R₂O/ODEMapster Engine and Linked data Web service

Carlos Buil-Aranda, Boris Villazón-Terrazas, Oscar Corcho and Asunción Gómez-Pérez,
{cbuil, bvillazon, ocorcho, asun}@fi.upm.es
Ontology Engineering Group, Laboratorio de Inteligencia Artificial
Departamento de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid

WWW09 - Madrid, Spain

Speaker: Carlos Buil-Aranda
Date: 24/04/2009
• Introduction

• $R_2O$

• ODEMapster

• NeOn Toolkit plug in

• Service Based Access to provide linked data
Existing approaches

Lifting XML Schemas to Ontologies – The concept finder algorithm
Philipp Kunfermann, Christian Drumm
SAP Research Center CEC Karlsruhe
SAP AG

Approach 1
Approach 2
Approach 3

new ontology
existing ontology
• Introduction
• \( R_2O \)
• ODEMapster
• NeOn Toolkit plug in
• Service Based Access for providing linked data
• **R₂O** is an extensible, fully declarative language to describe mappings between relational database schemas and ontologies.

• Out of scope: to create an ontology that reflects the DB schema.
ConceptMap definition

\[ \mathcal{E}_M(C') = [f_{C}^{Id}, e_{C}^{Cond}, e_{C}^{Reun}] \]

BNF:

```
conceptmapping-definition ::= conceptmap-def name
  identified-by+
    (uri-as selector)?
    (applies-if cond-exp)?
    (joins-via concept-join-exp)?
    documentation?
    (described-by propertymap-def)*

identified-by ::= identified-by literal

concept-join-exp ::= (join-exp conceptJoinOper cond-exp)?

conceptJoinOper ::= join | union | difference
```

Example:

```
<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>
```
\[ E_M(A) = [C, e_A^{Cond}, e_A^{Reun}, f_A^{Trf}] \]

**BNF:**

\[
\text{attributemap-def} ::= \text{attributemap-def name} \\
\text{use-dbcoll} ::= \text{use-dbcoll literal} \\
\text{selector} ::= \text{selector (applies-if cond Expr)?} \\
\text{newobj-type} ::= \text{newobject-type literal} \\
\text{to-concept} ::= \text{to-concept literal}
\]

**Example:**

```xml
<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>
```
\[ \mathcal{E}_M(R) = [C^O, C^D, e^C_{\text{Cond}}, e^C_{\text{Reun}}] \]

**BNF:**

```
relationmap-def ::= relationmap-def to-concept
  (applies-if cond-expr)?
  (joins-via relation-join-expr)?
relation-join-expr ::= join (join-expr cond-expr)?
to-concept ::= to-concept literal
```

**Example:**

```xml
<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>
  </to-concept>
</relationmap-def>
```
• Introduction

• R\textsubscript{2}O

• ODEMapster

• NeOn Toolkit plug in

• Service Based Access to provide linked data
The ODEMapster processor generates Semantic Web instances from relational instances based on the mapping description expressed in the R2O document.

- Batch process/materialization: DB records migrated to the ontology.
- On demand/virtualization: Querying the DB in terms of ontological terms.
• Introduction

• $\text{R}_2\text{O}$

• ODEMapster

• NeOn Toolkit plug in

• Service Based Access to provide linked data
Create a DB to Ontology mapping

1. Select Mappings
2. Click new Mapping
3. Select Database
4. Mapping File created
5. Mapping Editor opens
Create a DB to Ontology mapping

Select Mappings
Click new Mapping
Select Database
Mapping File created
Mapping Editor opens
Create a DB to Ontology mapping

Select Mappings → Click new Mapping → Select Database → Mapping File created → Mapping Editor opens
Create a DB to Ontology mapping

1. Select Mappings
2. Click new Mapping
3. Select Database
4. Mapping File created
5. Mapping Editor opens
Create a DB to Ontology mapping

Select Mappings → Click new Mapping → Select Database → Mapping File created → Mapping Editor opens
Create attribute mapping – Constant Operation

Select Field → Drag Field → Drop Field → Mapping established

View mappings:
- All
- None
- Constant
- Concat
- Get substring

Attribute operations:
- Constant
- Concat
- Get delimited

Relation operations:
- Smaller
- Equal
- Greater

Information:
- Constant

Maps one field to an attribute.
Select the field with the mouse, then drag the fields to the target attribute.
Create attribute mapping – Constant Operation

Select Field → Drag Field → Drop Field → Mapping established

- **Attribute operators**: Constant, Smaller, Equal, Greater
- **Relation operations**: Concat, Get delimited

Maps one field to an attribute. Select the field with the mouse, then drag the fields to the target attribute.

- **Database**:
  - Type filter text
  - Delete mapping
  - Scientific group

- **Ontology**:
  - Group
    - hasId
    - hasMeta
    - hasNameScientific
    - includesOrder
    - includesFamily
    - includesSpecies
  - Species
    - hasId
    - hasNameScientific
    - hasCodeTax
    - hasCodeAlpha3
    - hasNameEN
Create attribute mapping – Constant Operation

Select Field ➔ Drag Field ➔ Drop Field ➔ Mapping established

- **View mappings**
  - All
  - None
  - Const
    - Smaller
  - Concat
    - Equal
  - Get substring
    - Greater

- **Attribute operations**
  - Constant
  - Concat
  - Equal
  - Get delimited
  - Greater

- **Relation operations**

- **Information**
  - Constant
  - Maps one field to an attribute.
  - Select the field with the mouse, then drag the fields to the target attribute.

- **Database**
  - type filter text
  - Delete mapping

- **Ontology**
  - type filter text

- **Classes**
  - group
    - hasId
    - hasMeta
  - scientific_group
    - hasNameScientific
  - includedOrder
  - includesFamily
  - includesSpecies
  - species
    - hasId
    - hasMeta
    - hasNameScientific
    - hasCodeTax
    - hasCodeAlpha3
    - hasNameEN
Create attribute mapping – Constant Operation

Select Field > Drag Field > Drop Field > Mapping established

- **Attribute operations**:
  - Constant
  - Smaller
  - Concat
  - Equal
  - Greater
  - Get delimited

- **Relation operations**:

- **Information**:
  - Constant: Maps one field to an attribute.
  - Select the field with the mouse, then drag the fields to the target attribute.
Create attribute mapping - Concat operation

Select Concat

Select 1st Field

Select 2nd Field

Drag Field

Drop Field

Mapping established

Concatenates two fields.

Select the first field with the mouse, then hold down ctrl and select the second field. Now drag the fields to the mapping established.
Create attribute mapping - Concat operation

Select Concat > Select 1st Field > Select 2nd Field > Drag Field > Drop Field > Mapping established

Select: 1st Field, 2nd Field
Drag Field
Drop Field
Mapping established

Concatenates two fields.
Select the first field with the mouse, then hold down ctrl and select the second field. Now drag the fields to the
Create attribute mapping - Concat operation

Select Concat

Select 1st Field

Select 2nd Field

Drag Field

Drop Field

Mapping established

Select

Concat

Select 1st Field

Select 2nd Field

Drag Field

Drop Field

Mapping established

Select the first field with the mouse, then hold down ctrl and select the second field. Now drag the fields to the.
Create attribute mapping - Concat operation

Select Concat → Select 1st Field → Select 2nd Field → Drag Field → Drop Field → Mapping established

- **Select Concat**
- **Select 1st Field**
- **Select 2nd Field**
- **Drag Field**
- **Drop Field**
- **Mapping established**

### Attribute operations
- **Constant**
- **Concat**
- **Equal**
- **Get delimited**
- **Greater**

### Relation operations
- **Concatenates two fields**
- Select the first field with the mouse, then hold down ctrl and select the second field. Now drag the fields to the mapping established.
Create attribute mapping - Concat operation

Select Concat
Select 1st Field
Select 2nd Field
Drag Field
Drop Field
Mapping established

Concatenates two fields.
Select the first field with the mouse, then hold down ctrl and select the second field. Now drag the fields to the mapping established.
Create attribute mapping - Concat operation

- Select Concat
- Select 1st Field
- Select 2nd Field
- Drag Field
- Drop Field

Mapping established

Concatenates two fields.

Select the first field with the mouse, then hold down ctrl and select the second field. Now drag the fields to the right.
Query the ontology instances

Select Mapping & Click on Query

Query Editor Opens & Select ontology elements

Click on Execute query

[Image of a computer screen showing an ontology editor with a menu option for 'Query R2O Mapping']
Query the ontology instances

Select Mapping & Click on Query

Query Editor Opens & Select ontology elements

Click on Execute query

RDF output

Selection
- Instances only
- Instances with attributes
- Instances with relations
- Instances with both
- Custom selection

Concepts

Attributes and Relations

hasMeta

hasNameScientific

© R2O & ODEMapster
Carlos Buil-Aranda, Oscar Corcho, Asunción Gómez-Pérez, Boris Villazón-Terrazas
Query the ontology instances

Select Mapping & Click on Query

Query Editor Opens & Select ontology elements

Click on Execute query

RDF output

Execute query

Save output

Stop execution

© R2O & ODEMapster
Carlos Buil-Aranda, Oscar Corcho, Asunción Gómez-Pérez, Boris Villazón-Terrazas
• Introduction

• R₂O

• ODEMapster

• NeOn Toolkit plug in

• Service Based Access to provide linked data
Web Service Access

• Current approach needs high amount of time
  • Batch mode may be running for a long time
  • Even on demand mode depending on the database length

• WS-DAI Web services specification allows to make synchronous and asynchronous calls

• OGSA-DAI
  • Framework that allows to access, integrate, transform and deliver distributed and heterogeneous sources of data
  • Implements part of the WS-DAI specification
  • Resources, activities and workflows

• Output: stream of RDF data
New Resource (ODEMapster Engine)
R$_2$O and ODEMapster

Carlos Buil-Aranda, Boris Villazón-Terrazas, Oscar Corcho and Asunción Gómez-Pérez,

{cbuil, bvillazon, ocorcho, asun}@fi.upm.es
Ontology Engineering Group, Laboratorio de Inteligencia Artificial
Departamento de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid

WWW09 - Madrid, Spain