R_2O+ODEMapster: Upgrading Relational Legacy Data to the Semantic Web

Jesús Barrasa Rodríguez
jbarrasa@eui.upm.es

Ontology Engineering Group

What is R2O + ODEMapster?

- Framework to **upgrade relational legacy data to the Semantic Web**.
- Based on the **declarative description of mappings** between relational and ontology elements
- Exploitation of mappings by a domain independent processor.

The problem:

- No explicit semantics in databases
- Need to facilitate interchange, combination & automatic reasoning on their content.
- *“One of the main driving forces for the Semantic Web will be the expression on the Web, of the vast amount of relational database information in a way that can be processed by machines“*

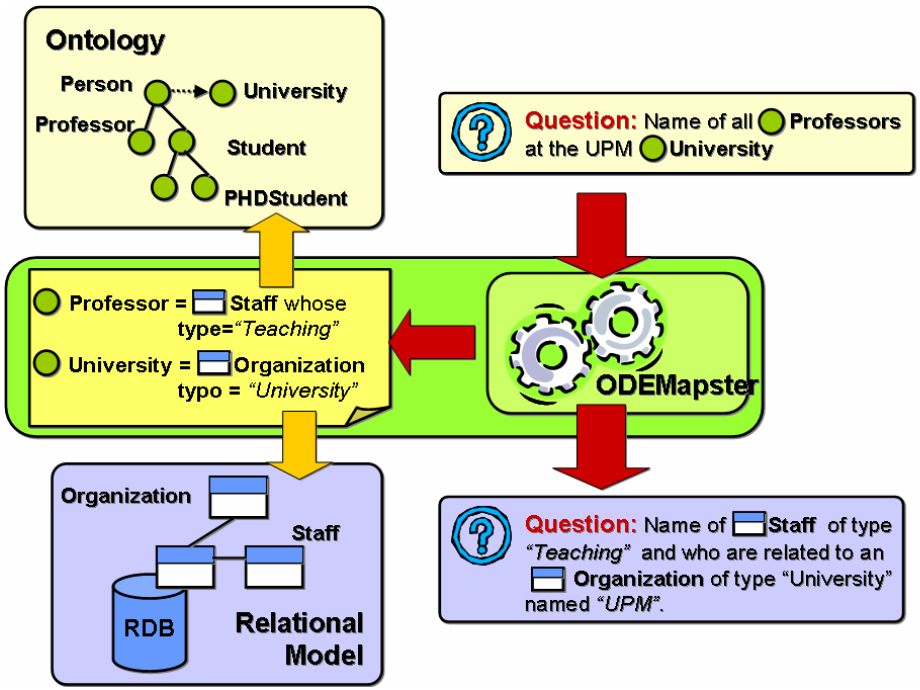
Tim Berners Lee (sept 98)

The approaches:

	Approach	Pros / Cons
Ontology	Created ad-hoc	Simpler mapping situations. Less expressivity required. Less flexible. Only intended use.
	Reuse existing one	Promotes reuse. More expressivity required to overcome differences Different semantic views for the same database.
Architect.	Wrapper	Non reusable Complex evolution and maintenance Better performance
	Generic engine + declarative definition	Simple evolution and maintenance Explicit mappings Allows composition and verifications Lower performance
Exploitation	Massive upgrade (batch)	Creates independent repository, no interference with local processing at source High query performance Inadequate if rapidly changing data source.
	Query driven (on demand)	Fresh information. Adequate for changing data sources. Delay in query processing Competes with local processing at source

- **Info Integration:** OBSERVER, PICSEL, MOMIS. Mediator approach. Wrapper dependent.
- **Upgrade:** D2R, KAON-Reverse. Only massive batch upgrade. Lack expressiveness

Our proposal:



- Mapping of **independently conceived, developed and maintained ontologies & databases**
- **Declarative definition of correspondences** with R_2O and exploitation by domain independent engine
- **Extensible set of primitives**, not limited by DBMS expressivity

The mapping language: R_2O

- Formal declarative mapping description language.
  - XML based
  - Extendable set of condition & transformation primitives

R_2O: Conditions

```
condition "match-regexp“
  arg-restriction
    on-param "string“
    has-column jobs.salaryRange
  arg-restriction
    on-param "regexp“
    has-value ([:digit:]*)-([:digit:]*)
```

Use of non-DBMS operations. Extendability

R_2O: Transformations

```
operation "concat"
  arg-restriction
    on-param "string1“
    has-value "http://net.test.r2o/job-“
  arg-restriction
    on-param "string2“
    has-transform
      operation "concat“
        arg-restriction
          on-param "string1“
          has-column jobs.id
        arg-restriction
          on-param "string2“
          has-column jobtypes.code
```

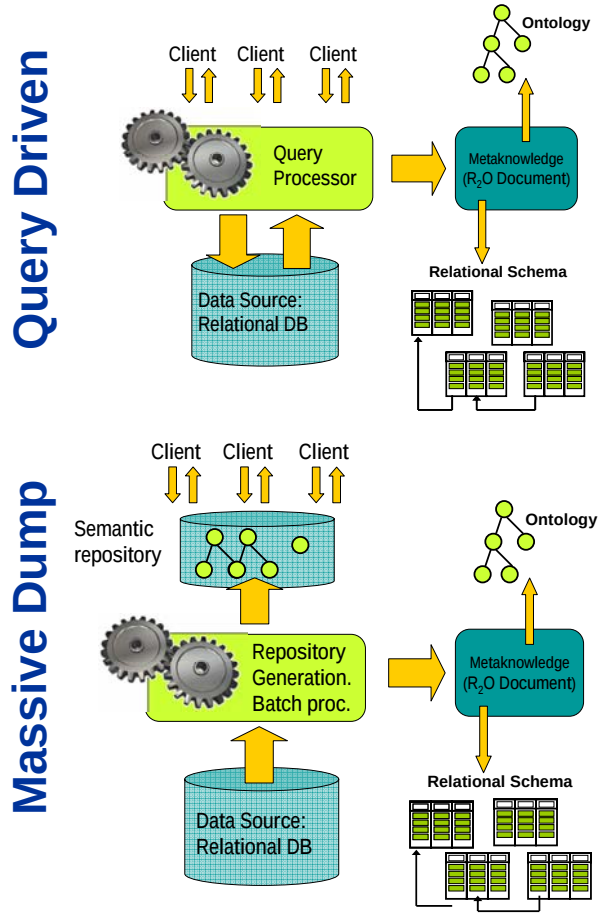
Embeded operations. Complex tranformation definitions.

R_2O: Attribute mappings

```
attributemap-def“http://net.onto/jobs#type"
  selector
    applies-if
      condition […condition desc 1…]
    aftertransform
      operation […transformation desc 1…]
  selector
    applies-if
    aftertransform …
```

If-then rule structure

The Query engine: ODEMapster



- Two operation modes:
  - **Query Driven:** Query translator behavior. Selective & on line upgrade
  - **Massive Dump:** Batch process. Creates a semantic RDF repository. Upgrades the full DB.

ODEMapster execution

- Query and R_2O mapping document are parsed
- Delegable part of query is translated into Data source's SQL
- Retrieved results are post-processed (non delegable inferences are carried out) and ontology instances are generated.

Further information

Ontology Engineering Group

http://www.oeg-upm.net

jbarrasa@eui.upm.es