

Publishing Linked Data from spreadsheets

Boris Villazón-Terrazas, Oscar Corcho

Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo sn, 28660 Boadilla del Monte, Madrid

<http://www.oeg-upm.net>

bvillazon@fi.upm.es

Phone: 34.91.3366605, Fax: 34.91.3524819

Slides available at: <http://www.slideshare.net/boricles/>

Acknowledgements:

Work distributed under the license Creative Commons Attribution-Noncommercial-Share Alike 3.0

Specification – Spreadsheet about statistics

• Unemployment

Mercado laboral Paro registrado

Paro registrado por CCAA y periodo.
Unidades:miles de personas (media anual)

	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
Andalucía	587,5	544	493,7	475,8	465,9	483,9	478,5	462,3	456,5	477,3	492,3	602,9
Aragón	58,5	51,4	45,9	40,6	40	42,5	41,4	39,2	39,6	37,7	35,9	48,2
Principado de Asturias	79,7	72,1	64,3	62,1	58	59,1	60,4	61,6	57,6	53,4	50	52,8
Illes Balears	36,8	33,3	29,8	29,1	30,8	35,4	38,8	39,3	38,3	36,1	37,2	49,1
Canarias	133,7	123,2	109,6	106,6	108,9	112,7	119,4	132,1	133,4	130,4	133,7	175,1
Cantabria	31,8	28,3	24,6	23,3	23,4	24,5	25	25,6	24,2	21,8	20,6	25,4
Castilla y León	145,5	133,7	123,1	120,2	114,9	116,1	116,4	116,9	113,3	110	106,1	123,3
Castilla-La Mancha	108,3	97,6	88,6	87	86,6	88,9	91,4	93,5	94	91,6	91,7	115,5
Cataluña	322,5	272,5	232,5	213,8	222,1	252,5	264	270,4	265,5	260,8	256,4	333,7
Comunitat Valenciana	279,5	240	201,9	169,6	158,9	171	179,8	190,9	197,7	195,9	201,4	277,9
Extremadura	80,1	77,2	71,4	70,5	70,3	74	75,5	78,1	77,3	76,2	74,6	64,5
Galicia	201,7	186,3	168	163,4	159,5	168,4	174,7	176,9	172,1	161,3	150,5	162,5
Comunidad de Madrid	321,6	284	242,1	223,2	218,7	244,7	253,8	248,1	226	216,5	221,7	281
Región de Murcia	59,9	51,9	44,6	44	43,7	45,2	45,7	45,2	44,2	44,4	46,6	69,8
Comunidad Foral de Navarra	27,6	25,1	22,3	21,3	21,1	21,9	21,7	21,7	22,3	21	20,6	25,3
País Vasco	135,1	117,9	104	94,1	88,4	88,9	89,7	84,8	80,8	75,6	84,6	
Rioja, La	12	10,9	9,6	9,1	8,9	9,7	9,5	9,8	10	10,2	10	12,8
Ceuta	4,7	4,4	4,1	4,4	4,7	4,8	5,2	6	6,5	7,4	6,9	7,4
Melilla	5,3	5,6	5,3	5,3	5,5	5,6	6,1	6,2	6,5	6,7	6,9	7,9

Notas:

1.- El INEM ha estimado series de paro registrado para los datos anteriores a mayo de 2005.

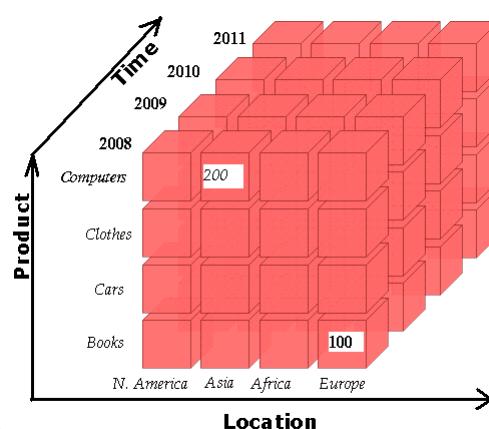


Data cubes

- A **cube** can be considered a multidimensional extension of 2-D tables (as in geometry a cube is a three-dimensional extension of a square)
- The term **hypercube** is sometimes used, especially for data with more than three dimensions
- Each **dimension** represents some attribute in the database (such as sales, profits, expenses...)
- The **cells** in the data cube represent the measure of interest. For example, they could contain a count for the number of times that attribute combination occurs in the database, or the minimum, maximum, sum or average value of some attribute
- Queries are performed on the cube to retrieve decision support information

3

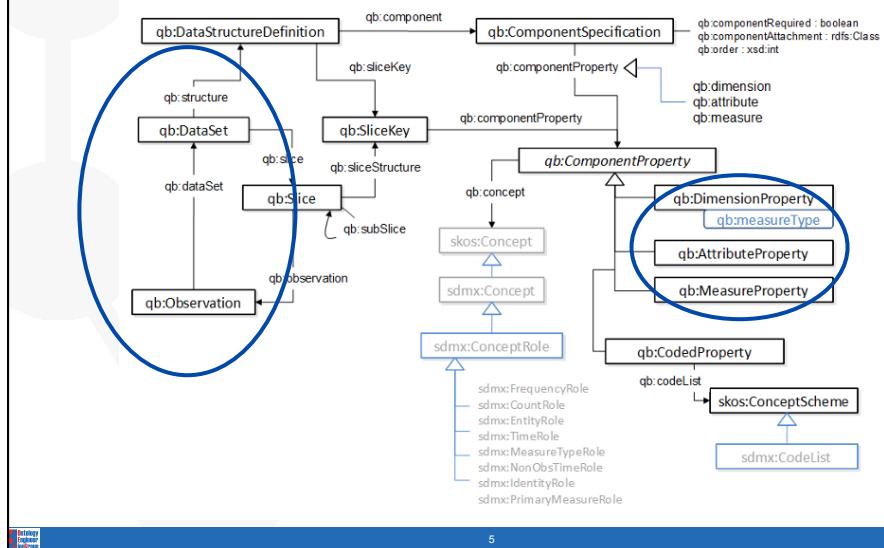
Data cubes



4

Modelling

- RDF Data Cube

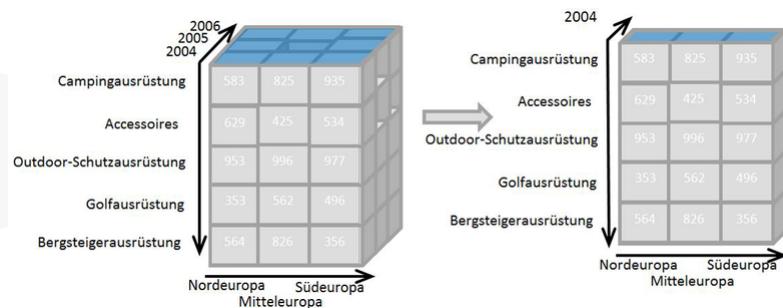


OLAP Cube

- OLAP stands for **Online Analytical Processing**
- Computer-based technique for analyzing business data in the search for business intelligence
- The elements of a dimension can be organized as a **hierarchy**, where a parent member summarizes its children
- Parent elements can further be **aggregated** as the children of another parent
- Different **operations** to facilitate analysis, aligning the data content with a familiar visualization

Common OLAP Operations

- **Slice** is the act of picking a rectangular subset of a cube by choosing a single value for one of its dimensions, creating a new cube with one fewer dimension

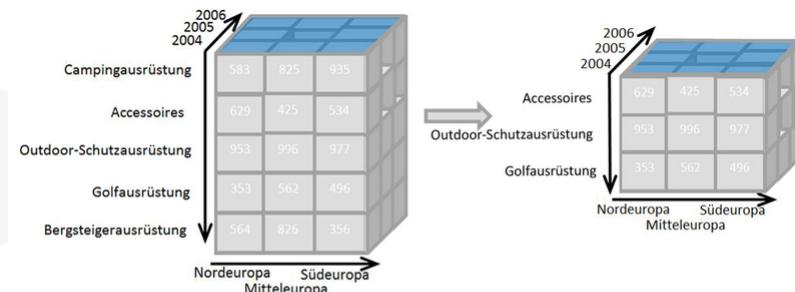


http://en.wikipedia.org/wiki/OLAP_cube

7

Common OLAP Operations

- **Dice** produces a subcube by allowing the analyst to pick specific values of multiple dimensions

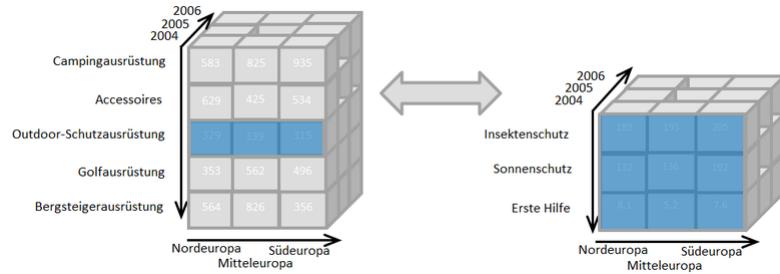


http://en.wikipedia.org/wiki/OLAP_cube

8

Common OLAP Operations

- *Drill Down/Up* allows the user to navigate among levels of data ranging from the most summarized (up) to the most detailed (down)

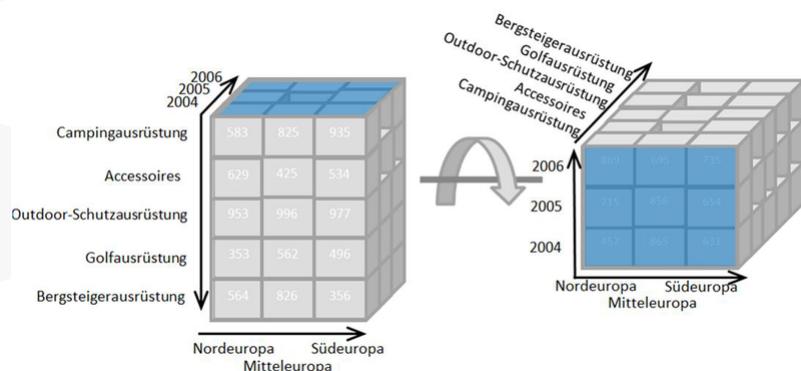


http://en.wikipedia.org/wiki/OLAP_cube

9

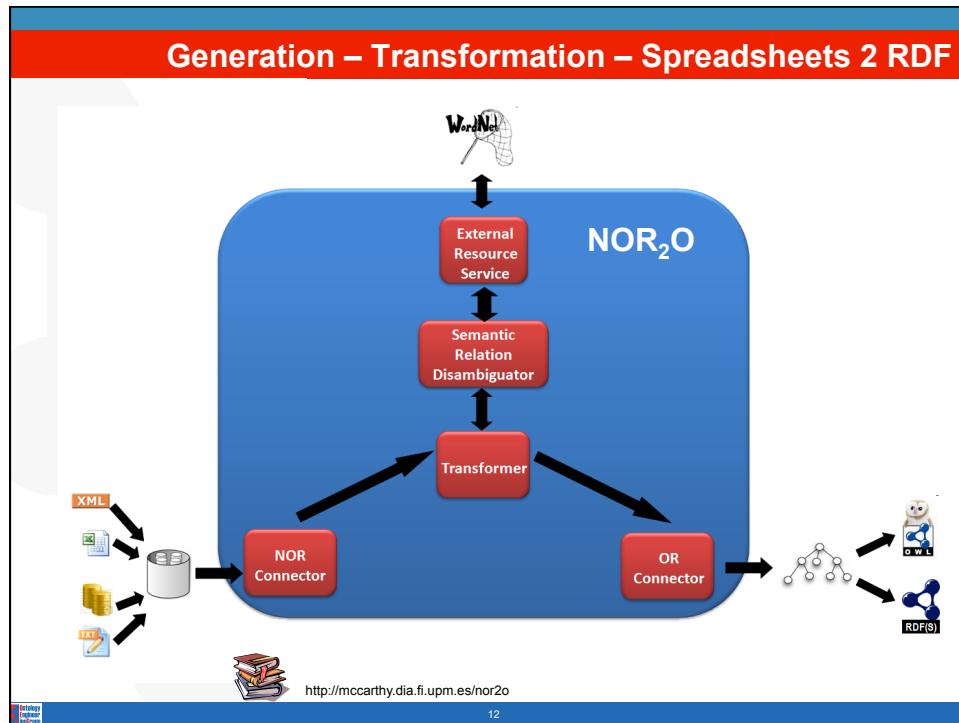
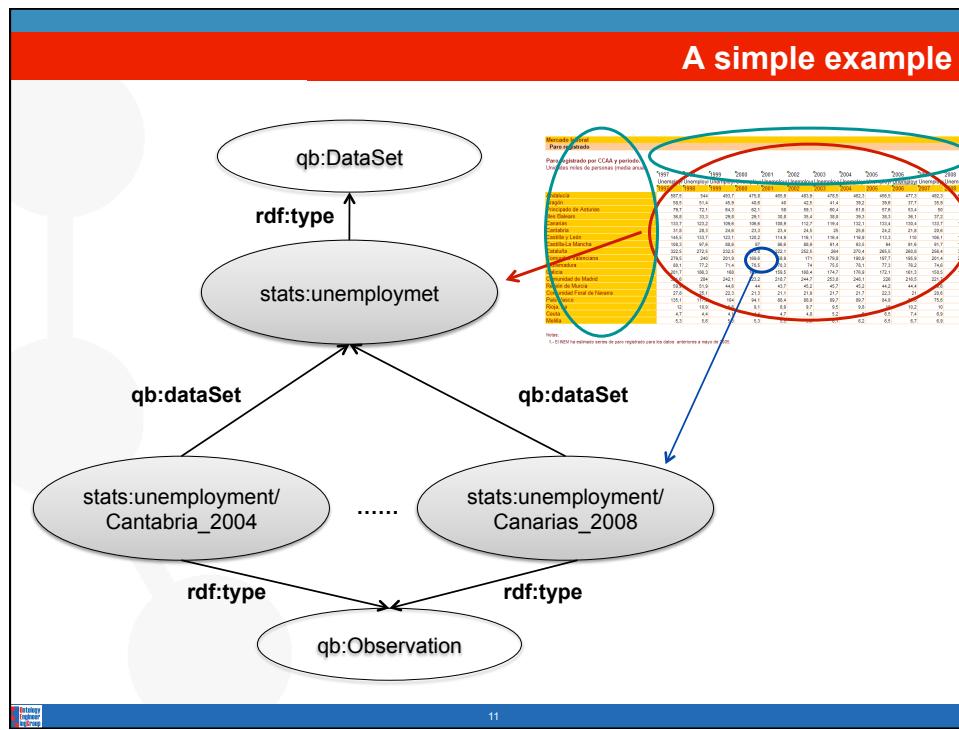
Common OLAP Operations

- *Pivot* allows an analyst to rotate the cube in space to see its various faces



http://en.wikipedia.org/wiki/OLAP_cube

10

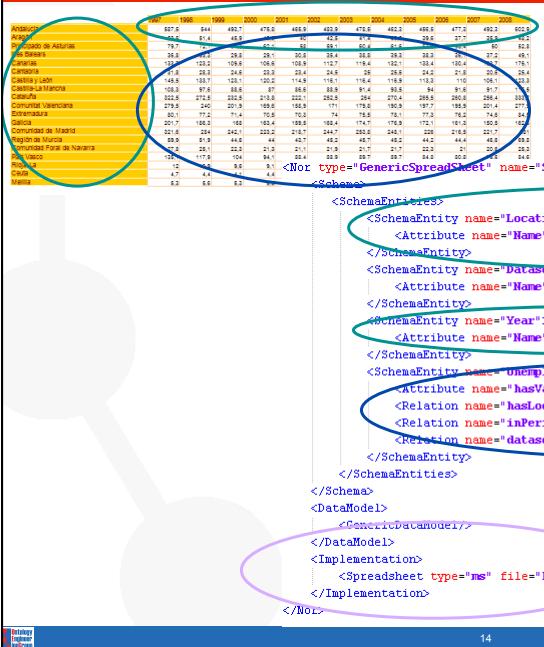


NOR₂O

- 3 configuration files
 - nor.xml, to describe the resource
 - prnor.xml to describe the transformation
 - or.xml to describe the ontology

13

NOR₂O – nor.xml



The screenshot shows a Microsoft Excel spreadsheet with data for Spanish employment. The columns represent years from 1990 to 2009, and the rows represent locations. A yellow oval highlights the first few rows and columns, while a blue oval highlights the last few rows and columns. A purple oval highlights the XML code below.

```
<Nor>
  <GenericSpreadSheet type="ms" file="Paro.xls"/>
</Nor>
```

```
<Schema>
  <SchemaEntities>
    <SchemaEntity name="Location">
      <Attribute name="Name" valueFrom="pcaxis.[A9:A25]" type="string"/>
    </SchemaEntity>
    <SchemaEntity name="Dataset">
      <Attribute name="Name" valueFrom="pcaxis.[B7:M7]" type="string"/>
    </SchemaEntity>
    <SchemaEntity name="Year">
      <Attribute name="Name" valueFrom="pcaxis.[B6:M6]" type="string"/>
    </SchemaEntity>
    <SchemaEntity name="unemploymentByLocationInPeriod" type="Nary">
      <Attribute name="hasValue" valueFrom="pcaxis.[B9:M25]" type="string"/>
      <Relation name="hasLocation" usingSpreadSheetColumn="A" destination="Location"/>
      <Relation name="inPeriod" usingSpreadSheetRow="6" destination="Year"/>
      <Relation name="dataset" usingSpreadSheetRow="7" destination="Dataset"/>
    </SchemaEntity>
  </SchemaEntities>
  <DataModel>
    <GenericDataModel/>
  </DataModel>
  <Implementation>
    <Spreadsheet type="ms" file="Paro.xls"/>
  </Implementation>
</Nor>
```

14

NOR₂O – prnor.xml

- How to transform the resource

```
<Prnor identifier="PR-NOR-CLLD-01" transformationApproach="Population">
  <Class from="UnemploymentByLocationInPeriod" identifier="cube:Observation"> ←
    <ObjectProperty from="hasLocation" to="property:geoArea"/>
    <ObjectProperty from="inPeriod" to="dimension:refPeriod"/>
    <ObjectProperty from="dataset" to="cube:DataSet"/>
    <DataTpeProperty from="hasValue" to="property:Unemployment" type="http://www.w3.org/2001/XMLSchema#double"/>
    <Individual from="UnemploymentByLocationInPeriod" identifier="unemployment_.[hasLocation]_[inPeriod]"/>
  </Class>
  <Class from="Dataset" identifier="cube:DataSet"> ←
    <DataTpeProperty from="Name" to="nombre" type="http://www.w3.org/2001/XMLSchema#string"/>
    <Individual from="Dataset" identifier="Name"/>
  </Class>
  <Class from="Year" identifier="Year"> ←
    <DataTpeProperty from="Name" to="nombre" type="http://www.w3.org/2001/XMLSchema#string"/>
    <Individual from="Year" identifier="year_.[Name]"/>
  </Class>
  <Class from="Location" identifier="Region"> ←
    <DataTpeProperty from="Name" to="nombre" type="http://www.w3.org/2001/XMLSchema#string"/>
    <Individual from="Location" identifier="geoes:ComunidadAutonomia_[Name]"/>
  </Class>
</Prnor>
```

Reusing GeoLinkedData URIs!

NOR₂O – or.xml

```
<Or name="Unemployment Index" ontologyURI="http://stats.ull.es/resource/" →
  ontologyFile="paro.rdf" implementation="OWL" alreadyExist="no" separator="">
  <prefix name="geose" uri="http://geo.linkeddata.es/resource/"/>
  <prefix name="cube" uri="http://purl.org/linked-data/cube#"/>
  <prefix name="property" uri="http://stats.ull.es/property/"/>
  <prefix name="dimension" uri="http://purl.org/linked-data/sdmx/2009/dimension#"/>
  <prefix name="rdf" uri="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <prefix name="year" uri="http://reference.data.gov.uk/id/year/"/>
</Or>
```

QB4OLAP

- QB4OLAP adds to Data Cube vocabulary (QB) the capability of representing:
 - dimension levels
 - level members
 - rollup relations between levels and level members
 - associating aggregate functions to measures
- It allows to represent OLAP cubes in RDF
- It allows to implement OLAP operators (such as Roll-up, Slice, and Dice) as SPARQL queries directly on this RDF representation



<http://publishing-multidimensional-data.googlecode.com/git/index.html>

17

NOR₂O – Exercise

- **What you need:**
 - Java 1.5 o superior
- **What to do:**
 - Download NOR2O
<http://oeg-dev.dia.upm.es/nor2o/#download>
 - Extract the files in your workspace folder.
 - The current distribution comes with three examples in folders **example1**, **example2** and **example3**.
 - Files in the wiki
 - “Paro” case: Unemployment for each city by date, genre and age
 - “Atlas” case: Number of health centers for each city by year



18

NOR₂O – Exercise

- **What you need:**
 - Java 1.5 o superior

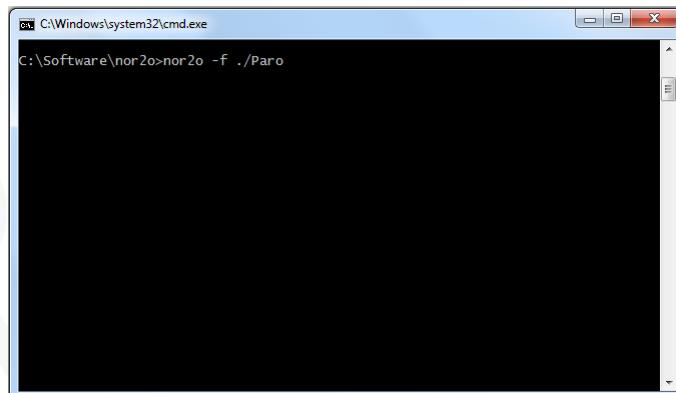
- **What to do:**

- Download NOR2O
<http://oeg-dev.dia.fi.upm.es/nor2o/#download>
- Extract the files in your workspace folder.
- The current distribution comes with three examples in folders example1, example2 and example3.
- Files in the wiki
 - “Paro” case: Unemployment for each city by date, genre and age
 - “Atlas” case: Number of health centers for each city by year
- Run nor2o from the command line
- Wait until it finishes and check the generated RDF file

19

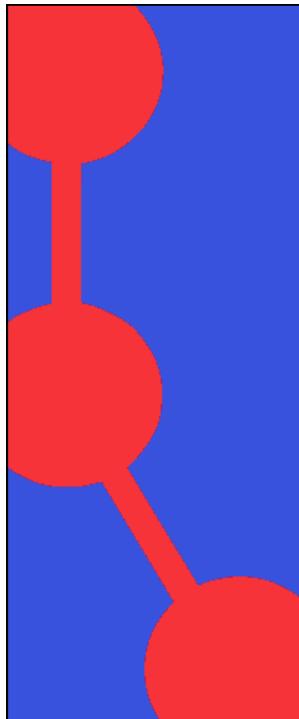
NOR₂O – Exercise

- Run nor2o from the command line



```
cmd C:\Windows\system32\cmd.exe
C:\Software\nor2o>nor2o -f ./Paro
```

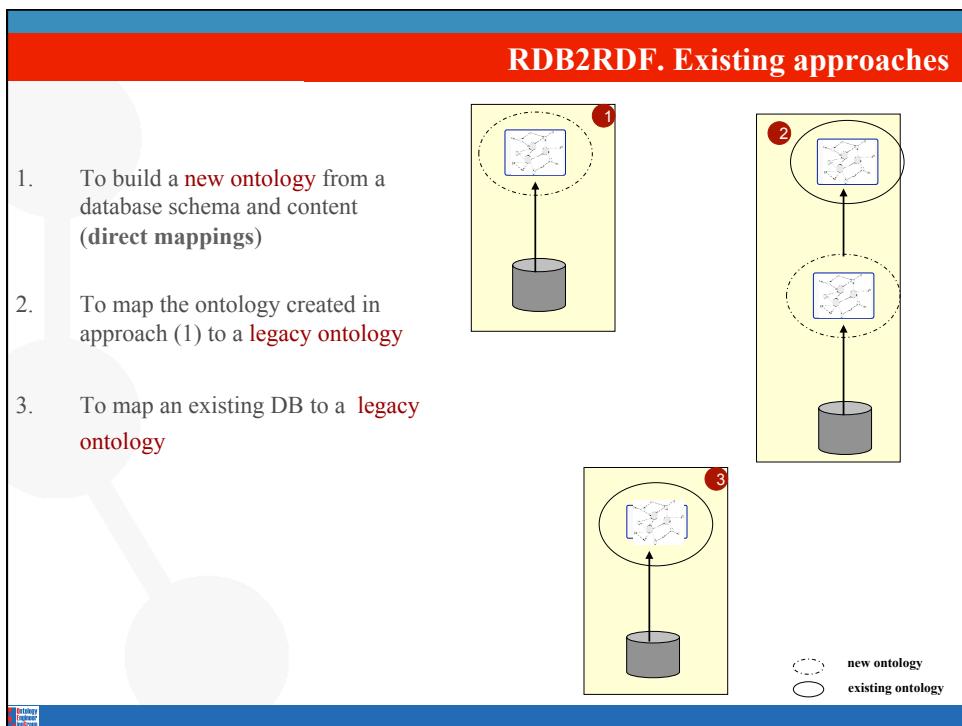
20



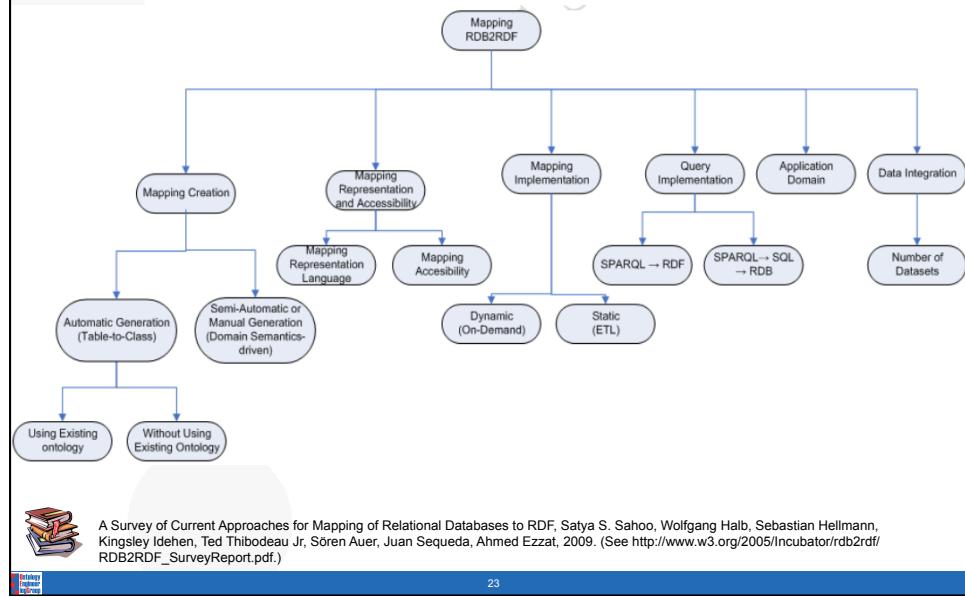
Publishing Linked Data from RDB

Boris Villazón-Terrazas. Oscar Corcho
Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo sn, 28660 Boadilla del Monte, Madrid
<http://www.oeg-upm.net>
ocorcho@fi.upm.es
Phone: 34.91.3366605, Fax: 34.91.3524819

Acknowledgements: Freddy Priyatna, Jan Schulte, Richard Cyganiak and many others that we may have omitted.



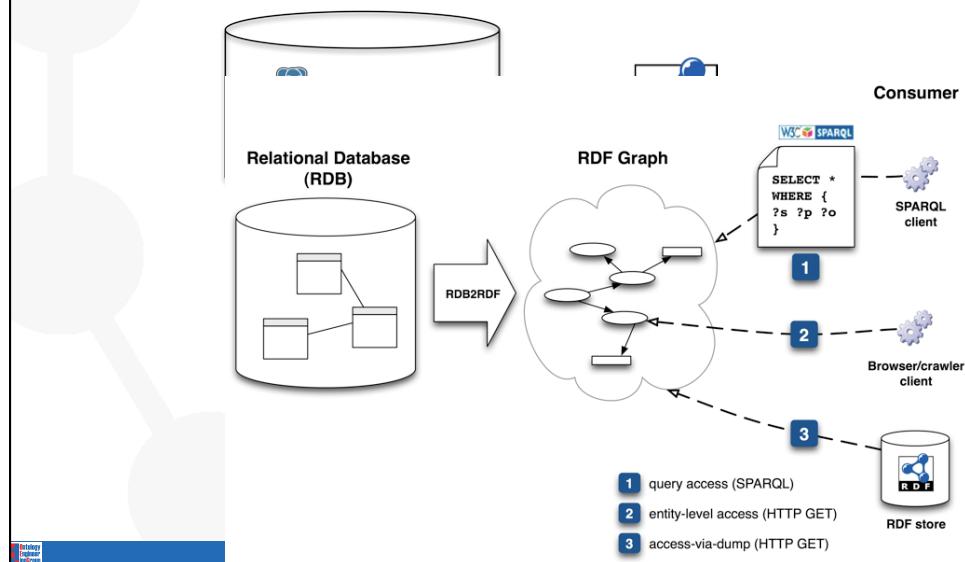
A Survey of Current Approaches for Mapping of Relational Databases to RDF



23

Transformation – RDB2RDF

- A majority of dynamic Web content is backed by relational databases (RDB), and so are many enterprise systems.



Direct Mapping

- The direct mapping defines an RDF Graph representation of the data in an RDB. The direct mapping takes as input an RDB (data and schema), and generates an RDF graph that is called the direct graph.

```
CREATE TABLE "Addresses" (
    "ID" INT, PRIMARY KEY("ID"),
    "city" CHAR(10),
    "state" CHAR(2)
)

CREATE TABLE "People" (
    "ID" INT, PRIMARY KEY("ID"),
    "fname" CHAR(10),
    "addr" INT,
    FOREIGN KEY("addr") REFERENCES "Addresses"("ID")
)

INSERT INTO "Addresses" ("ID", "city", "state") VALUES (18, 'Cambridge', 'MA')
INSERT INTO "People" ("ID", "fname", "addr") VALUES (7, 'Bob', 18)
INSERT INTO "People" ("ID", "fname", "addr") VALUES (8, 'Sue', NULL)
```

25

Direct Mapping - example

People		
PK		→ Address(ID)
ID	fname	addr
7	Bob	18
8	Sue	NULL

Addresses		
PK		
ID	city	state
18	Cambridge	MA

```
@base <http://foo.example/DB/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

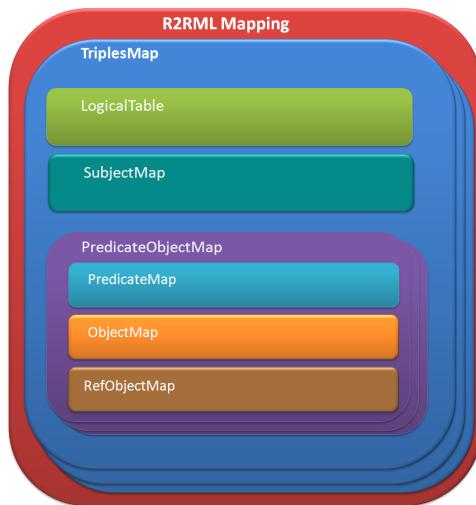
<People[ID=7]> rdf:type <People> .
<People[ID=7]> <People#ID> 7 .
<People[ID=7]> <People#fname> "Bob" .
<People[ID=7]> <People#addr> 18 .
<People[ID=7]> <People#ref-addr> <Addresses[ID=18]> .
<People[ID=8]> rdf:type <People> .
<People[ID=8]> <People#ID> 8 .
<People[ID=8]> <People#fname> "Sue" .

<Addresses[ID=18]> rdf:type <Addresses> .
<Addresses[ID=18]> <Addresses#ID> 18 .
<Addresses[ID=18]> <Addresses#city> "Cambridge" .
<Addresses[ID=18]> <Addresses#state> "MA" .
```

26

The W3C RDB2RDF Working Group

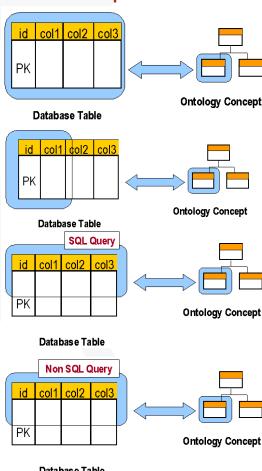
- Created in 2007
- W3C Recommendations in September 2012
 - R2RML: RDB to RDF Mapping Language - <http://www.w3.org/TR/r2rml/>
 - Direct Mapping - <http://www.w3.org/TR/rdb2rdf-mapping/>
 - R2RML and Direct Mapping Test Cases - <http://www.w3.org/2001/sw/rdb2rdf/test-cases/>
 - RDB2RDF Implementation Report - <http://www.w3.org/2001/sw/rdb2rdf/implementation-report/>
 - R2RML mappings are themselves expressed as RDF graphs and written down in Turtle syntax.



27

R2O (Relational-to-Ontology) Language

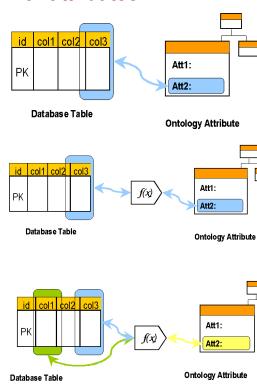
For concepts...



A view maps exactly one concept in the ontology.

One or more concepts can be extracted from a single data field (not in INF).

For attributes...



A column in a database view maps directly an attribute or a relation.

A column in a database view maps an attribute or a relation after some transformation.

A set of columns in a database view map an attribute or a relation.

27

OEG's background knowledge in RDB2RDF

- R2O and ODEMMapster
 - GaV wrapper generation (no mediators)
 - Syntactic sugar for the generation of SQL queries.
 - Simple use of this language and processor in the domains of fund finding, cultural information, and fisheries.
 - NeOn Toolkit plugin for common mappings

The screenshot shows the RDB2RDF interface. On the left, there is a 'Mapping Navigator' pane listing various database tables and their columns. In the center, there is a 'New mapping' dialog box with sections for 'Attribute operations' (Contains, Greater, Smaller, Constant, Equal, Contains, If is A, Get deleted, Greater) and 'Relation operations' (Contains, Greater, Smaller, Equal, If is A, Get deleted). Below the dialog is a 'Database' section showing tables like t1_organizer, t1_company, and t1_geography. To the right, there is a 'Triple Editor' pane displaying a graph of triples with nodes and edges. At the bottom, there is a 'Log' pane showing the command: `CREATE TABLE "Student" ("ID" integer, "Name" varchar(50), PRIMARY KEY ("ID")); INSERT INTO "Student" ("ID", "Name") VALUES(10,'Venus');`.

Barrasa J, Corcho O, Gómez-Pérez A. (2004) R2O, an extensible and semantically based database-to-ontology mapping language. In: Proceedings of the Second Workshop on Semantic Web and Databases, SWDB 2004.

R2RML - example I

```
CREATE TABLE "Student" (
  "ID" integer,
  "Name" varchar(50),
  PRIMARY KEY ("ID")
);
INSERT INTO "Student" ("ID", "Name") VALUES(10,'Venus');
```

Student	
ID (PK) INTEGER	Name VARCHAR(50)
10	Venus

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.com/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@base <http://example.com/base/> .

<TriplesMap1>
  a rr:TriplesMap;

  rr:logicalTable [ rr:tableName "\"Student\""] ;

  rr:subjectMap [ rr:template "http://example.com/Student/{\"ID\"}/{\"Name\"}"; ];

  rr:predicateObjectMap
  [
    rr:predicate    rdf:type;
    rr:object       foaf:Person;
  ];
.
```

Subject	Predicate	Object	Graph
<http://example.com/Student/10/Venus>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://xmlns.com/foaf/0.1/Person>	

R2RML – example II - input

```
CREATE TABLE "Student" (
  "ID" integer,
  "Name" varchar(50),
  PRIMARY KEY ("ID")
);
INSERT INTO "Student" ("ID", "Name") VALUES(10,'Venus');
```

Student	
ID (PK) INTEGER	Name VARCHAR(50)
10	Venus

```
<TriplesMap1>
  a rr:TriplesMap;

  rr:logicalTable [ rr:tableName "\"Student\""; ];

  rr:subjectMap [ rr:template "http://example.com/Student/{\"ID\"}/{\"Name\"}";
    rr:graph ex:PersonGraph;
  ];

  rr:predicateObjectMap [
    rr:predicate      rdf:type;
    rr:object        foaf:Person;
  ];

  rr:predicateObjectMap [
    rr:predicate      foaf:name;
    rr:objectMap      [ rr:column "\"Name\""]
  ];
```

Subject	Predicate	Object	Graph
<http://example.com/Student/10/Venus>	<http://xmlns.com/foaf/0.1/name>	"Venus"	<http://example.com/PersonGraph>
<http://example.com/Student/10/Venus>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://xmlns.com/foaf/0.1/Person>	<http://example.com/PersonGraph>

R2RML - example III - input

Student		
ID (PK) INTEGER	FirstName VARCHAR(50)	LastName VARCHAR(50)
10	Venus	Williams
11	Fernando	Alonso
12	David	Villa

Student_Sport	
ID_Student (PK) INTEGER	ID_Sport (PK) INTEGER
10	110
11	111
11	112
12	111

Sport	
ID (PK) INTEGER	Description VARCHAR(50)
110	Tennis
111	Football
112	Formulal

```
CREATE TABLE "Student" (
  "ID" integer PRIMARY KEY,
  "FirstName" varchar(50),
  "LastName" varchar(50)
);
CREATE TABLE "Sport" (
  "ID" integer PRIMARY KEY,
  "Description" varchar(50)
);
CREATE TABLE "Student_Sport" (
  "ID_Student" integer,
  "ID_Sport" integer,
  PRIMARY KEY ("ID_Student","ID_Sport"),
  FOREIGN KEY ("ID_Student") REFERENCES "Student"("ID"),
  FOREIGN KEY ("ID_Sport") REFERENCES "Sport"("ID")
);

INSERT INTO "Student" ("ID", "FirstName", "LastName") VALUES (10, 'Venus', 'Williams');
INSERT INTO "Student" ("ID", "FirstName", "LastName") VALUES (11, 'Fernando', 'Alonso');
INSERT INTO "Student" ("ID", "FirstName", "LastName") VALUES (12, 'David', 'Villa');

INSERT INTO "Sport" ("ID", "Description") VALUES (110, 'Tennis');
INSERT INTO "Sport" ("ID", "Description") VALUES (111, 'Football');
INSERT INTO "Sport" ("ID", "Description") VALUES (112, 'Formulal');

INSERT INTO "Student_Sport" ("ID_Student", "ID_Sport") VALUES (10, 110);
INSERT INTO "Student_Sport" ("ID_Student", "ID_Sport") VALUES (11, 111);
INSERT INTO "Student_Sport" ("ID_Student", "ID_Sport") VALUES (11, 112);
INSERT INTO "Student_Sport" ("ID_Student", "ID_Sport") VALUES (12, 111);
```

R2RML - example III - mapping

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ex: <http://example.com/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@base <http://example.com/base/> .

<TriplesMap1>
  a rr:TriplesMap;

  rr:logicalTable [ rr:tableName "\"Student\""; ] ;

  rr:subjectMap [ rr:template "http://example.com/student/{\"ID\"}"; ];

  rr:predicateObjectMap
  [
    rr:predicate ex:firstName ;
    rr:objectMap [ rr:column "\"FirstName\"" ]
  ];
  rr:predicateObjectMap
  [
    rr:predicate ex:lastName ;
    rr:objectMap [ rr:column "\"LastName\"" ]
  ];
```

33

R2RML - example III - mapping (cont.)

```
<TriplesMap2>
  a rr:TriplesMap;

  rr:logicalTable [ rr:tableName "\"Sport\""; ] ;

  rr:subjectMap [ rr:template "http://example.com/sport/{\"ID\"}"; ];

  rr:predicateObjectMap
  [
    rr:predicate ex:id ;
    rr:objectMap [ rr:column "\"ID\"" ]
  ];
  rr:predicateObjectMap
  [
    rr:predicate ex:description ;
    rr:objectMap [ rr:column "\"Description\"" ]
  ];
.

<LinkMap_1_2>
  a rr:TriplesMap;

  rr:logicalTable [ rr:tableName "\"Student_Sport\""; ];

  rr:subjectMap [ rr:template "http://example.com/student/{\"ID_Student\"}"; ];

  rr:predicateObjectMap [
    rr:predicate ex:plays ;
    rr:objectMap [ rr:template "http://example.com/sport/{\"ID_Sport\"}" ]
  ].
```

34

R2RML - example III - output

Subject	Predicate	Object	Graph
<http://example.com/student/10>	<http://example.com/lastName>	"Williams"	
<http://example.com/student/10>	<http://example.com/firstName>	"Venus"	
<http://example.com/student/12>	<http://example.com/lastName>	"Villa"	
<http://example.com/student/12>	<http://example.com/firstName>	"David"	
<http://example.com/student/11>	<http://example.com/lastName>	"Alonso"	
<http://example.com/student/11>	<http://example.com/firstName>	"Fernando"	
<http://example.com/sport/110>	<http://example.com/description>	"Tennis"	
<http://example.com/sport/110>	<http://example.com/id>	"110"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/sport/111>	<http://example.com/description>	"Football"	
<http://example.com/sport/111>	<http://example.com/id>	"111"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/sport/112>	<http://example.com/description>	"Formulal"	
<http://example.com/sport/112>	<http://example.com/id>	"112"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/student/10>	<http://example.com/plays>	<http://example.com/sport/110>	
<http://example.com/student/12>	<http://example.com/plays>	<http://example.com/sport/111>	
<http://example.com/student/11>	<http://example.com/plays>	<http://example.com/sport/112>	
<http://example.com/student/11>	<http://example.com/plays>	<http://example.com/sport/111>	

35

R2RML - example IV - input

Student		
ID (PK) INTEGER	FirstName VARCHAR(50)	LastName VARCHAR(50)
10	Venus	Williams
11	Fernando	Alonso
12	David	Villa

Student_Sport	
ID_Student (PK) INTEGER	ID_Sport (PK) INTEGER
10	110
11	111
11	112
12	111

Sport	
ID (PK) INTEGER	Description VARCHAR(50)
110	Tennis
111	Football
112	Formulal


```

CREATE TABLE "Student" (
    "ID" integer PRIMARY KEY,
    "FirstName" varchar(50),
    "LastName" varchar(50)
);
CREATE TABLE "Sport" (
    "ID" integer PRIMARY KEY,
    "Description" varchar(50)
);
CREATE TABLE "Student_Sport" (
    "ID_Student" integer,
    "ID_Sport" integer,
    PRIMARY KEY ("ID_Student", "ID_Sport"),
    FOREIGN KEY ("ID_Student") REFERENCES "Student"("ID"),
    FOREIGN KEY ("ID_Sport") REFERENCES "Sport"("ID")
);
INSERT INTO "Student" ("ID", "FirstName", "LastName") VALUES (10, 'Venus', 'Williams');
INSERT INTO "Student" ("ID", "FirstName", "LastName") VALUES (11, 'Fernando', 'Alonso');
INSERT INTO "Student" ("ID", "FirstName", "LastName") VALUES (12, 'David', 'Villa');

INSERT INTO "Sport" ("ID", "Description") VALUES (110, 'Tennis');
INSERT INTO "Sport" ("ID", "Description") VALUES (111, 'Football');
INSERT INTO "Sport" ("ID", "Description") VALUES (112, 'Formulal');

INSERT INTO "Student_Sport" ("ID_Student", "ID_Sport") VALUES (10, 110);
INSERT INTO "Student_Sport" ("ID_Student", "ID_Sport") VALUES (11, 111);
INSERT INTO "Student_Sport" ("ID_Student", "ID_Sport") VALUES (11, 112);
INSERT INTO "Student_Sport" ("ID_Student", "ID_Sport") VALUES (12, 111);

```

36

R2RML - example IV - mapping

```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ex: <http://example.com/>
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@base <http://example.com/base/> .

<TriplesMap1>
  a rr:TriplesMap;
  rr:logicalTable [ rr:sqlQuery """
    SELECT "Student"."ID" as ID,
           "Student"."FirstName" as FirstName,
           "Student"."LastName" as LastName,
           "Sport"."Description" as Description,
           "Sport"."ID" as Sport_ID
      FROM "Student","Sport","Student_Sport"
     WHERE "Student"."ID" = "Student_Sport"."ID_Student"
       AND "Sport"."ID" = "Student_Sport"."ID_Sport";
    """;
  rr:subjectMap [ rr:template "http://example.com/{ID}/{FirstName};{LastName}" ];
  rr:predicateObjectMap
  [
    rr:predicate ex:id ;
    rr:objectMap [ rr:column "ID" ]; ]
  rr:predicateObjectMap
  [
    rr:predicate ex:firstName ;
    rr:objectMap [ rr:column "FirstName" ]; ]
  rr:predicateObjectMap
  [
    rr:predicate ex:lastName ;
    rr:objectMap [ rr:column "LastName" ]; ]
  rr:predicateObjectMap
  [
    rr:predicate ex:plays ;
    rr:objectMap [ rr:template "http://example.com/{Sport_ID}/{Description}" ] ]
  .
```

37

R2RML - example IV - mapping (cont.)

```
<TriplesMap2>
  a rr:TriplesMap;

  rr:logicalTable [ rr:tableName  "\"Sport\""] ;

  rr:subjectMap [ rr:template "http://example.com/{"ID"}>{"Description"}"; ];

  rr:predicateObjectMap
  [
    rr:predicate ex:id ;
    rr:objectMap [ rr:column "\"ID\""]; ]
  ];

  rr:predicateObjectMap
  [
    rr:predicate ex:description ;
    rr:objectMap [ rr:column "\"Description\""] ]
  ];
  .
```

38

R2RML - example IV - output

Subject	Predicate	Object	Graph
<http://example.com/110/Tennis>	<http://example.com/description>	"Tennis"	
<http://example.com/110/Tennis>	<http://example.com/id>	"110"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/10/Venus;Williams>	<http://example.com/plays>	<http://example.com/110/Tennis>	
<http://example.com/10/Venus;Williams>	<http://example.com/lastName>	"Williams"	
<http://example.com/10/Venus;Williams>	<http://example.com/firstName>	"Venus"	
<http://example.com/10/Venus;Williams>	<http://example.com/id>	"10"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/111/Football>	<http://example.com/description>	"Football"	
<http://example.com/111/Football>	<http://example.com/id>	"111"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/12/David;Villa>	<http://example.com/plays>	<http://example.com/111/Football>	
<http://example.com/12/David;Villa>	<http://example.com/lastName>	"Villa"	
<http://example.com/12/David;Villa>	<http://example.com/firstName>	"David"	
<http://example.com/12/David;Villa>	<http://example.com/id>	"12"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/112/Formula1>	<http://example.com/description>	"Formula1"	
<http://example.com/112/Formula1>	<http://example.com/id>	"112"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/11/Fernando;Alonso>	<http://example.com/lastName>	"Alonso"	
<http://example.com/11/Fernando;Alonso>	<http://example.com/firstName>	"Fernando"	
<http://example.com/11/Fernando;Alonso>	<http://example.com/id>	"11"^^<http://www.w3.org/2001/XMLSchema#integer>	
<http://example.com/11/Fernando;Alonso>	<http://example.com/plays>	<http://example.com/111/Football>	
<http://example.com/11/Fernando;Alonso>	<http://example.com/plays>	<http://example.com/112/Formula1>	

39

R2RML processors

- <http://code.google.com/p/oeg-obdi/>
- <https://github.com/jpcik/morph>
- <https://github.com/boricles/morph>

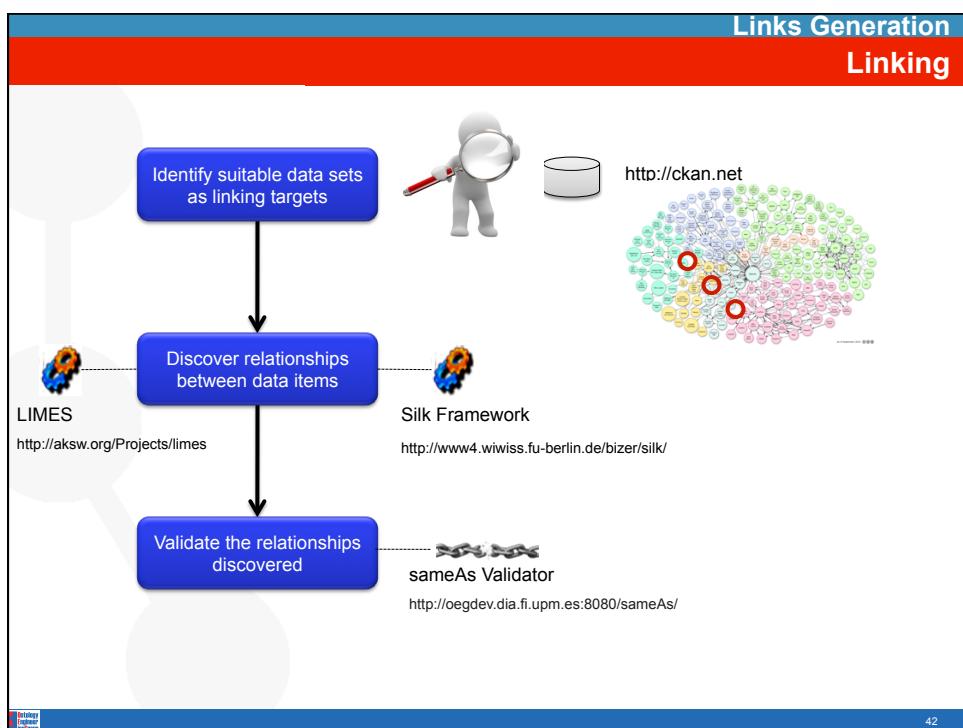
Table 2.: R2RML Processors.

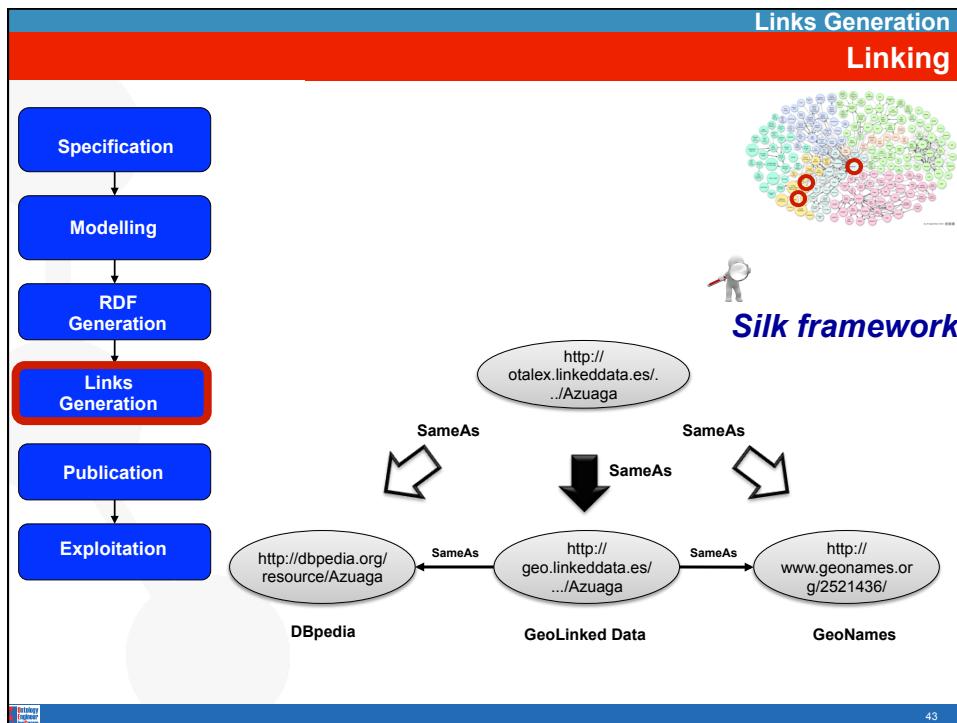
Name	Contact	Home
OpenLink Virtuoso	Ivan Mikhailov	http://virtuoso.openlinksw.com
RDF-RDB2RDF	Toby Inkster	https://metacpan.org/release/RDF-RDB2RDF
XSPARQL	Nuno Lopes	http://xsparql.deri.org
morph	Jean-Paul Calbimonte	https://github.com/jpcik/morph
ultrawrap	Juan Sequeda	http://www.capsenta.com/
db2triples	Julien Homo and Laurent Mazuel	https://github.com/antidot/db2triples



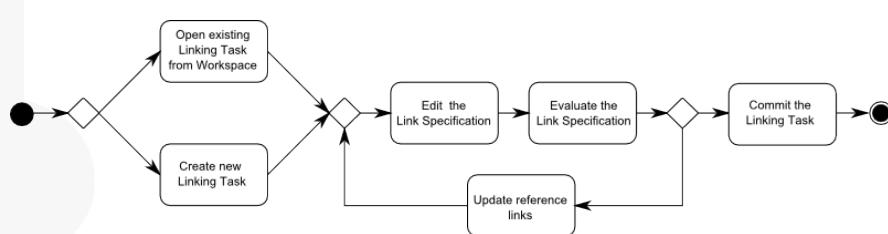
RDB2RDF Implementation Report. Boris Villazón-Terrazas, Michael Hausenblas.
<http://www.w3.org/2001/sw/rdb2rdf/implementation-report/>

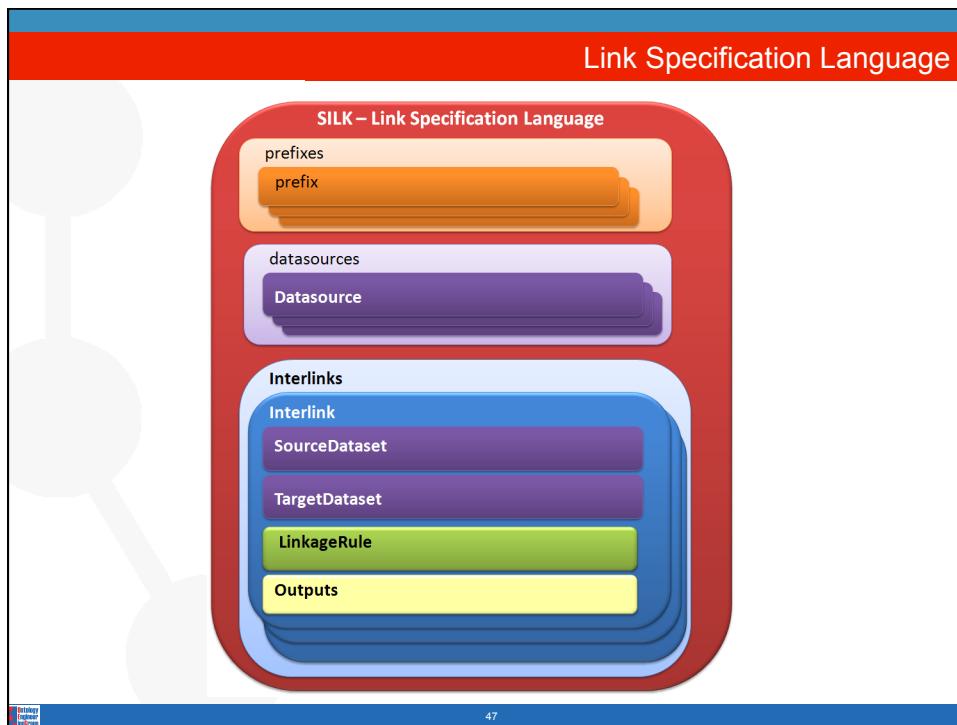
40





Links Generation Ejemplo- Linking





Workspace

Silk Workbench Workspace

FirstProject

- dbpedia
 - endpointURI: <http://dbpedia.org/sparql>
 - retryCount: 3
 - retryPause: 1000
- keggGene
- musicBrainz
- artistsTask
 - source: dbpedia
 - target: musicBrainz
 - source dataset: ?a rdf:type dbpediaowl:Artist .
 - target dataset: ?b rdf:type musicbrainz:Artist .
 - link type: owl:sameAs
- moviesTask

SecondProject

Workspace components

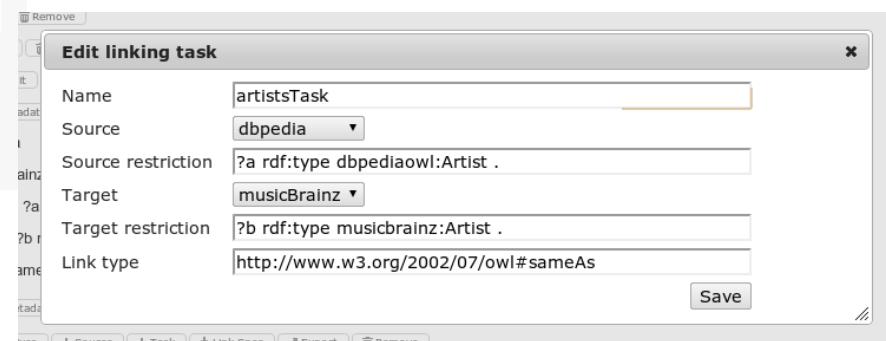
- A **project** holds the following information:
 - All URI prefixes which are used in the project.
 - A list of data sources
 - A list of linking tasks
- A **data source** holds all information that is needed by Silk to retrieve entities from it:



49

Workspace components: Linking Tasks

- A **linking task** consists of the following elements:
 - Metadata
 - A link specification
 - Positive and negative reference links



50

Linkage rules editor

- Clicking on the **OPEN** button opens the *Linkage Rules Editor* for a specific linking task

The screenshot shows the Silk Workbench interface with the 'Linkage rules editor' open. On the left, there's a sidebar with various property paths and operators listed under categories like 'Source: sides', 'Restriction: t', 'Target: drugbank', and 'Comparators'. The main area displays a 'PROPERTY PATHS:' section with a diagram showing nodes connected by arrows. One node is labeled 'Levenshtein distance (Comparator)'. Another node is labeled 'Lower case (Transformation)'. A third node is labeled 'Maximum (Aggregator)'. The diagram illustrates how these components are used together in a linkage rule.

Operators: transformations

A *transformation* can be used to normalize the values prior to comparison.

Function and parameters	Description
<code>removeBlanks</code>	Remove whitespace from a string.
<code>removeSpecialChars</code>	Remove special characters (including punctuation) from a string.
<code>lowerCase</code>	Convert a string to lower case.
<code>upperCase</code>	Convert a string to upper case.
<code>capitalize(allWords)</code>	Capitalizes the string i.e. converts the first character to upper case. If 'allWords' is set to true, all words are capitalized and not only the first character. By default 'allWords' is set to false.
<code>stem</code>	Apply word stemming to the string.
<code>alphaReduce</code>	Strip all non-alphabetic characters from a string.
<code>numReduce</code>	Strip all non-numeric characters from a string.
<code>replace(string search, string replace)</code>	Replace all occurrences of "search" with "replace" in a string.
<code>regexReplace(string regex, string replace)</code>	Replace all occurrences of a regex "regex" with "replace" in a string.

Operators: transformations

<code>stripPrefix</code>	Strip the prefix from a string.
<code>stripPostfix</code>	Strip the postfix from a string.
<code>stripUriPrefix</code>	Strip the URI prefix (e.g. <code>http://dbpedia.org/resource/</code>) from a string.
<code>concat</code>	Concatenates strings from two inputs.
<code>logarithm([base])</code>	Transforms all numbers by applying the logarithm function. Non-numeric values are left unchanged. If base is not defined, it defaults to 10.
<code>convert(string sourceCharset, string targetCharset)</code>	Converts the string from “sourceCharset” to “targetCharset”
<code> tokenize([regex])</code>	Splits the string into tokens. Splits at all matches of “regex” if provided and at whitespaces otherwise.
<code>removeValues(blacklist)</code>	Removes specific values (i.e. stop words) from the value set. ‘blacklist’ is a comma-separated list of words.

Operators: comparators

- A comparison operator **evaluates two inputs** and **computes the similarity** based on a user-defined **distance measure** and a user-defined **threshold**.
- The **distance measure always outputs 0** for a **perfect match**, and a higher value for an imperfect match.
- Only **distance values between 0 and threshold** will result in a **positive similarity score**.
- Therefore it is **important to know** how the **distance measures** work and what the **range of their output values** is in order to **set a threshold value sensibly**.

Operators: comparators

- **Parameters:** Every time we use a comparator we need to set up some parameters

Parameter	Description
required (optional)	If required is true, the parent aggregation only yields a confidence value if the given inputs have values for both instances.
weight (optional)	Weight of this comparison. The weight is used by some aggregations such as the weighted average aggregation.
threshold	The maximum distance. For normalized distance measures, the threshold should be between 0.0 and 1.0.
Inputs	The 2 inputs for the comparison.

Operators: comparators

- **Character-based distance metrics:**
 - compare strings on the character level.
 - They are well suited for handling typographical errors

Measure	Description	Normalized
levenshteinDistance	Levenshtein distance. The minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, or substitution of a single character	No
levenshtein	The levenshtein distance normalized to the interval [0,1]	Yes
jaro	Jaro distance metric. Simple distance metric originally developed to compare person names.	Yes
jaroWinkler	Jaro-Winkler distance measure. The Jaro–Winkler distance metric is designed and best suited for short strings such as person names	Yes
equality	0 if strings are equal, 1 otherwise.	Yes
inequality	1 if strings are equal, 0 otherwise.	Yes

Operators: comparators

- **Token-based** distance metrics:
 - Suitable for other cases, for example:
 - Strings where parts are reordered e.g. “John Doe” and “Doe, John”
 - Texts consisting of multiple words

Measure	Description	Normalized
jaccard	Jaccard distance coefficient	Yes
dice	Dice distance coefficient	Yes
softjaccard	Soft jaccard similarity coefficient. Same as Jaccard distance but values within a Levenshtein distance of maxDistance are considered equivalent.	Yes

Operators: comparators

- **Special purpose** distance metrics:
 - to compare specific types of data e.g. numeric values.

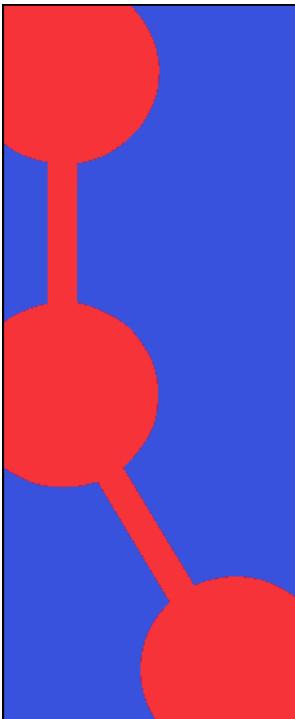
Measure	Description	Normalized
num(float minValue, float maxValue)	Computes the numeric difference between two numbers Parameters: minValue, maxValue The minimum and maximum values which occur in the datasource	No
date	Computes the distance between two dates	No
dateTime	Computes the distance between two date time values	No
wgs84(string unit, string curveStyle)	Computes the geographical distance between two points.	No

Operators: aggregators

- A comparison operator **evaluates two inputs** and **computes the similarity** based on a user-defined distance measure and a user-defined threshold.
- The **distance measure always outputs 0** for a perfect match, and a higher value for an imperfect match.
- Only **distance values between 0 and threshold** will result in a **positive similarity score**.
- Therefore it is **important to know** how the distance measures work and what the **range of their output** values is in order to **set a threshold value sensibly**.

Appendix: Installation guide

- It can be found at:
 - https://www.assembla.com/spaces/silk/wiki/Silk_Workbench



Publishing Linked Data

Boris Villazón-Terrazas. Oscar Corcho
Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo sn, 28660 Boadilla del Monte, Madrid
<http://www.oeg-upm.net>
ocorcho@fi.upm.es
Phone: 34.91.3366605, Fax: 34.91.3524819

Publication - Virtuoso Open-source edition

- <http://sourceforge.net/projects/virtuoso/files/virtuoso/6.1.4/virtuoso-opensource-win32-20111101.zip/download>
- Unzip to C:\software
- ODBC Registration:
 - Set up VIRTUOSO_HOME variable pointing out to virtuoso-opensource directory:
 - cd %VIRTUOSO_HOME%\lib
 - regsvr32 virtodbc.dll
- Creating a Windows Service
 - cd %VIRTUOSO_HOME%\database
 - SET PATH=%PATH%;%VIRTUOSO_HOME%\bin;%VIRTUOSO_HOME%\lib
 - virtuoso-t -? //to verify
 - virtuoso-t +service screate +instance "Instance Name" +configfile virtuoso.ini
 - virtuoso-t +service list //to verify
 - virtuoso-t -l "Instance Name" +service start //start the service

• <http://localhost:8890/conductor>

• Upload the generated files

- Ontology: <http://research.ull.es/graph/ontology> - research.owl
- Dataset: <http://research.ull.es/graph/dataset> - research.rdf
- Links: <http://research.ull.es/graph/links> - output.nt

- <http://localhost:8890/sparql>

Virtuoso endpoint

Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?Concept where {[] a ?Concept} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#))

Results Format:

Execution timeout: milliseconds (values less than 1000 are ignored)

Options: Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see [details](#))

65

- The Sindice RDF search engine

Enable effective discovery

Sindice - Data Web Services

Billion pieces of reusable information can already be found across hundreds of millions web pages which embed RDF and Microformats. Start consuming this data today with Sindice Data Web services.

[LEARN MORE →](#)

LATEST DATA

140316 (sigma) 136 triples <http://www.slideshare.net/andreulutzbecker>
140315 (sigma) 136 triples <http://www.slideshare.net/andreulutzbecker>
140314 (sigma) 136 triples <http://www.slideshare.net/andreulutzbecker>
140313 (sigma) 136 triples <http://www.slideshare.net/andreulutzbecker>

HIGHLIGHTS

Anything to Triples Any23

Introduction

Anything to Triples (any23) is a library and web service that converts data in RDF format from a variety of Web document formats.

SINDICE TWEET

Mon Jun 21 16:57:31 2010 Sindice at #semtech2010 , visit us at the DERIC Booth Wed and Thurs. Plus see our presentation Thurs <http://bit.ly/cKOVYx>

SINDICE BLOG

Sindice planned downtime this weekend JUN 09, 2010
HiDue to an expansion of one of our datacentres (and the electrical work that this implies), Sindice and related services such as sig.ma w...
[\(More →\)](#)

Any23 v0.4.0 Released MAY 27, 2010
Dear All, the Sindice FBK team is proud to announce the Any23 0.4.0 release. [\(More →\)](#)

Any23 v0.3.0 Released APR 23, 2010
Dear All, we're pleased to announce the Any23 0.3.0 release! Please keep in mind this is a beta, so expect some bugs in a few days. [\(More →\)](#)

66

Enable effective discovery

- Sitemap Protocol
 - Used by web crawlers
 - Efficiently find all your content & discover what has been updated



A sitemap file contains information regarding one or more URLs on your Web site. The information that is stored there helps search engines better spider your website.

67

Sitemap.xml example

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/
0.9">
  <url>
    <loc>http://yoursite/</loc>
  </url>
  <url>
    <loc>http://yoursite/products/53546</loc>
  </url>
  <url>
    <loc>http://yoursite/products/98421</loc>
  </url>
  <url>
    <loc>http://yoursite/products/41003</loc>
    <lastmod>2010-06-24</lastmod>
    <changefreq>daily</changefreq>
  </url>
</urlset>
```

Optional parts

68

Sitemap Protocol: Huge sitemaps

- Gzip-compress your sitemap
- Limit: 50k URLs or 10MB
 - split into multiple sitemap files
 - add a *sitemap index file*

69

Sitemap Protocol: Discovery

- Publish the sitemap file
- Add a line to `http://yoursite/robots.txt`
 - Web site owners use the `/robots.txt` file to give instructions about their site to web robots; this is called *The Robots Exclusion Protocol*.

Sitemap: `http://yoursite/sitemap.xml`

70

sitemap4rdf

- Simple command line tool
- Sends a SPARQL query to list all URIs
- Generates sitemap

```
sitemap4rdf http://yoursite/sparql http://yoursite/resource/
```

Example:

```
sitemap4rdf http://geo.linkeddata.es/sparql http://geo.linkeddata.es/resource/
```

- run sitemap4rdf specifying the **SPARQL endpoint** and the **prefix of the URLs** to include in the Sitemap

71

Submit the sitemap location - Sindice

- <http://sindice.com/main/submit>

Home | About | Search | **Submit** | Forum | Dev

Ping Form (one URL per line)

http://yoursite/
http://yoursite/products/12345

Submit Pages

Submit Sitemap form ([read me](#))

Email (optional, used for feedback):

Sitemap URL:

Submit Sitemap

[Home](#) | [About](#) | [Search](#) | [Submit](#) | [Forum](#) | [Dev](#) [Blog](#) | [Contact](#) | Copyright © 2008-2009 DERI

72

Submit the sitemap location - Google

- <https://www.google.com/webmasters/tools/>

Google webmaster tools

geo.linkeddata.es Back to Home

Dashboard Messages Site configuration Sitemaps Crawler access Sitelinks Change of address Settings Your site on the web Diagnostics Labs

Sitemaps

Submit a Sitemap to tell Google about pages on your site we might not otherwise discover.

Submitted URLs 249,318 116,514 URLs in web index

Submit a Sitemap

Sitemap	Status	Type	Downloaded	URLs submitted	URLs in web index
sitemap.xml.gz	✓	Sitemap	Sep 23, 2010	50,000	42,075
sitemap1.xml.gz	✓	Sitemap	Sep 19, 2010	50,000	7,123
sitemap2.xml.gz	✓	Sitemap	Sep 21, 2010	24,659	9,059
sitemap_index.xml	✓	Index	Sep 24, 2010	124,659	58,257

Show submissions: By me (4) - All (4)

Delete Resubmit

Help with: Creating and submitting Sitemaps Sitemap errors Sitemap details Sitemaps for multiple sites

73

