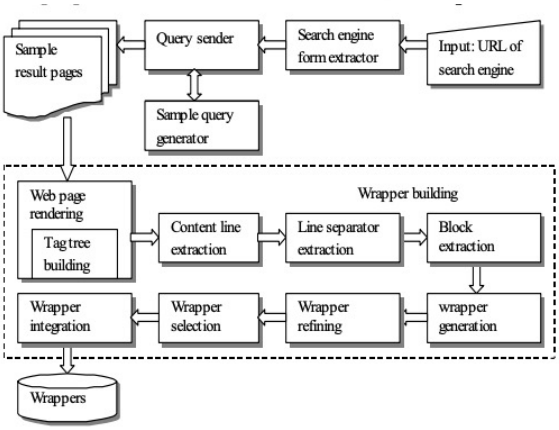
→ *data intensive* web sites

Online library

Name	Books				
	Title	Description	Editions		
			Details	Year	Price
John Smith	Database Primer	This book …	First Edition, Paperback	1998	20$
			Second Edition, Hard Cover	2000	30$
	Computer Systems				
Paul Jones	XML at Work				
	HTML and Scripts				
	JavaScripts				
…	…				



An example



Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, and Clement Yu. 2005. Fully automatic wrapper generation for search engines. In Proceedings of the 14th international conference on World Wide Web (WWW '05). Association for Computing Machinery, New York, NY, USA, 66-75. https://doi.org/10.1145/1060745.1060760

34

Bibliography

- A. Doan, A. Halevy and Z. Ives, Principles of Data Integration, Morgan Kaufmann, 2012
- L. Dong, D. Srivastava, Big Data Integration, Morgan & Claypool Publishers, 2015
- Roberto De Virgilio, Fausto Giunchiglia, Letizia Tanca (Eds.): Semantic Web Information Management – A Model-Based Perspective. Springer 2009, ISBN 978-3-642-04328-4
- M. Lenzerini, Data Integration: A Theoretical Perspective, Proceedings of ACM PODS, pp. 233-246, ACM, 2002, ISBN: 1-58113-507-6
- Clement T. Yu, Weiyi Meng, Principles of Database Query Processing for Advanced Applications , Morgan Kaufmann, 1998, ISBN: 1558604340

40

POLITECNICO
MILANO 1863

Second part – ontologies

41

Ontologies: a way to solve the problem of automatic semantic matching

- 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*)

→ an ER diagram, a class diagram, any conceptual schema *is a kind of ontology!*

42

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

43

Aims…

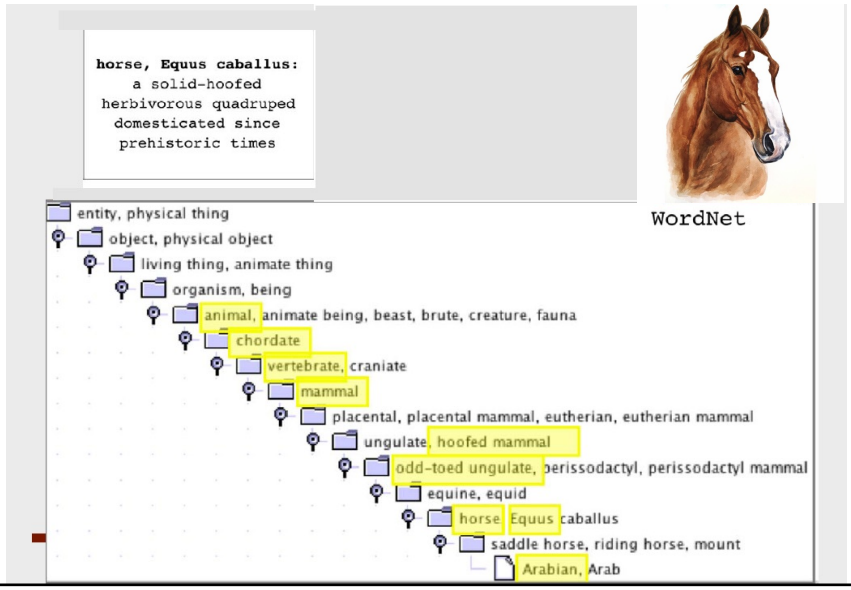
- A formal specification allows for use of a common vocabulary for *automatic knowledge sharing*
- Formally specifying a *conceptualization* means giving a unique meaning to the terms that define the knowledge about a given domain
- *Shared:* an ontology captures knowledge which is common, thus *over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

44

Ontology types

- **Taxonomic ontologies**
  - Definition of concepts through terms, their hierarchical organization, and additional (pre-defined) 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



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
- Relationships between concepts:
  - Taxonomies (IS_A),
  - Meronymies (PART_OF),
  - Synonymies, homonymies, ...
  - User-defined associations,

47

Formal Definitions

O = (C, R, I, A)

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

I: Instances

- Specified in some logic-based language
- Organized in a ISA hierarchy
- *I is an* instance collection, stored in the information source

48

Formal Definitions

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

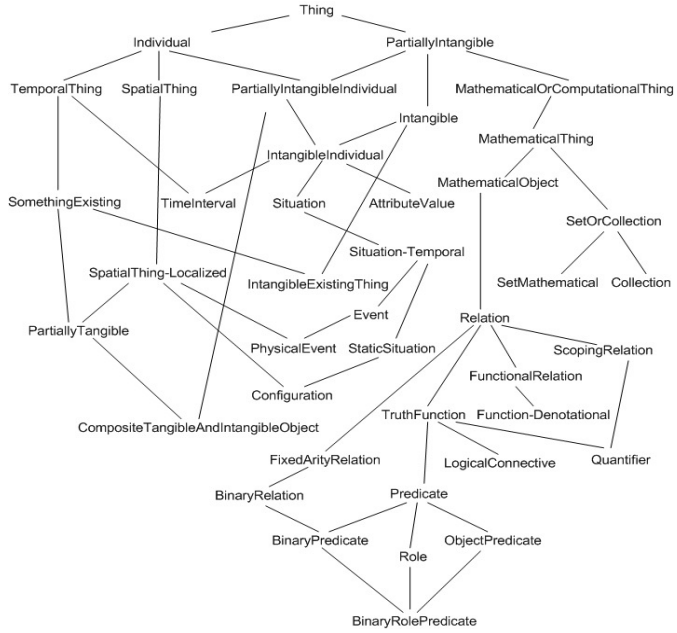
- a ***T-Box****:* contains all the concept and role definitions, and also contains all the axioms of our *logical theory* (e.g. "A father is a Man with a Child").

- 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)). It describes the *instances.*

OpenCyc

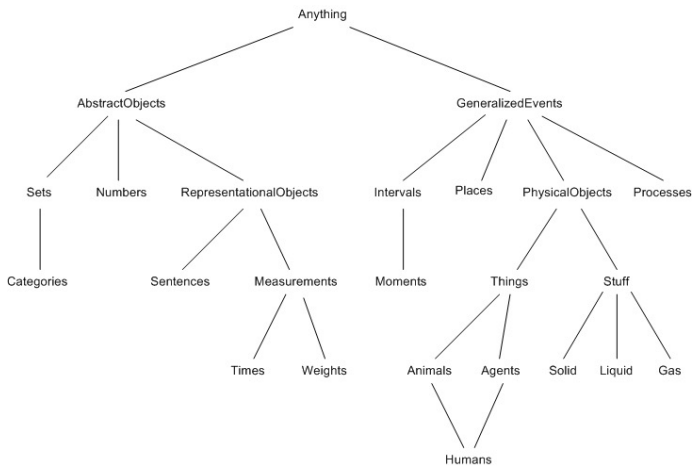
- The open source version of the Cyc technology, started in 1984 at MCC.
- Available until early 2017 as OpenCyc under an open source (Apache) license.
- Later, Cyc was made available to AI researchers under a research-purpose license as ResearchCyc.
- Cyc is a long-term artificial intelligence project that aims to assemble a comprehensive ontology and knowledge base that spans the basic concepts and rules about how the world works.
- The entire Cyc ontology contains hundreds of thousands of terms and millions of assertions relating the terms to each other, forming an ontology whose domain is all of human consensus reality.

Top level concepts of Cyc



51

Top level concepts of the Russel and Norvig ontology



52

semantic interoperability → *Semantic Web*

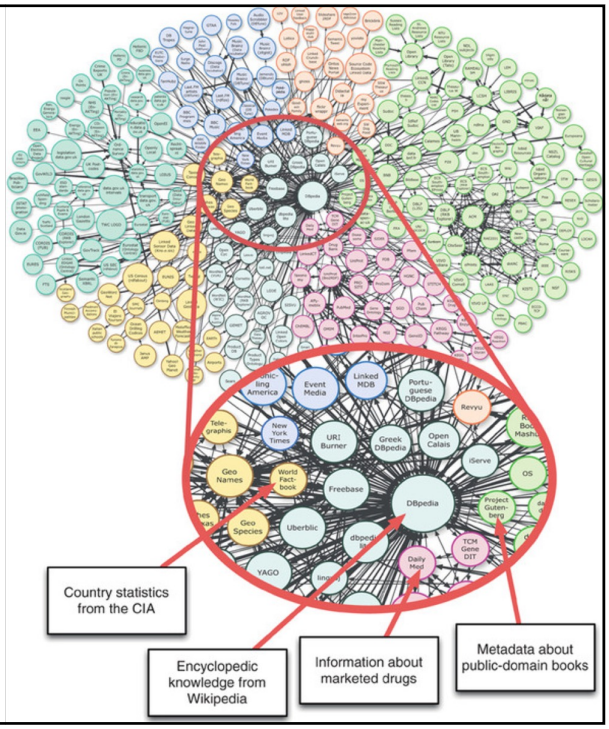
- A vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.
- Built on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data(*).
- 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.

(*) Also different implementations of RDF exist, not based on XML (e.g. Turtle)

Linked Data

- Linked Data is a W3C-backed movement about connecting data sets across the Web. It describes a method of publishing structured data so that it can be interlinked and become more useful.
- It builds upon standard Web technologies such as HTTP, RDF and URIs, but extends them to share information in a way that can be read automatically by computers, enabling data from different sources to be connected and queried.
- A subset of the wider Semantic Web movement, which is about adding meaning to the Web (Tim Berners-Lee)
- Open Data describes data that has been uploaded to the Web and is accessible to all
- Linked Open Data: extend the Web with a data commons by publishing various open datasets as RDF on the Web and by setting RDF links among them

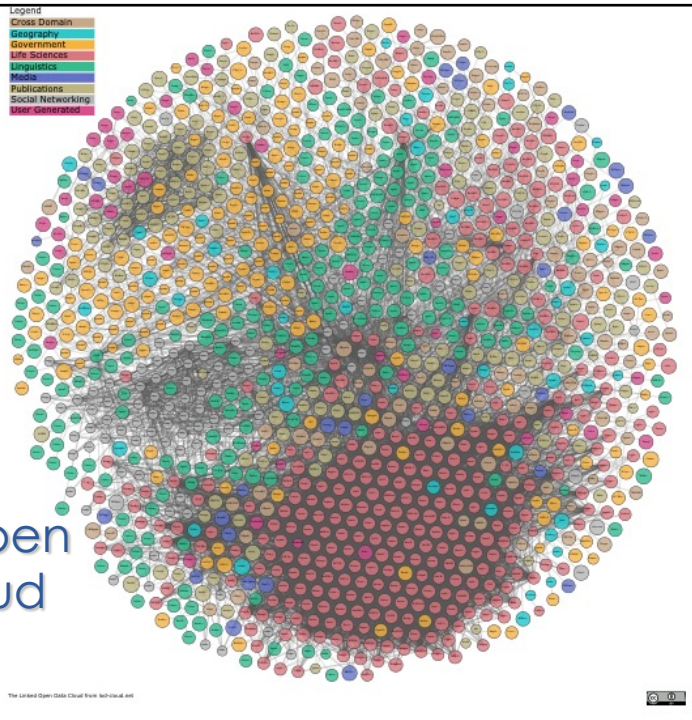
Linked Open Data



55

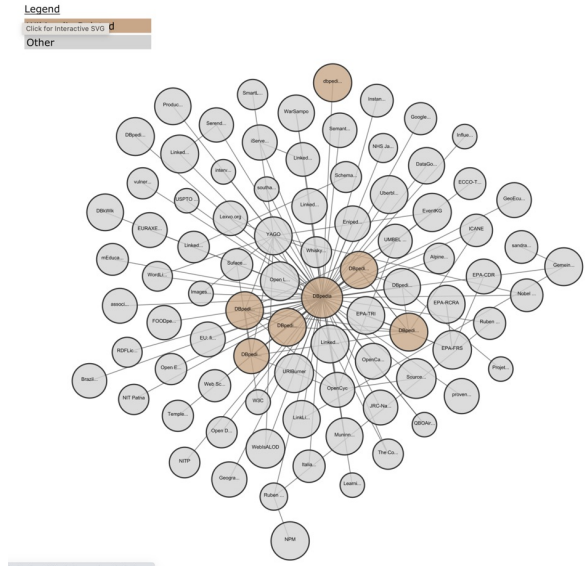
Linked Open Data Cloud Diagram

https://lod-cloud.net/



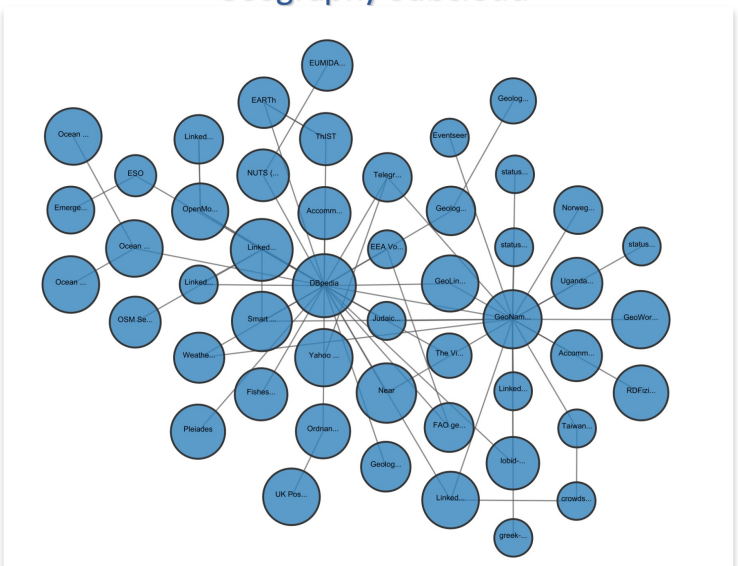
56

Cross-Domain Subcloud



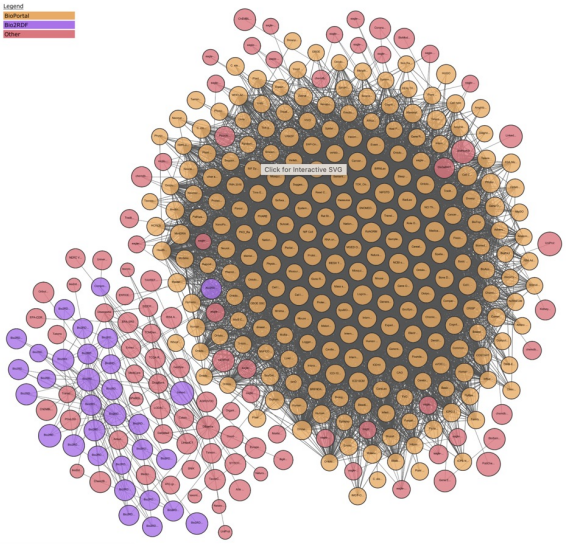
57

Geography Subcloud



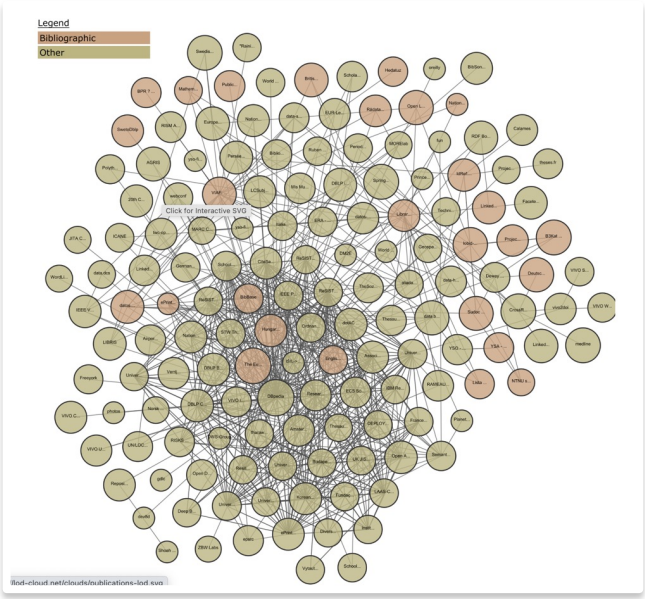
58

Life Sciences Subcloud



59

Publications Subcloud



60

Some famous datasets

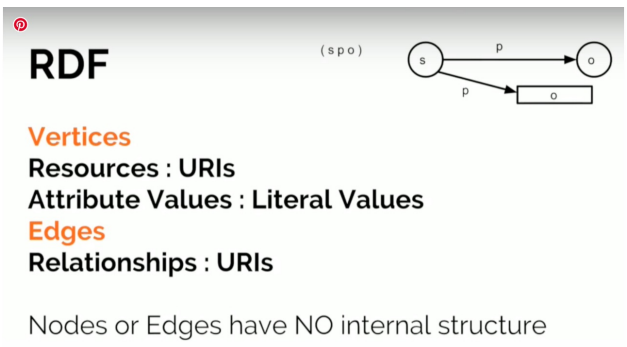
- CKAN – registry of open data and content packages provided by the Open Knowledge Foundation
- DBpedia – a dataset containing data extracted from Wikipedia; it contains about 4 million concepts described by some billion triples, including abstracts in 11 different languages
- GeoNames provides RDF descriptions of more than 7,500,000 geographical features worldwide.
- YAGO (Yet Another Great Ontology) is an ever-growing open source knowledge base developed at the Max Planck Institute for Computer Science in Saarbrücken. It is automatically extracted from Wikipedia and other sources.
- UMBEL – a lightweight reference structure of 20,000 subject concept classes and their relationships derived from OpenCyc, which can act as binding classes to external data; also has links to 1.5 million named entities from DBpedia and YAGO
- FOAF – a dataset describing persons, their properties and relationships

61

RDF

At the core of RDF is this notion of a triple **subject-predicate-object,** a statement that represents two vertices connected by an edge:

- **Subject:** a resource, or a node in the graph
- **Predicate:** an edge – a relationship
- **Object:** another node or a literal value

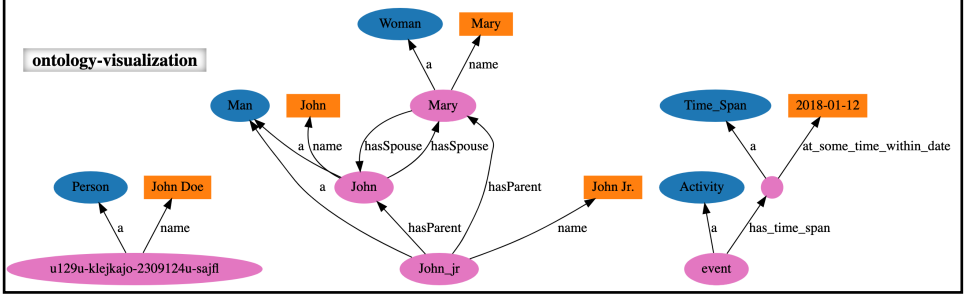


62

62

```
:John a :Man ;
    :name "John" ;
    :hasSpouse :Mary .
:Mary a :Woman ;
    :name "Mary" ;
    :hasSpouse :John .
:John_jr a :Man ;
    :name "John Jr." ;
    :hasParent :John, :Mary .
:Time_Span a owl:Class .
:event a :Activity ;
    :has_time_span [
        a :Time_Span ;
        :at_some_time_within_date "2018-01-12"^^xsd:date
        ] .
:u129u-klejkajo-2309124u-sajfl a :Person ;
    :name "John Doe" .
```

A sample RDF file



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>
```

RDF and 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

- 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.

Reasoning services for ontologies

Services for the Tbox

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

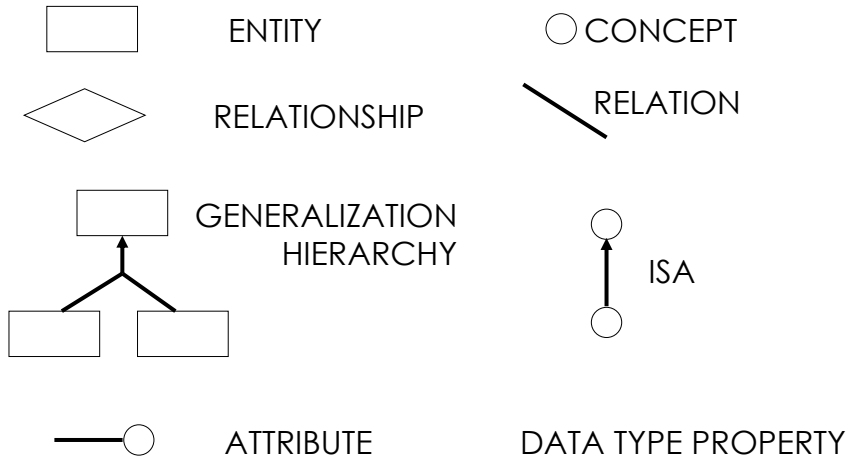
Services for the Abox

- 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

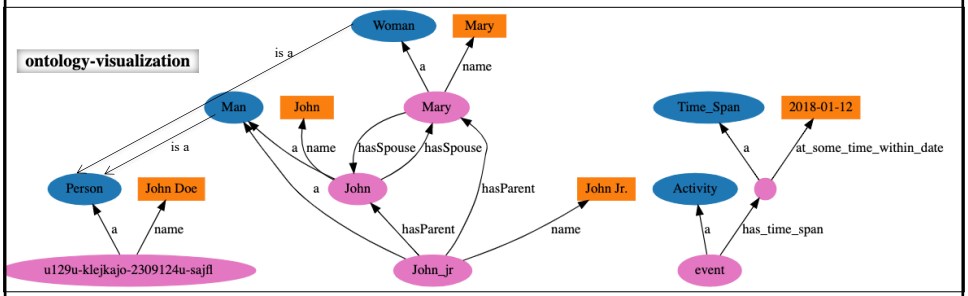
- analysis of the features of a descriptive ontology (data structures, instance management, constraint definition, queries)
- compare these features with the functionality provided by current representation approaches from the database world

ER vs.ontology



74

Let's see how this corresponds to an ER diagram



75 **Caution:** An ER schema does not have VALUES !

75

Comparison

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

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

7

DB versus ontologies

How should we improve database conceptual models to fulfill ontology requirements ?

- Supporting defined concepts and adding the necessary reasoning mechanisms
- Managing missing and incomplete information: semantic differences between the two assumptions made w.r.t. missing information (*Closed World Assumption* vs. *Open World Assumption*)
- Databases are assumed to represent certain data: a tuple in the database is true, any tuple NOT in the database is false (Closed World Assumption)

Ontologies and integration problems

- Discovery of "equivalent" concepts (mapping)
  - What does equivalent mean? → again we look for some kind of similarity
- 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*
- Again we need some kind of *similarity measure.*
- Recall: a *similarity value* is usually a number in the interval [0,1]
- *Caution:* this time the similarity measure takes into account *semantics*, not only on the structure of the words as in the examples given in the previous lectures !!!!

More on similarity

- The concept of similarity is a basic concept in human cognition.
- Similarity plays an essential role in taxonomy, recognition, case-based reasoning and many other fields. There are many aspects of the concept of similarity that have eluded formalization.
- According to Zadeh(*), "Formulation of a valid, general-purpose definition of similarity is a challenging problem".
- There do exist many special-purpose definitions which have been employed with success in cluster analysis, search, classification, recognition and diagnostics.

(*) The "inventor" of fuzzy sets and fuzzy logic

As already seen, *similarity* is strictly related to *distance*

Self-identity is the property which says that the distance between identical objects is zero. This translates to the following self-identity axiom:

Axiom 2.1.1. *For all x in S, $d(x, x) = 0$.*

Positivity is the property which says that distinct objects have a nonzero distance:

Axiom 2.1.2. *For all $x \neq y$ in S, $d(x, y) > 0$.*

Symmetry says that the order of two elements does not matter for the distance between them:

Axiom 2.1.3. *For all x and y in S, $d(x, y) = d(y, x)$.*

The *triangle inequality* says that the distance between y and z does not exceed the sum of the distance between y and x and the distance between x and z:

Axiom 2.1.4. *For all $x, y, z \in S$, $d(y, z) \leq d(y, x) + d(x, z)$.*

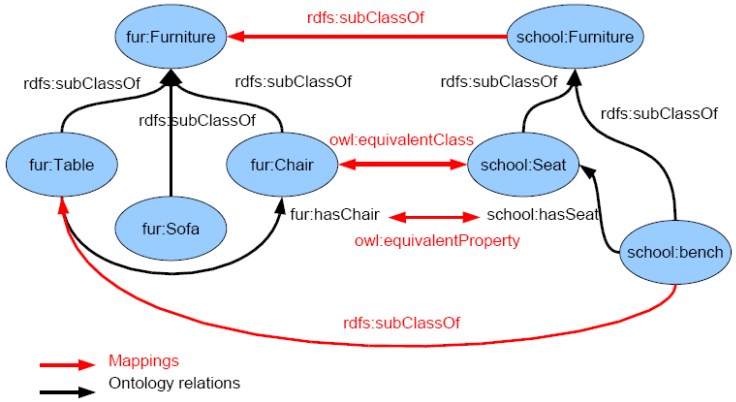
Similarity

- While dealing with distance-based similarity measures, examples have been constructed where every distance axiom is clearly violated by dissimilarity measures, and particularly *the triangle inequality*, consequently the corresponding similarity measure disobeys transitivity.
- For these cases a different attitude has been taken and more general concepts of distance have been proposed: a distinction is made between *perceived dissimilarity* and *judged dissimilarity.*

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



84

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

85

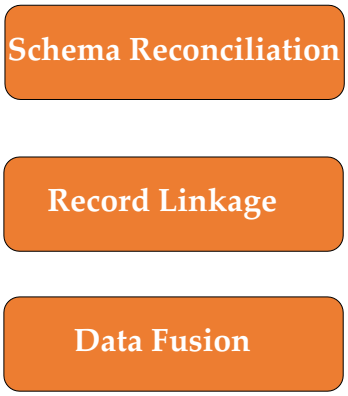
Reasons for ontology mismatches

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
- 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

86

Recall the steps of Data Integration



Schema reconciliation: mapping the **data structure**

Record linkage: data matching based on **the same content**

Data fusion: reconciliation of **non-identical content**

87

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 (record linkage)

An ontology instead of a global schema:

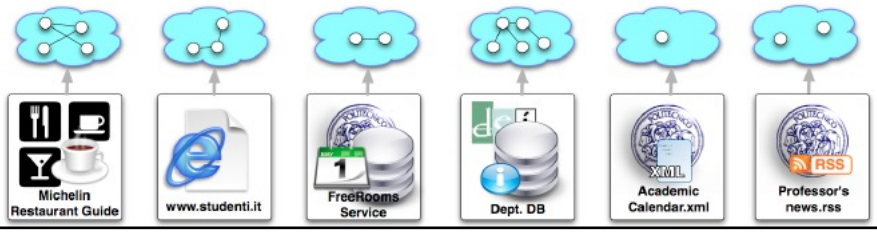
- Schema-level representation only in terms of ontologies
- Ontology mapping, merging, etc. instead of schema integration
- Integrated ontology used as a schema for querying

88

An ontology instead of a global schema

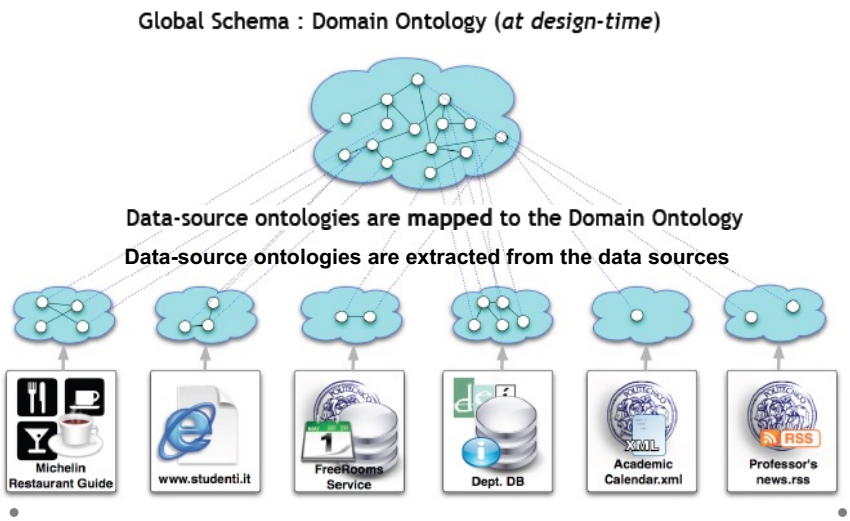
- Data-source heterogeneity is solved by extracting the semantics in an ontological format (potentially *at run-time*)
- Automatic Wrapper generation + Query translation will bridge among two models.
- Not an easy task:
  - several issues, e.g., impedance mismatch
  - unstructured data sources

Note:the impedance mismatch is the problem that occurs when the database model is different from the programming language model



89

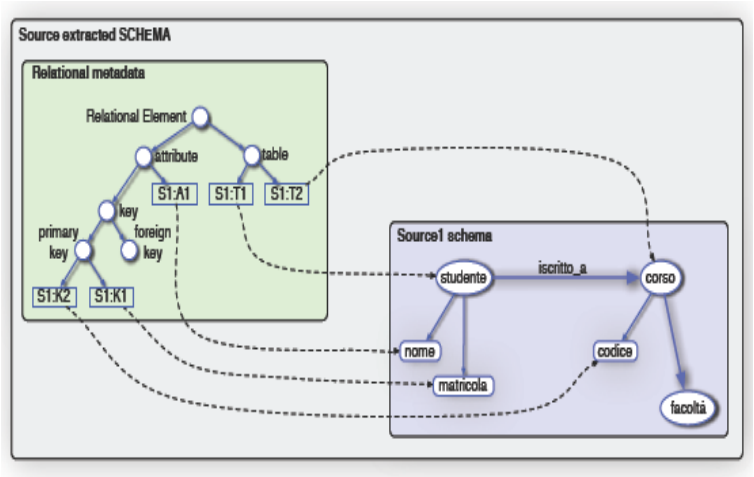
An ontology instead of a global schema



More: how can ontologies support integration?

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

Ontology extraction from a relational schema: example



Ontology query processing

Ontologies require query languages as well, for:

- Schema exploration (when the schema is replaced by an ontology)
- Reasoning on the schema
- Instance querying (when the instance is contained in an ontology, like in the Semantic Web case)
- Example of ontology query language: SPARQL (W3C)

Examples of SPARQL queries

We can give an informal idea of **SPARQL** focusing on the SELECT queries. In particular we can say that a query *q* is a structure like:
SELECT ?X1 … ?Xn WHERE P
With *?X1 … ?Xn* as the set of variables and *P* as the graph pattern.

Let's consider this **example** where we have a simple SPARQL query which asks for all projects in which some PhD student is involved:
SELECT ?Y
WHERE { ?X rdf:type PhDStudent. ?X inProject ?Y }

This next **example** of query instead retrieves employees who are PhD students or professors together with their projects:
SELECT ?X ?Y
WHERE { { ?X rdf:type PhDStudent. UNION ?X rdf:type Professor. }
AND ?X inProject ?Y. }

Ontology query processing versus database query processing

When we use ontologies to interact with databases, we have to take care of:

- 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)

The new application context (recall)

- A (possibly large) number of data sources
- Heterogeneous 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
- ➢ Time-variant data (e.g. WEB and social media)
- ➢ Mobile, transient data sources (e.g. sensor values)

…..as you can see, everything becomes more and more dynamic.

97

Next Lectures

From next lecture, we'll talk about the second important subject of this course:

Data Warehouses:
a fully consolidated paradigm for *business analytics*.

After that, we'll come back to the frontiers of Data Management and Integration, which at the moment still constitute hot research topics

98

98

Bibliography

- A. Doan, A. Halevy and Z. Ives, Principles of Data Integration, Morgan Kaufmann, 2012
- L. Dong, D. Srivastava, Big Data Integration, Morgan & Claypool Publishers, 2015
- Roberto De Virgilio, Fausto Giunchiglia, Letizia Tanca (Eds.): Semantic Web Information Management – A Model-Based Perspective. Springer 2009, ISBN 978-3-642-04328-4
- M. Lenzerini, Data Integration: A Theoretical Perspective, Proceedings of ACM PODS, pp. 233-246, ACM, 2002, ISBN: 1-58113-507-6
- Clement T. Yu, Weiyi Meng, Principles of Database Query Processing for Advanced Applications , Morgan Kaufmann, 1998, ISBN: 1558604340