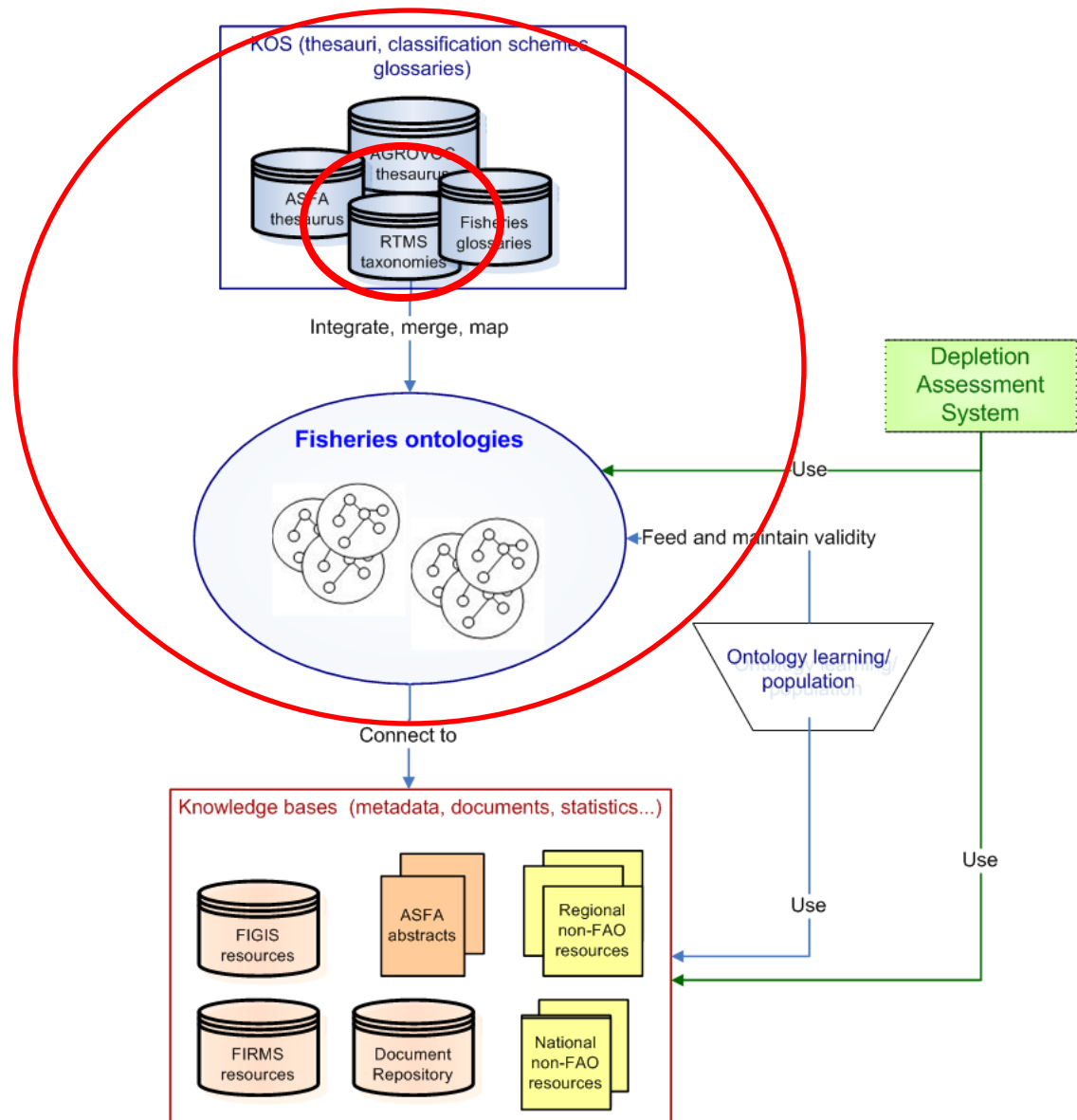


Boris Villazón-Terrazas, Freddy Priyatna  
bvillazon@fi.upm.es, fpriyatna@delicias.dia.fi.upm.es  
Ontology Engineering Group. Laboratorio de Inteligencia Artificial  
Departamento de Inteligencia Artificial  
Facultad de Informática  
Universidad Politécnica de Madrid

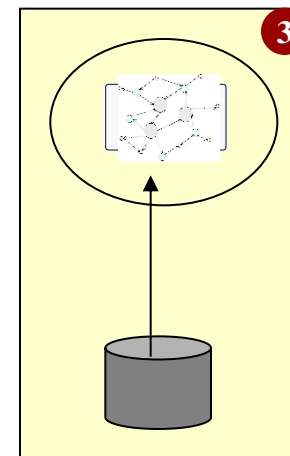
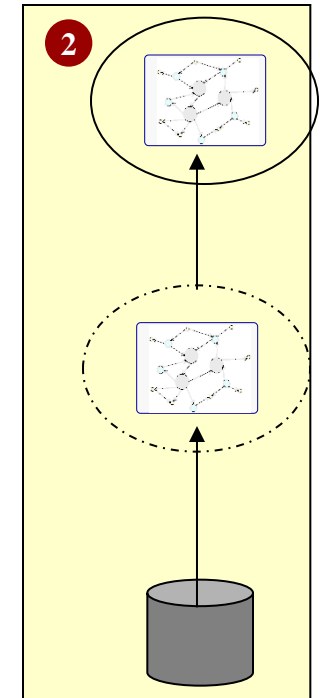
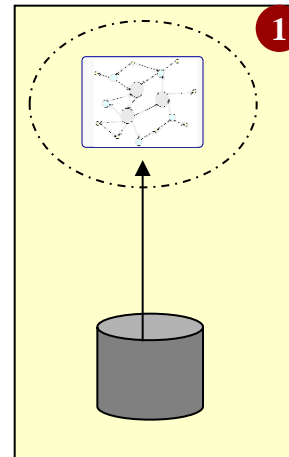
- Introduction
- $R_2O$ 
  - ConceptMap definition
  - AttributeMap definition
  - RelationMap definition
- ODEMapster
  - Query Driven
  - Massive batch data upgrade
- FAO Use Case
- Multiple Database to RDF





- Ontologies that will be used by the FSDAS (Ontology Driven Fisheries Stock Depletion Alert System )

# Existing approaches

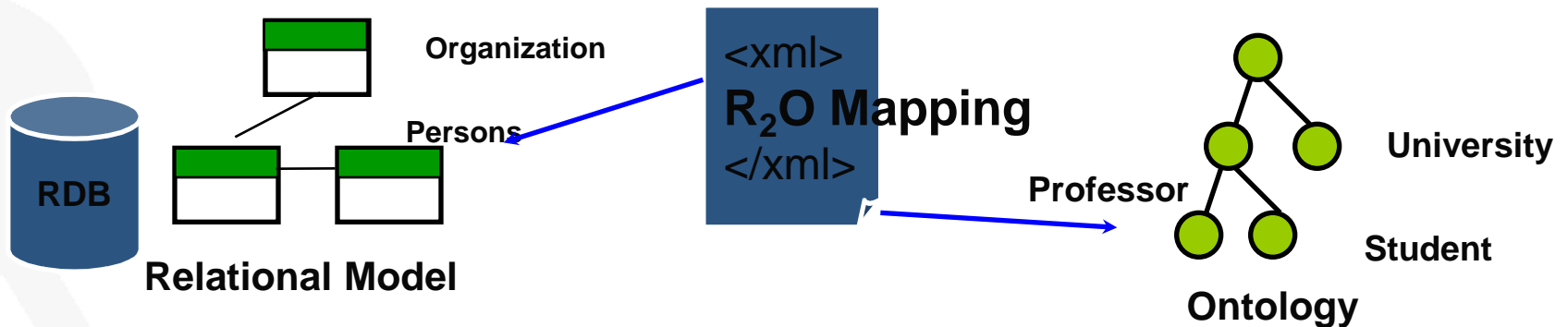
1. To build a **new ontology** from a database schema and content (OntoStudio, KaOn Reverse)
2. To map the ontology created in approach (1) to a **legacy ontology** (NeOn toolkit UKARL)
3. To map an existing DB to a **legacy ontology** (NeOn Toolkit UPM)

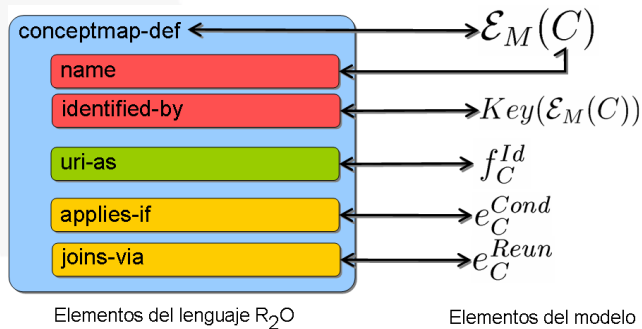


 new ontology  
 existing ontology

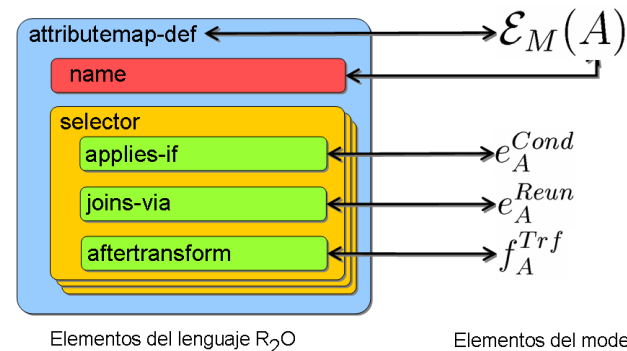
- Introduction
- $R_2O$ 
  - ConceptMap definition
  - AttributeMap definition
  - RelationMap definition
- ODEMapster
  - Query Driven
  - Massive batch data upgrade
- FAO Use Case
- Multiple Database to RDF

- R<sub>2</sub>O is an extensible, fully declarative language to describe mappings between relational database schemas and ontologies.

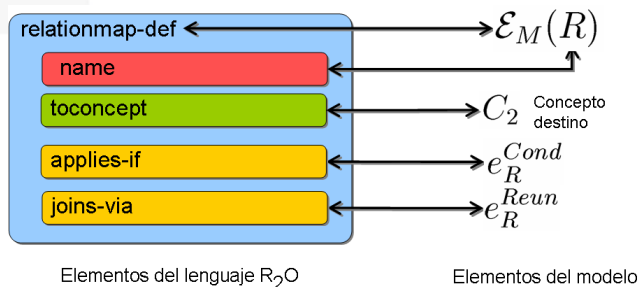




```
<conceptmap-def name="Customer">
  <identified-by> Table key </identified-by>
  <uri-as> operation </uri-as>
  <applies-if> condition </applies-if>
  <joins-via> expression </joins-via>
  <documentation>description ...</documentation>
  <described-by>attributes,relations</described-by>
</conceptmap-def>
```



```
<attributemap-def name="http://esperonto/ff#Title">
  <aftertransform>
    <operation oper-id="constant">
      <arg-restriction on-param="const-val">
        <has-column>fsb_ajut.titol</has-column>
      </arg-restriction>
    </operation>
  </aftertransform>
</attributemap-def>
```

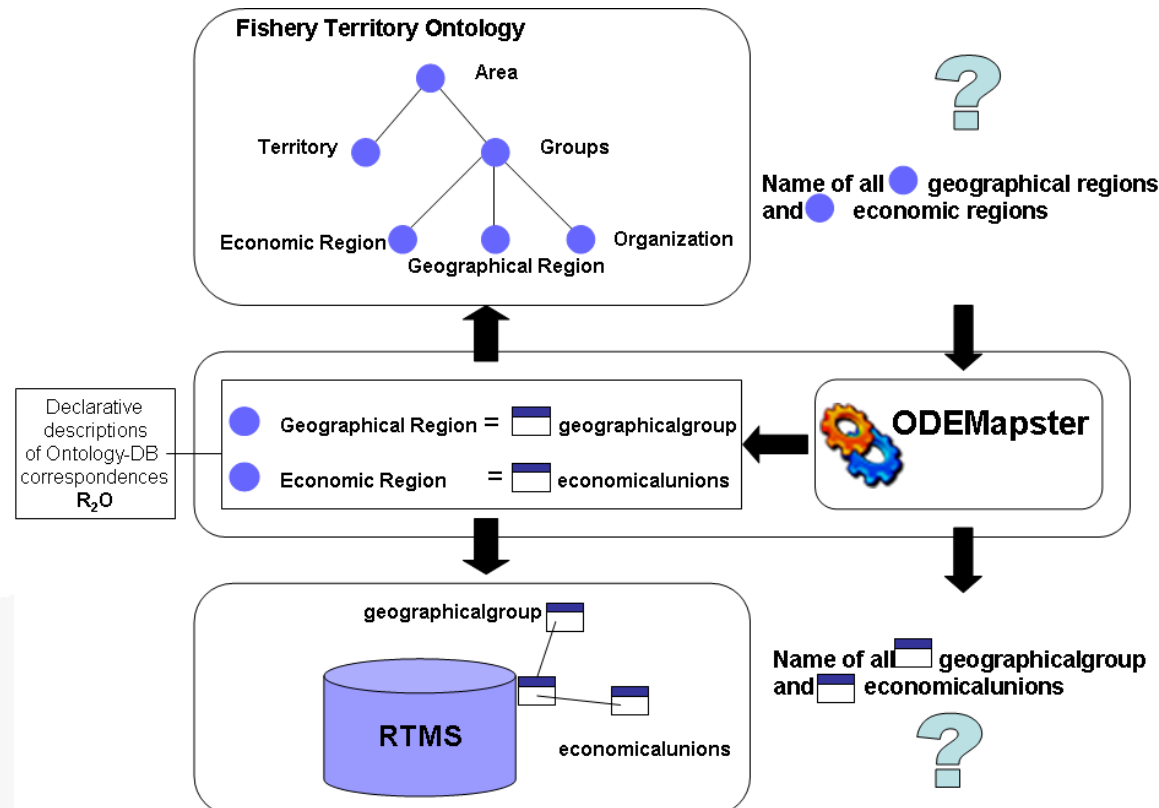


```
<relationmap-def name="http://esperonto/ff#isCandidateFor">
  <to-concept name="http://esperonto/ff#FundOpp">
  <joins-via>
    <operation oper-id="equals">
      <arg-restriction on-param="value1">
        <has-column>fsb_ajut.id</has-column>
      </arg-restriction>
      <arg-restriction on-param="value2">
        <has-column>fsb_candidate.forFund</has-column>
      </arg-restriction>
    </operation>
  </joins-via>
</relationmap-def>
```

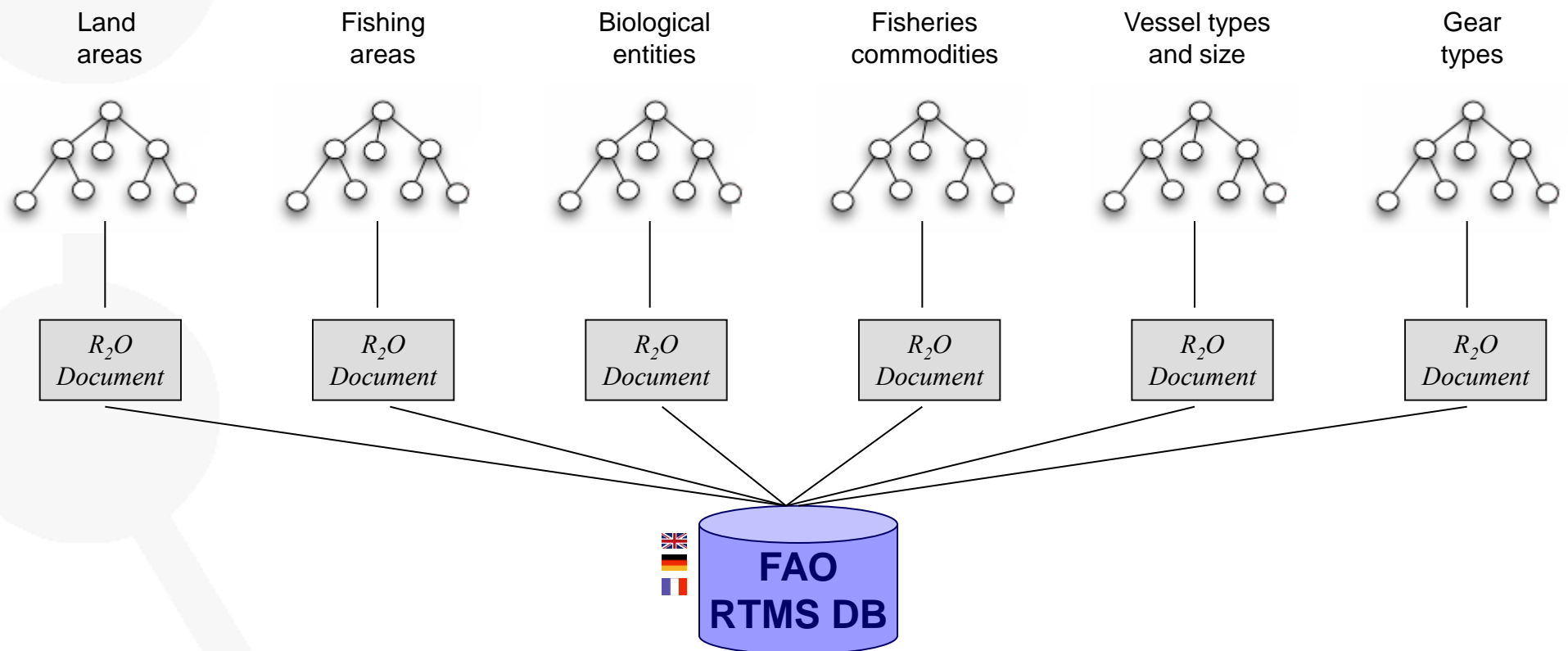
- Introduction
- $R_2O$ 
  - ConceptMap definition
  - AttributeMap definition
  - RelationMap definition
- ODEMapster
  - Query Driven
  - Massive batch data upgrade
- FAO Use Case
- Multiple Database to RDF



- The ODEMapster processor generates Semantic Web instances from relational instances based on the mapping description expressed in the  $R_2O$  document
  - Batch process: DB records migrated to the ontology
  - On demand: Querying the DB in terms of ontological terms



- Introduction
- $R_2O$ 
  - ConceptMap definition
  - AttributeMap definition
  - RelationMap definition
- ODEMapster
  - Query Driven
  - Massive batch data upgrade
- **FAO Use Case**
- Multiple Database to RDF



Land areas	
Concepts	4
Properties	25
Instances	289

Fishing areas	
Concepts	5
Properties	14
Instances	134

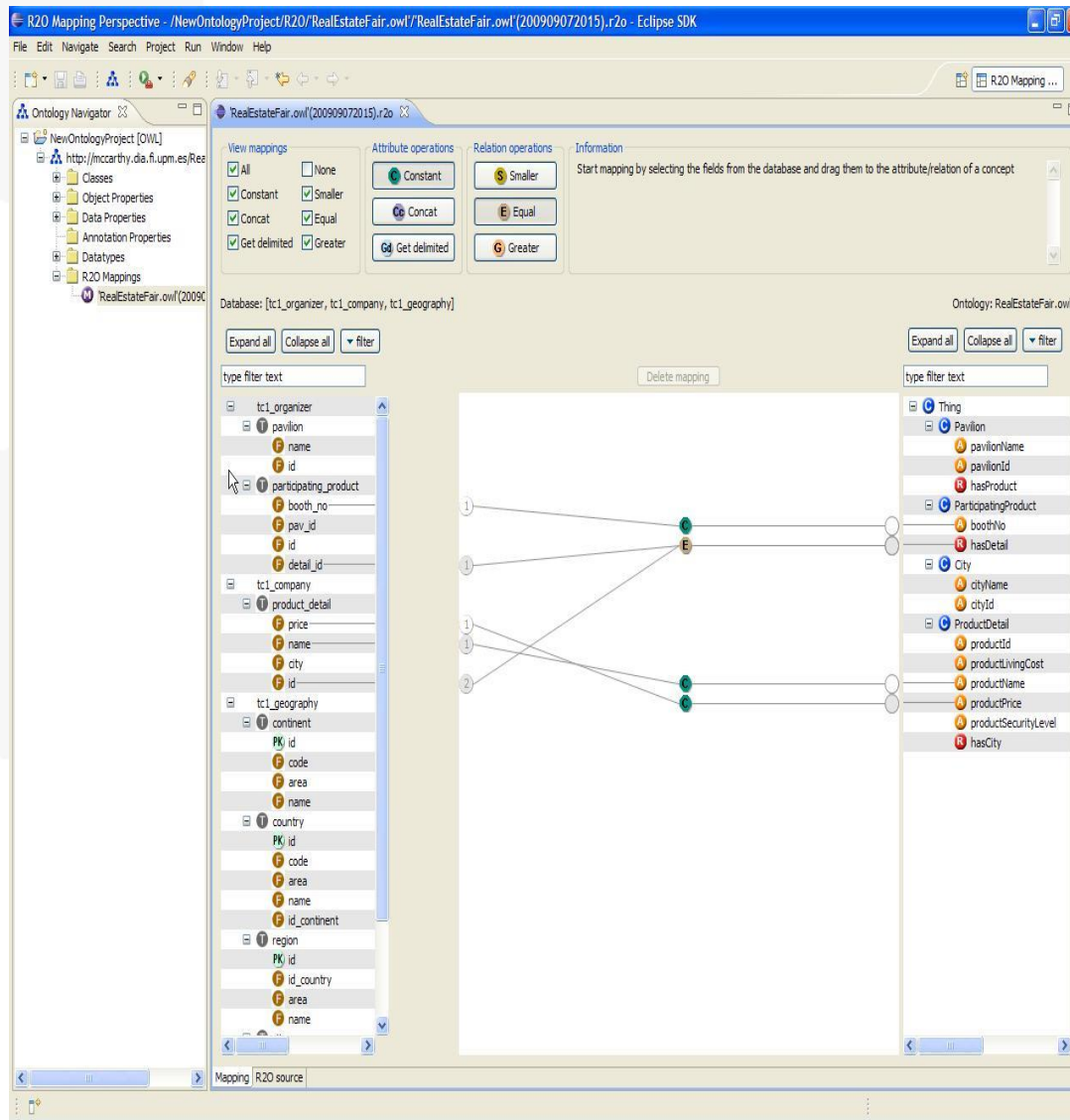
Biological entities	
Concepts	5
Properties	21
Instances	11571

Fisheries commodities	
Concepts	5
Properties	14
Instances	1380

Vessel types and size	
Concepts	5
Properties	20
Instances	120

Gear types	
Concepts	4
Properties	14
Instances	0

- Introduction
- $R_2O$ 
  - ConceptMap definition
  - AttributeMap definition
  - RelationMap definition
- ODEMapster
  - Query Driven
  - Massive batch data upgrade
- FAO Use Case
- Multiple Database to RDF

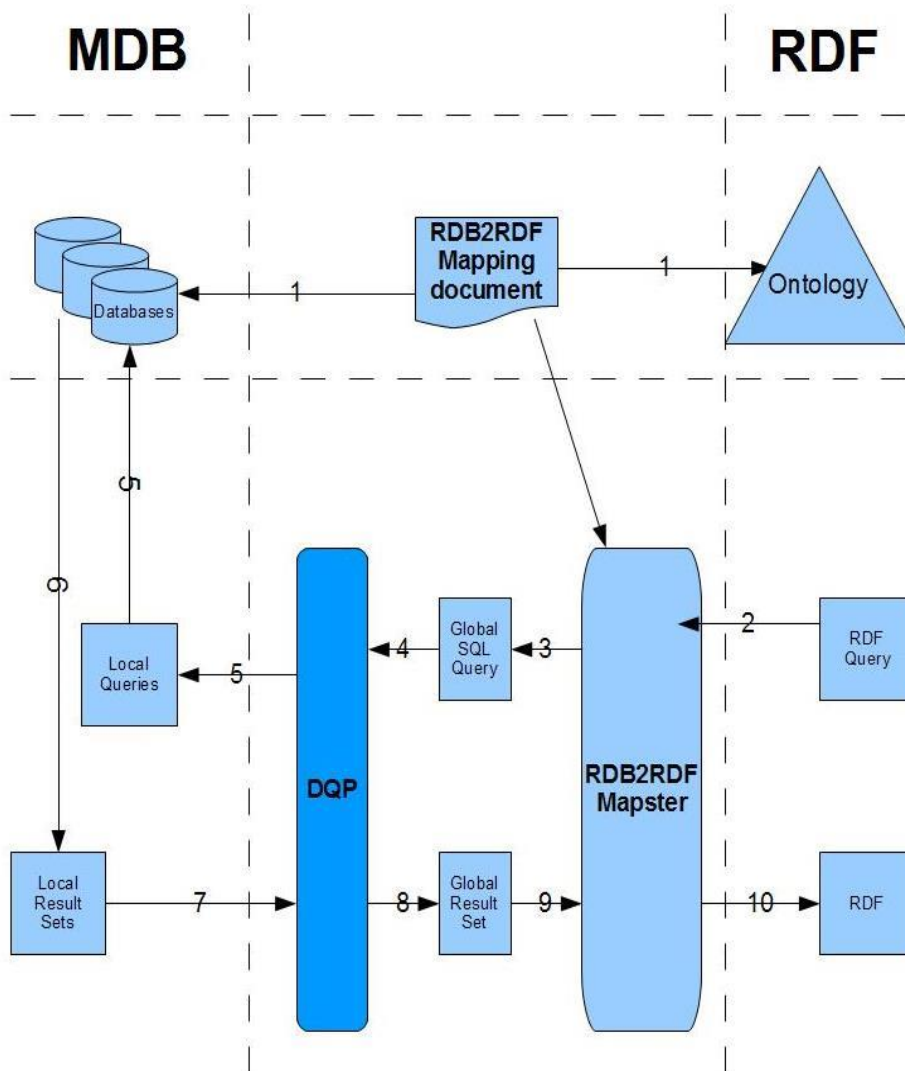


## • 3 Mapping Creation Steps

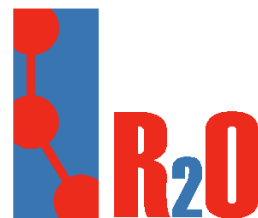
- Load Ontology
- Load Database(s)
- Create mapping

## • 2 Usage Modes

- Online mode (run time query execution)
- Offline mode (materialized RDF dump)



- Mapping defined
- RDF Query Posed
- RDF Query to Global Query
- Global query is sent to DQP
- DQP distributes global queries into local queries and evaluate them
- Databases return local results
- Local results sent to DQP
- DQP integrates the results into global result
- Global result is sent to be transformed into RDF
- RDF documents containing RDF instances



Boris Villazón-Terrazas, Freddy Priyatna  
bvillazon@fi.upm.es, fpriyatna@delicias.dia.fi.upm.es  
Ontology Engineering Group. Laboratorio de Inteligencia Artificial  
Departamento de Inteligencia Artificial  
Facultad de Informática  
Universidad Politécnica de Madrid