The Semantic Web

Serge Abiteboul
INRIA Saclay, Collège de France, ENS Cachan

Organization

- Introduction
- Ontologies
- Querying ontologies
- Integrating data sources

Introduction

The goals

First step from a web of text (for humans) to a web of knowledge (for machines)

Attach semantics to information published on the web

- Improve precision of query results
- Facilitate integration of data sources

Difficulties

- Mismatch between structured and unstructured data
- Heterogeneity between data sources
- Imprecision, incompleteness, possibly inconsistencies of information

The semantic web

The **Semantic web** is an evolving extension of web standard to introduce semantics

Standards of the W3C:

- Naming entities: URI
- Facts/relations: RDF
- Constraints on them: RDF/S or OWL
- Linked data
- Queries: SPARQL

Uniform resource identifiers

The web talks about **resources**

A resource is anything on the Internet that can be referred to by a **Uniform Resource Identifier** (URI), i.e., a string of characters

- A web page, identified by a URL
- A fragment of an XML document
- A web service,
- A thing, an object, a concept, a property, etc.

Resources are described using **semantic annotations**: logical assertions that relate resources to some terms in pre-defined **ontologies**

Ontologies

Ontologies

Descriptions providing a shared understanding of a given domain

- A controlled vocabulary
- Understandable by **humans**
- Formally defined so that it can also be processed by **machines**
- Logical semantics to enablea **reasoning**

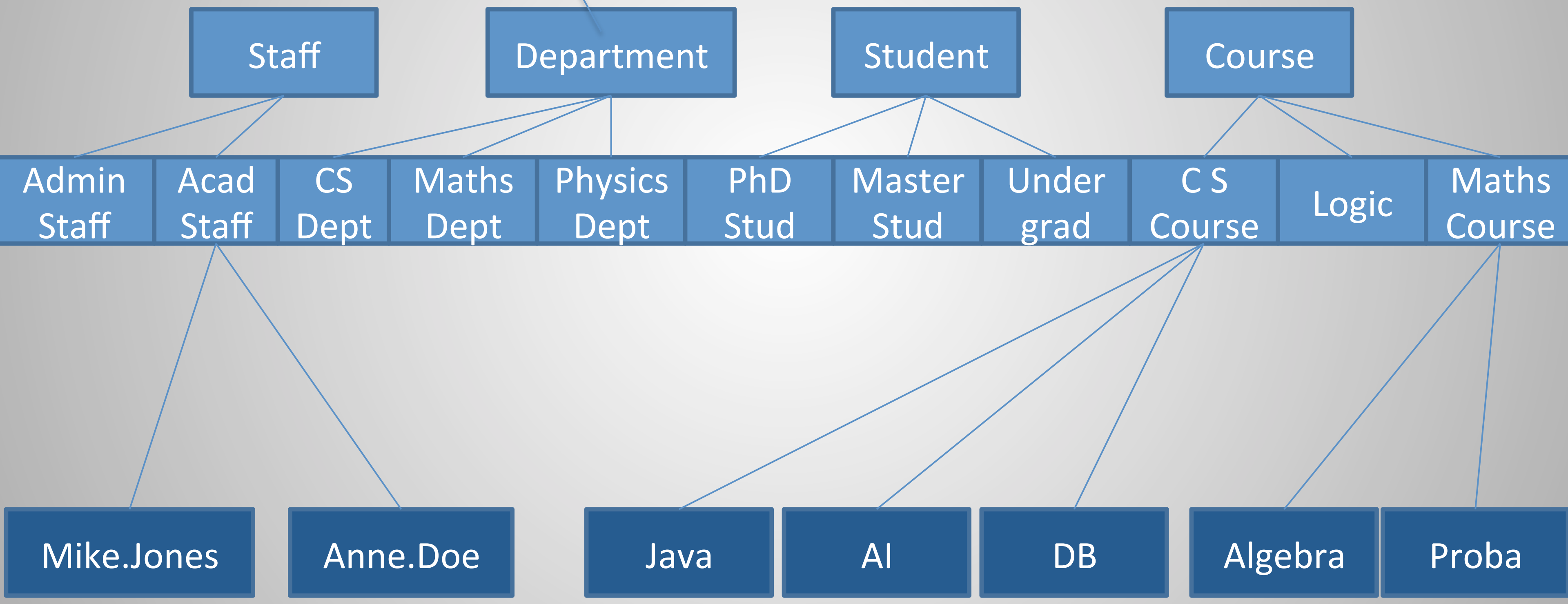
Reasoning is essential for

- Better answering
  - more precise answers
  - refining queries with too many answers
  - relaxing queries with no answer
- Better integrating data sources
  - Relating objects in different data sources enabling their integration
  - Detecting/resolving inconsistencies or redundancies

Classes and class hierarchy

A taxonomy: a hierarchy of classes

CS Dept **isa** Department



Objects in these classes

Instance of a class

A class is interpreted as a set of objects

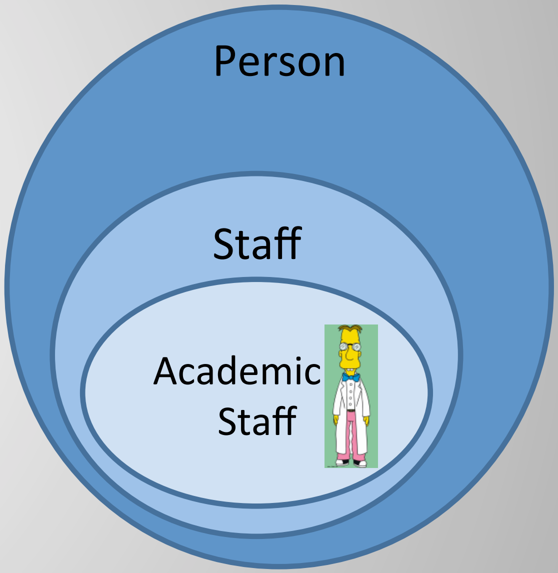
- Mike.Jones **instanceOf** AcademicStaff
- AcademicStaff (Mike.Jones)

The relation isa is interpreted as set inclusion

- AcademicStaff **isa** Staff
- $\forall x$ (AcademicStaff(x) $\Rightarrow$ Staff(x))

Inference

- Staff (Mike.Jones)



Relations

Declaration of relations with their signatures

- TeachesIn(AcademicStaff, Course)
- TeachesTo(AcademicStaff, Student),
- Leads(Staff, Department)

Instances of relations

Relations are interpreted as binary relations between objects

- TeachesIn(Mike.Jones, Java)
- $\forall$x,y (TeachesIn(x, y) $\Rightarrow$ AcademicStaff(x) $\wedge$ Course(y))

Ontology = schema + instance (aka Knowledge base)

Database **schema**

- The class hierarchy
  - The set of class names and the **isa** relation
- The signatures of relations
- Other constraints that are used for
  - checking data consistency (like dependencies in databases)
  - inferring new facts

Database **instance**

- The set of base facts that forms the database
- The set of facts that may be inferred

Ontology languages

RDF: to describe facts

RDFS: to define simple ontologies about RDF facts

OWL: a richer ontology language

We present them rapidly

We mention a family of ontology languages, **description logics**

- OWL may be seen as a syntax for a description logic

RDF triples

Subject Predicate Object

In English: Dupond leads the CS department

In Logic: Leads(Dupond,CS department)

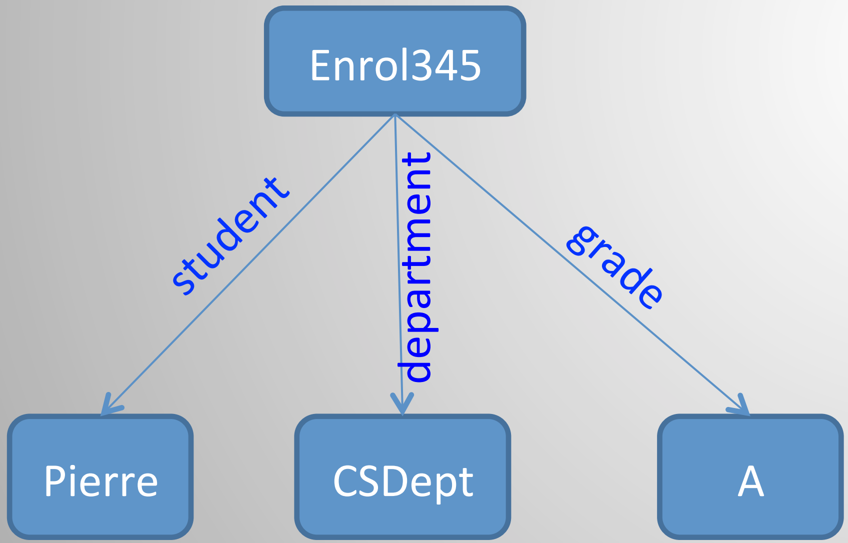
More triples < Dupond TeachesIn UE111 > < Dupond TeachesTo Pierre >

< Pierre EnrolledIn CSDept > < Pierre RegisteredTo UE111 >

With web resources:

Beyond binary relations Enrolment(Pierre, CSDept, A)

Student	Department	Grade
Pierre	CSDept	A

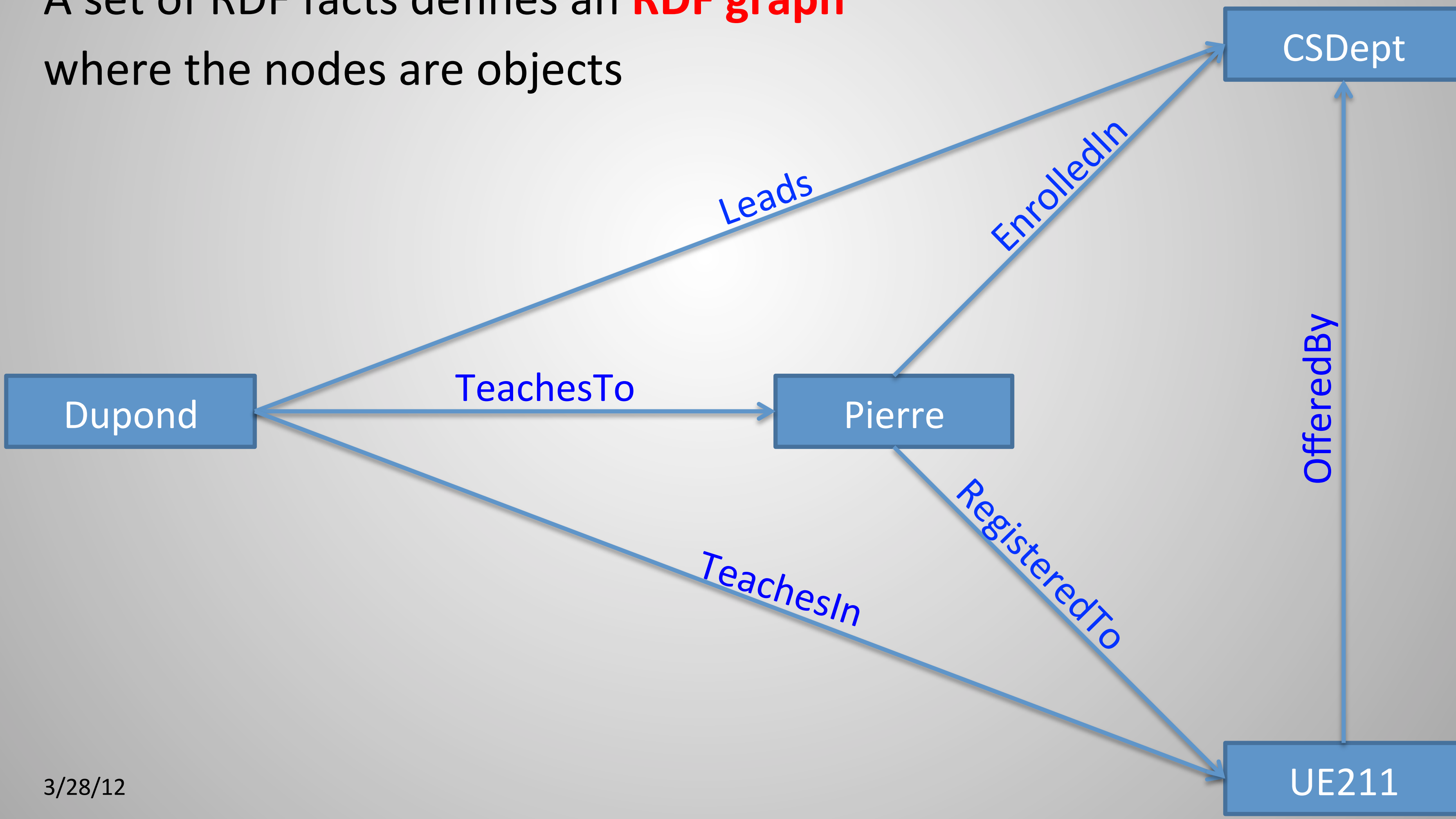


Enrol345	Student	Pierre
Enrol345	Department	CSDept
Enrol345	Grade	A

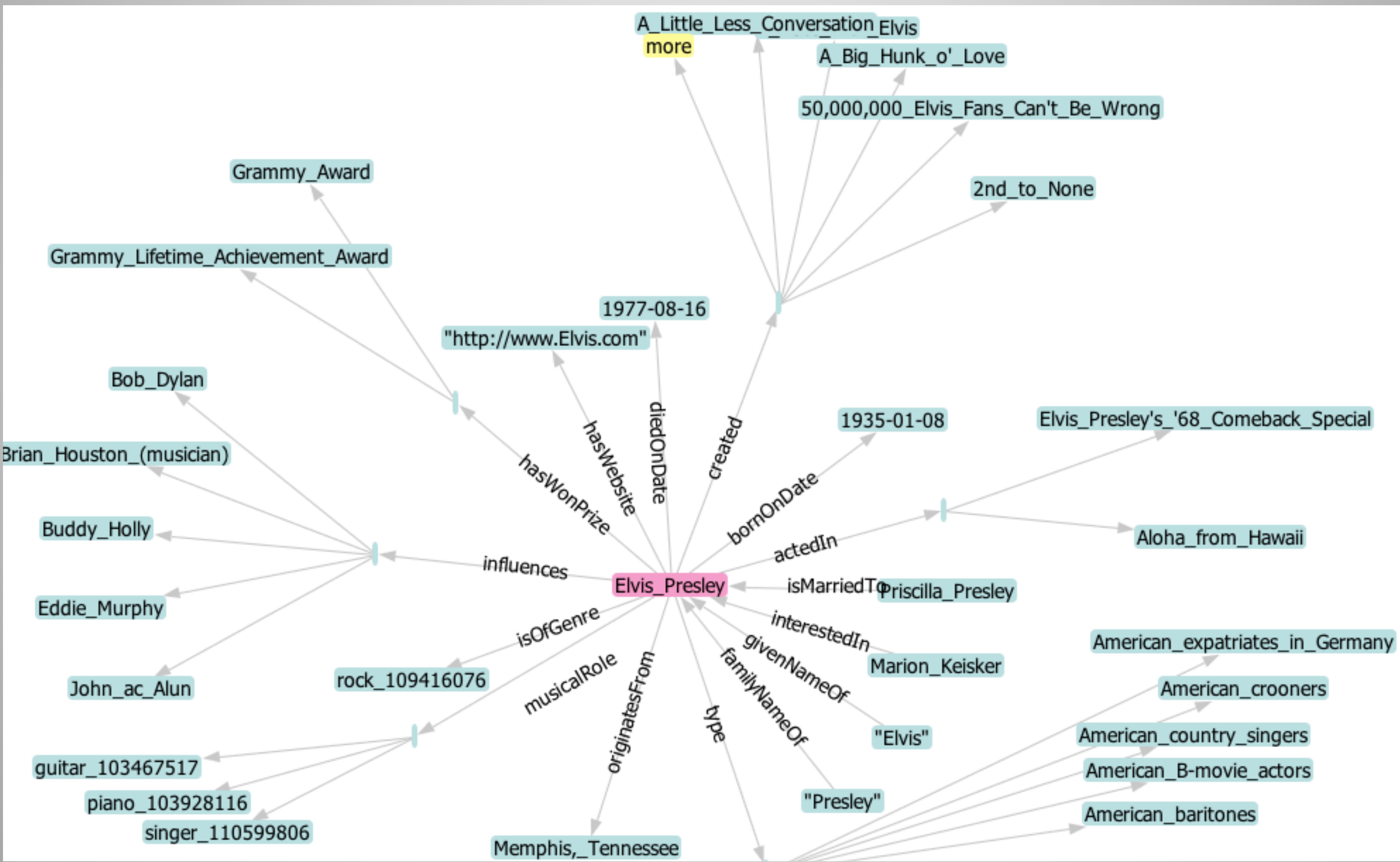
Rather inelegant? Yes

RDF graph

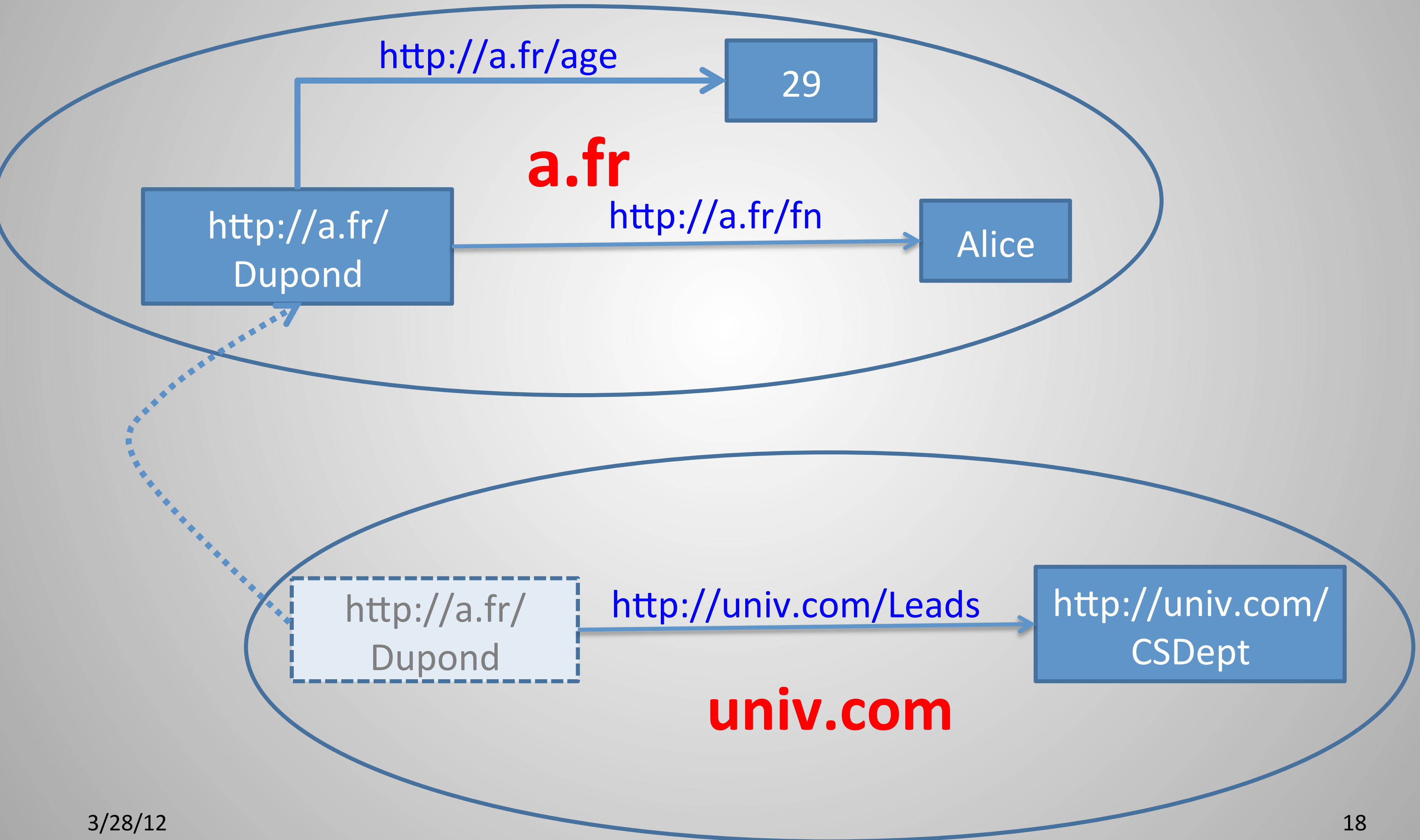
A set of RDF facts defines an **RDF graph** where the nodes are objects



Example of an RDF Graph: Elvis in Yago



The RDF graph is global



Some standard vocabularies

rdf: The basic RDF vocabulary

rdfs: RDF Schema vocabulary

dc: Dublin Core (predicates for describing documents)

s: Schema.org (predicates for describing web content)

- Vocabulary for people, movies, events, etc

cc: Creative Commons (types of licenses)

RDFS: RDF Schema

The schema in RDF is super simplistic

An **RDF Schema** defines the schema of a richer ontology

Do net get confused

- RDFS can use RDF as syntax
- I.e., RDFS statements can be expressed as RDF triples using RDFS keywords for **properties** and **objects**

Examples for RDF Schema – using RDF syntax

Declaration of classes and subclass relationships

– < Staff **rdf:type rdfs:Class** > < Java **rdfs:subClassOf** CSCourse >

Declaration of instances

– < Dupond **rdf:type** AcademicStaff >

Declaration of relations

– < RegisteredTo **rdf:type** rdf:Property >

Declaration of subproperty relationships

– < LateRegisteredTo **rdfs:subPropertyOf** RegisteredTo >

Declaration of domain/range restrictions for predicates

– < TeachesIn **rdfs:domain** AcademicStaff >

– < TeachesIn **rdfs:range** Course >

i.e. TeachesIn(AcademicStaff , Course)

Owl

OWL extends RDFS with the possibility to express additional constraints

- Disjointness between classes
- Constraints of functionality and symmetry on predicates
- Intentional class definitions
- Class union and intersection

Examples

- Departments can be lead only by professors
- Only professors or lecturers may teach to undergraduate students.

Description Logics

Philosophy: isolate **decidable** fragments of first-order logic allowing reasoning about classes and binary relations

These fragments are called Description Logics

The DL jargon:

- the classes are called **concepts**
- the properties are called **roles**
- the schema is called the **Tbox**
- the instance is called the **Abox**
- the ontology = Tbox + Abox

Semantics of main concepts

$I(C1 \sqcap C2) = I(C1) \cap I(C2)$

$I(\forall R.C) = \{o1 \mid \forall\ o2\ [(o1,o2) \in I(R) \Rightarrow o2 \in I(C)]\}$

$I(\exists R.C) = \{o1 \mid \exists\ o2.[(o1,o2) \in I(R) \land o2 \in I(C)]\}$

$I(\neg C) = dom(I) - I(C)$

$I(R^{-}) = \{(o2,o1) \mid (o1,o2) \in I(R)\}$

The kind of questions that are considered

Satisfiability checking: Given an ontology K = <T ,A>, is K satisfiable?

– I.e., is the ontology consistent? does there exist a possible world?

Subsumption checking: Given a Tbox T and two concept expressions *C* and *D, does* $T \models C \sqsubseteq D$?

– I.e.. is C a subclass of C' in any possible world

Instance checking: Given an ontology K = <T ,A>, an individual *e* and a concept expression *C, does* $K \models C(e)$?

Query answering: Given an ontology K = <T ,A>, and a concept expression *C,* finds the set of individuals *e* such that $K \models C(e)$?

These problems are undecidable for full OWL

Querying ontologies

Querying using RDFS

RDFS statements can be used to infer new triples

Example

- Base fact *ResponsibleOf (durand,ue111)*
- Rule: *ResponsibleOf (X,Y)⇒Professor (X)*
- rule *Professor (X)⇒AcademicStaff (X)*

If we ask the query "who is in the Academic Staff?", we want Durand in the answer

For this, we can use inference by saturation

- Keep inferring new facts until a fixpoint is reached
- Only polynomially many facts can be added
- In ptime

More complex languages: description logics

Develop as a good compromise between expressive power and reasonable complexity of query answering

– Example: dl-light also in ptime but much richer

Avoid saturation by using query reformulation

Answering queries by reformulation

Professor(Jim), HasTutor(John,Mary), TeachesTo(John,Bill)

HasTutor (y,z)←Student (y)

Student $\sqsubseteq$ *∃HasTutor* in DL jargon

Query q0: *q0(x)←TeachesTo(x,y) ∧ HasTutor (y,z)*

Query *q1* computes answers to *q0:*

q1(x)←TeachesTo(x,y) ∧ Student (y)

One can use standard query processors to answer the query

Difficulties

For some logics, reformulation is not possible

For some logics, inconsistencies

- It may be the case that there is no model satisfying all the statements

SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) is a query language for RDF

```
SELECT ?dep
WHERE {
<http://a.fr/Dupond>   <http://univ.com/leads>   ?dep }
```

Find me all the values for ?dep such that the triple is true

Pattern matching over the RDF graph

Many gadgets

Some ontologies provide “SPARQL endpoints”, i.e. a service than can receive SPARQL queries sent by a machine or typed by a human

Integration of data sources

Goal

Obtain data from different data sources with a single query/interface

- The data source have been developed independently, are autonomous and heterogeneous

Example:

- Sciences: query different genetic databases
- Business: query different catalogs from different vendors
- Accounting: integrate financial data from different branches

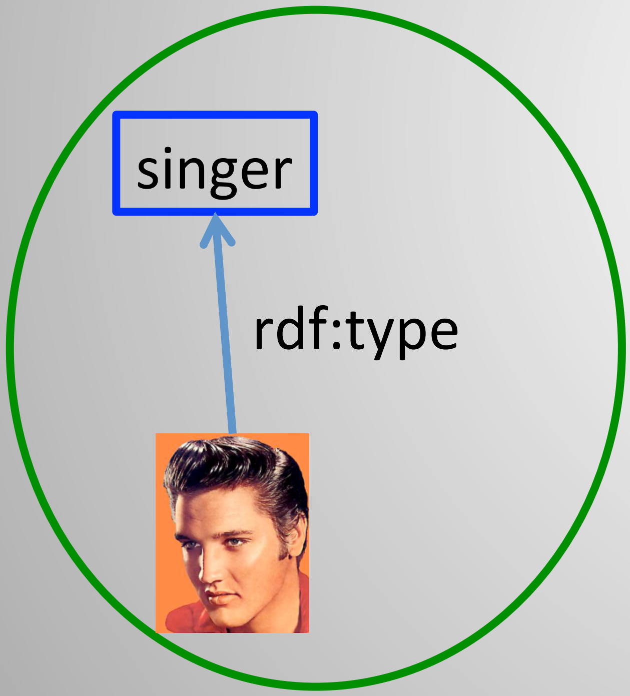
Use **semantics** to describe connections between data sources

1. Specify links
2. Specify views

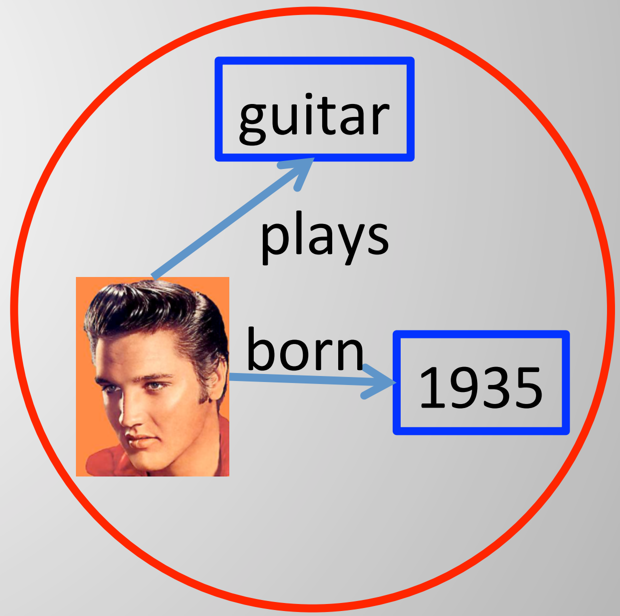
Specify links

Many ontologies talk about the same entity with different URIs

This is bad, because we cannot join the information



Elvisopedia
(http://elvisopedia.org/)

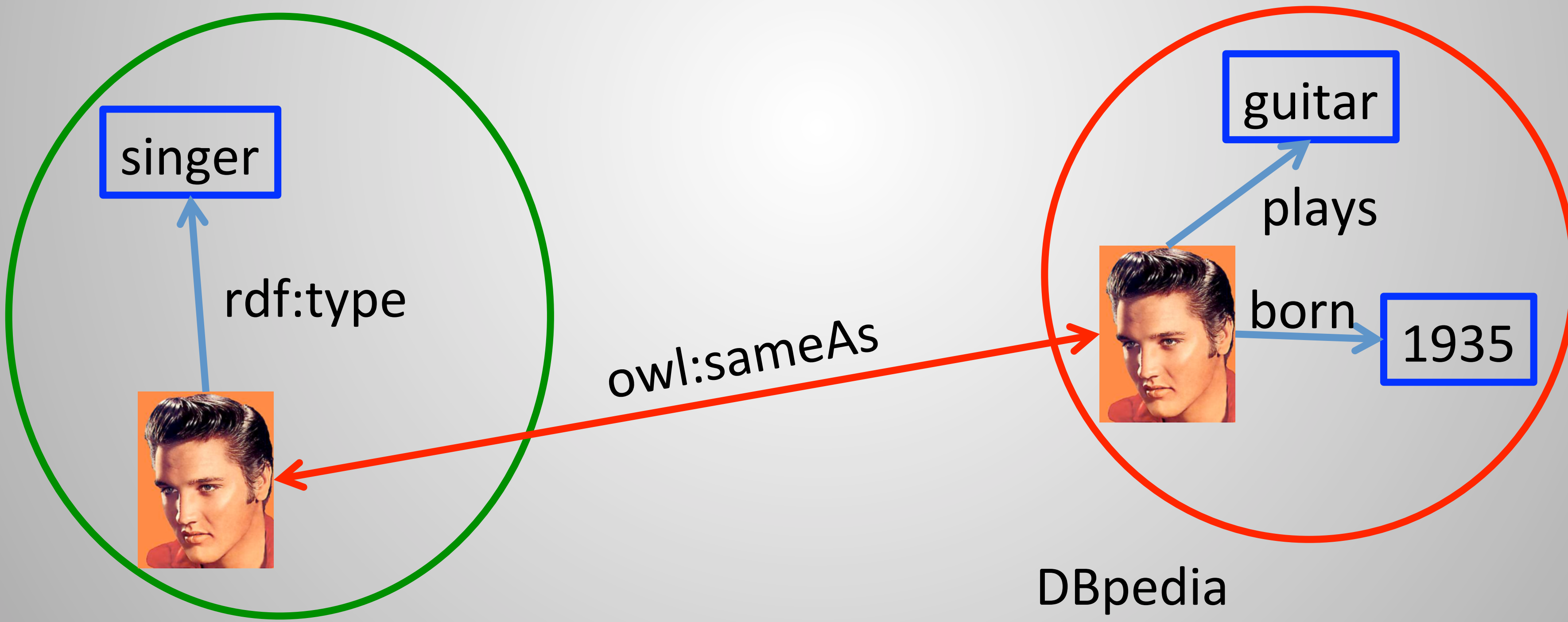


DBpedia
(http://dbpedia.org/)

Specify links

OWL provides vocabulary to link equivalent entities

http://elvisopedia.org/Elvis owl:sameAs http://dpbedia.org/Elvis



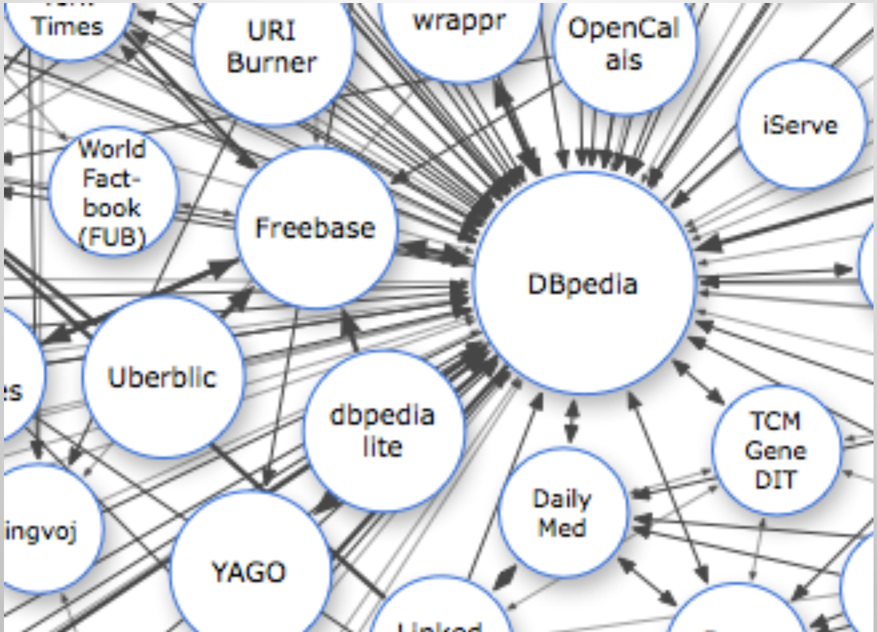
DBpedia
(http://dbpedia.org/)

Elvisopedia
(http://elvisopedia.org/)

Specify links: The Linking Data Project

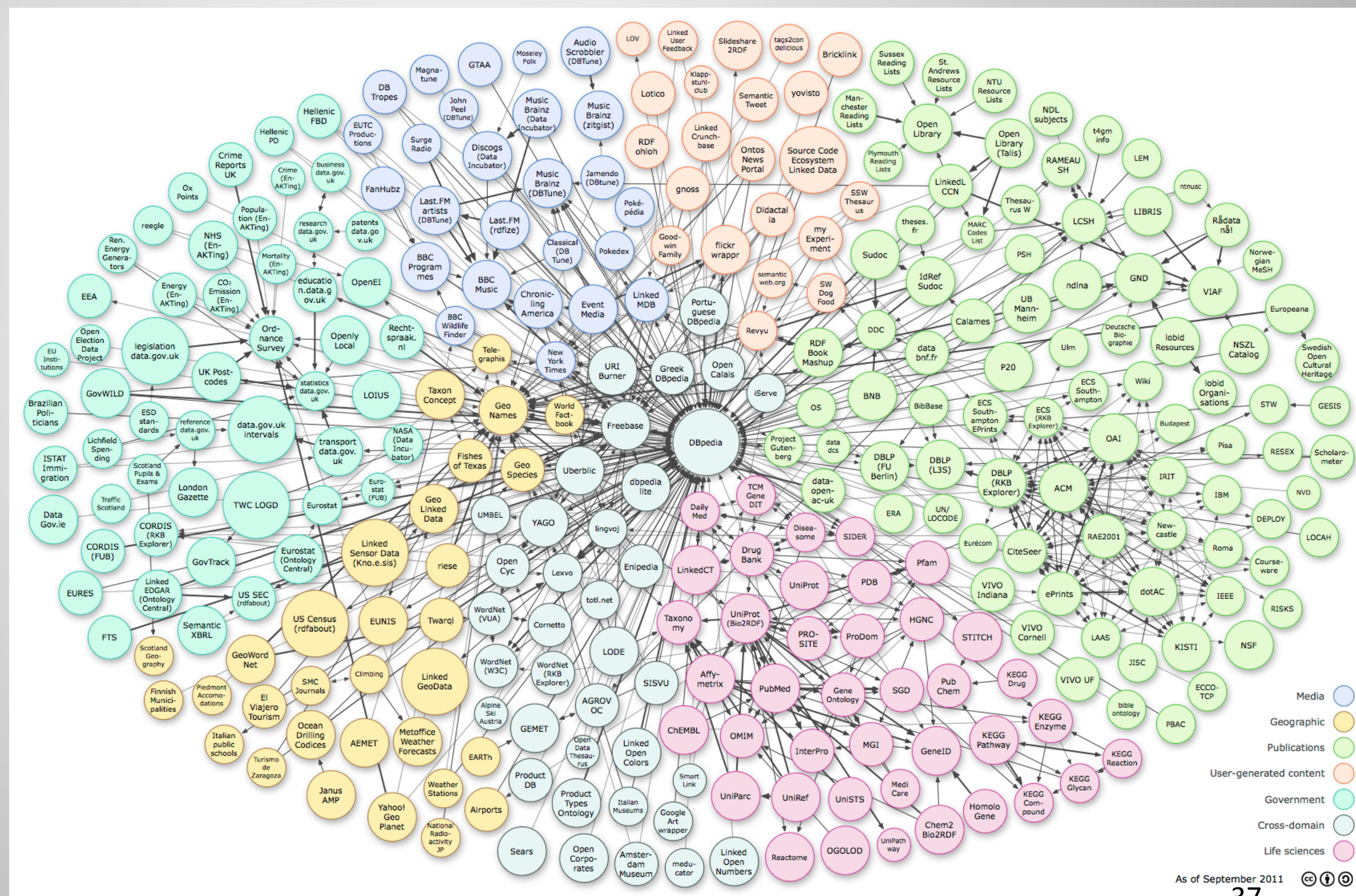
The **Linking Open Data Project** aims to interlink all open RDF data sources into one gigantic RDF graph

LINKINGOPENDATA
W3C SWEO Community Project



Specify links: The Linked Data Cloud

As of 2011: 295 ontologies, 25 billion triples, 400m links



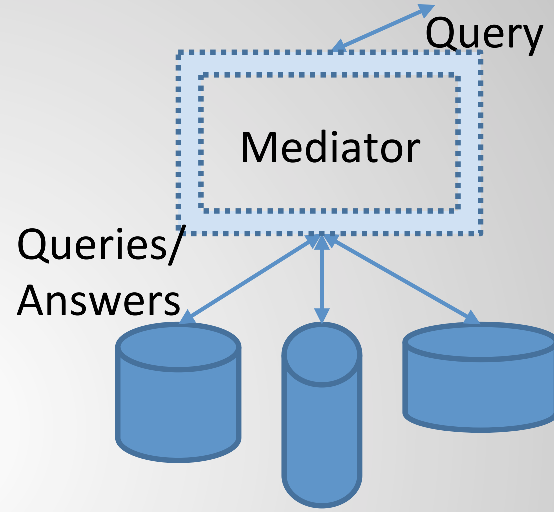
http://richard.cyganiak.de/2007/10/lod/imagemap.html

Recall

Specify views

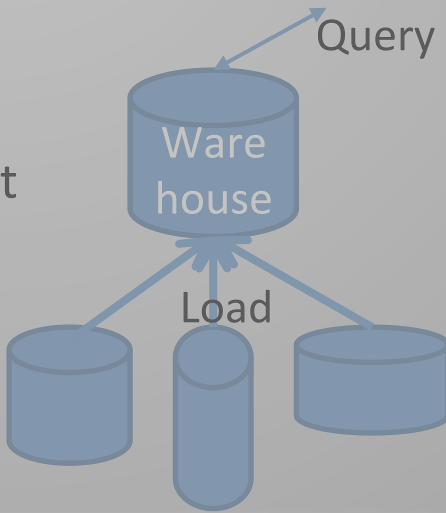
Mediating approach

- Global instance is virtual
- Query: cost of reformulation
- Creation and updates: no cost



Warehousing approach

- Global instance is materialized
- Query evaluation is very efficient
- Updates are costly



Recall

Specify views

Local-As-Views (LAV) approach: the local relations are defined as views over the global relations

Query processing

- Rewriting the users queries (expressed using global relations) in terms of local relations ⇒ logical query plans
- Combine the answers of logical query plans to obtain the result

Global-As-Views (GAV) approach: the global relations are defined as views over the local relations

Algorithms

Several algorithms have been proposed

- **Bucket**
- Minicon: an optimization of Bucket
- Inverse-rules: in the spirit of algorithm for GAV
  (less efficient but simple to explain)

A jewel of data integration

The bucket algorithm

By example

Setting

Input

- A set of local relations defined as conjunctive views over the global schema
- A conjunctive query over the global schema

Output

- A set of conjunctive queries over the local relations that answers the query

Example

Global schema

- Student(studentName), University(uniName), Program(title), MasterProgram(title), Course(code), EnrolledIn(studentName,title), EnrolledInCourse(studentName, code), PartOf(code,title), RegisteredTo(studentName, uniName), OfferedBy(title, uniName)

Rules

- **S1.Catalogue**(U,P) :- FrenchUniversity(U), Program(P), OfferedBy(P,U), OffereBy(P',U), MasterProgram(P'),
- **S2.Erasmus**(S,C,U) :- Student(S), EnrolledInCourse(S,C), PartOf(C,P), OfferedBy(P,U), EuropeanUniversity(U), RegisteredTo(S,U'), EuropeanUniversity(U'), U≠U'
- **S3.CampusFrance**(S,P,U) :- NonEuropeanStudent(S), EnrolledInProgram(S,P), Program(P), Offeredby(P,U), FrenchUniversity(U), RegisteredTo(S,U)
- **S4.Mundus**(P,C) :- MasterProgram(P), OfferedBy(P,U), OfferedBy(P,U'), EuropeanUniversity(U), NonEuropeanUniversity(U), PartOf(C,P)

Example

q(x) :- **RegisteredTo(s,x)**, EnrolledIn(s,p), MasterProgram(p)

We use the view definition

S3. CampusFrance(S,P,U) :-

NonEuropeanStudent(S), EnrolledIn (S,P),

Program(P), Offeredby(P,U), FrenchUniversity(U),

RegisteredTo(S,U)

We record that

Bucket(RegisteredTo(s,x)) contains S3.CampusFrance(s, v1,x)

Combining the buckets

q(x) :- RegisteredTo(s,x), EnrolledIn(s,p), MasterProgram(p)

Combining the buckets

- Bucket(RegisteredTo(s,x)) = { S3.CampusFrance(s, v1,x) }
- Bucket(EnrolledInProgram(s,p)) = {S3.CampusFrance(s, p,v2) }
- Bucket(MasterProgram(p)) = { S1.Catalogue(v3,v4), S4.Mundus(p,v5) }

2 candidate rewritings:

- r1(x) :- S3.CampusFrance(s, v1,x), S3.CampusFrance(s, p,v2), S1.Catalogue(v3,v4)
- r2(x) :- S3.CampusFrance(s, v1,x), S3.CampusFrance(s, p,v2), S4.Mundus(p,v5)

Homomorphism Theorem

Testing the candidates

q(x) :- RegisteredTo(s,x), EnrolledIn (s,p), MasterProgram(p)

r1(x) :- S3.CampusFrance(s, v1,x), S3.CampusFrance(s, p,v2), S1.Catalogue(v3,v4)

Expand(r1(x)) :- NonEuropeanStudent(s), EnrolledIn(s,v1), Program(v1), Offeredby(v1,x), FrenchUniversity(x), RegisteredTo(s,x), EnrolledIns,p), Program(p), Offeredby(p,v2), FrenchUniversity(v2), RegisteredTo(s,v2), FrenchUniversity(v3), Program(v4), OfferedBy(v4,v3), OffereBy(v5,v3), MasterProgram(v5)

Expand(r1(x)) $\nsubseteq$ q(x)

r1 is not a valid rewriting

test that r2 is one

Conclusion

More and more structured information on the web

More and more semantic on the web

E.g., the UK government makes much of its data available online in RDF – by law

Enriching the standard web

- Publishing semantic descriptions of web services/pages
- Microdata an upcoming W3C standard to annotate HTML pages with RDF data

web applications and search engines start using such semantic annotations

- The DBpedia Mobile App retrieves data from the Linked Open Data Cloud to show places of interest around you

More and more structured data on the web

More and more structured data published notably public

- Lots of tables in html or pdf
- Lots of data in deep web behind forms
- Typically better quality than unstructured data

Many ontologies: e.g., DBPedia or Yago

Need: tools for searching, visualization, linking, integration

Building ontologies

Extract one from existing text sources

- Yago built from Wikipedia
- Difficult: Natural language processing is complex

Have humans collaborate to build it

- Freebase: Freebase is an open, Creative Commons licensed graph database with millions of entities
- Linked data: publish RDF links between web data

Integrate different ontologies by aligning their concepts and relations

- Paris [SuchanekAbiteboulSenellart]

More reasoning

The scalability of reasoning on web data requires light-weight ontologies

- Reasoning should be feasible – polynomial
- Preferable if query answering can be performed with a relational database engines

RDFS is OK but too limited

Full OWL is too complex

References

Web data management,

Abiteboul, Manolescu, Rigaux, Rousset, Senellart

webdam.inria.fr/Jorge

Semantic web, Fabian Suchanek,

suchanek.name/work/teaching/inf347/inf347_sw.ppt

inria informatiques mathématiques

COLLÈGE DE FRANCE 1530

ENS CACHAN

Ouverture des données publiques

François Bancilhon

Informaticien & PDG

Carrière académique

- INRIA, MCC, Paris Sud
- Encore un des plus cités en BD

Carrière industrielle

- Fondateur ou dirigeant de O2 Technology , Arioso, Xylème, Ucopia, Mandriva, Data Publica

Dirige l'Initiative Services Mobiles pour l'INRIA

CEO de Data Publica

Archivage du Web

Julien Masanès

Conservateur de bibliothèque & Directeur

A été en charge de l'archivage du web à la BNF

Fondateur de l'« International Internet Preservation Consortium »

Fondateur de l'« International Web Archiving Workshop »

Directeur de la Fondation « Internet Memory »