W3C

Use Cases and Requirements for Mapping Relational Databases to RDF

Editors' Draft 20 April 2010

This version:

$Id: Overview.xml,v 1.39 2010/05/11 17:21:15 eric Exp $

Latest version:

http://www.w3.org/2001/sw/rdb2rdf/use-cases/

Previous version:

http://www.w3.org/2001/sw/rdb2rdf/use-cases/

Editors:
Eric Prud'hommeaux, W3C <eric@w3.org> (Editor)
Michael Hausenblas, DERI, NUI Galway <michael.hausenblas@deri.org> (Editor)
Sören Auer, Universität Leipzig <auer@informatik.uni-leipzig.de> (Author)
Lee Feigenbaum, Cambridge Semantics <lee@thefigtrees.net> (Author)
Daniel Miranker, University of Texas at Austin <miranker@cs.utexas.edu> (Author)
Angela Fogarolli, University of Trento <afogarol@disi.unitn.it> (Author)
Juan Sequeda, University of Texas at Austin <jsequeda@cs.utexas.edu> (Author)

Abstract

The need to share data with collaborators motivates custodians and users of relational databases (RDB) to expose relational data on the Web of Data. This document examines a set of use cases from science and industry, taking relational data and exposing it in patterns conforming to shared RDF schemata. These use cases expose a set of functional requirements for exposing relational data as RDF in the RDB2RDF Mapping Language (R2RML).

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current

W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This is a Editors' Working Draft.

The documents produced by this Working Group are:

* RDB2RDF Mapping Language (R2RML) Specification
* RDB2RDF Mapping Language (R2RML) Conformance Tests

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

1 Introduction

The majority of dynamic Web content is backed by relational databases (RDB), and so are many Enterprise systems [DynaWebSites].

On the other hand, in order to expose structured data on the Web, the Resource Description Framework (RDF) [RDF] is used. This document reviews use cases and requirements for a relational to RDF mapping (RDB2RDF) with the following structure:

1. The remainder of this section motivates why mapping RDBs to RDF is necessary and needed and highlights the importance of a standard.
2. In the next section real-world RDB2RDF use cases are reviewed.
3. The last section discusses requirements regarding a RDB2RDF mapping language, driven by an analysis of the before-mentioned use cases.

1.1 Why Mapping RDBs to RDF?

The Web of Data [reference] is constantly growing due to its compelling potential of facilitating data integration and retrieval. At the same time, however, ~~closed~~ RDB systems host ~~the majority of enterprise information and contain a vast amount of semantic information~~ a vast amount of structured data in relational tables augmented with integrity constraints. In order to make ~~these~~ this huge amount of relational data available for the Web of Data, a connection must be established between RDBs and a format suitable for the Web of Data.

The advantages of creating an RDF view of relational data are inherited from the Web of Data and can be summarized based on the tasks they facilitate:

* Integration: data in different RDBs can be integrated using RDF semantics and mechanisms; in this sense, the Web of Data can be imagined as one big database. Moreover, information in the database can be integrated with information that comes from other data sources.
* Retrieval: once data are published in the Web of Data (as opposed to relational databases ~~RDBMSs~~), queries can span different data sources and more powerful retrieval methods can be built since they follow an Open World Assumption.

Data in the Web should be defined and linked in a way ~~which~~ that makes it accessible for humans and machines [LinkedData]. The Web of data is a scalable environment with explicit semantics where not only humans can navigate information, but also machines are able to find connections and use them to navigate through the information space. In order to realise this global information space, we need to:

* follow a common model to describe, connect and access resources;
* name resources in an unambiguous way

The most common way to publish resources in the Web of Data follows the RDF model [RDF] and uses ~~Unique~~ Uniform Resource Identifiers ([URI]) for resource identification, ~~in this way~~ thereby facilitating the creation of a comprehensive and flexible resource description.

In the following, we will use RDB2RDF to denote any technique that takes as an input a RDB (schema and data) and produces ~~a~~ one or more RDF graphs, as depicted in the following figure:

RDB2RDF principle

The consumer of the RDF Graph (virtual or materialized) essentially can access the RDF data in different ways:

1. Query access, which means the agent issues a SPARQL query against an endpoint exposed by the systems and processes the results (typically the result is a SPARQL result set in XML or JSON);
2. Entity-level access, which means the agent performs an HTTP GET on a URI exposed by the systems and processes the result (typically the result is an RDF graph);
3. Dump access, which means the agent performs an HTTP GET on dump of the entire RDF graph, for examples in Extract, transform, and load (ETL) processes.

1.2 Why a standard RDB2RDF method?

With the advent of the Web of Data, researcher and practitioners have compared the RDB model and the RDF model [RDB-RDF] and surveyed different approaches to map them [RDBMSMapping]. As noted in the survey on existing RDB2RDF mapping approaches [RDB2RDFSurvey], a couple of proposals how to tackle the RDB2RDF mapping issues are known.

As in the case of any standards, a fundamental motivation for a standard language for mapping relational data to RDF is to allow people to "talk" the same language. It enables creation of tools, expertise, and most importantly, reuse of specifications and collaboration among developers.

[comments: A much stronger and appealing motivation could be mirroring of schema and possibly some or all of data in databases from different vendors. I would state it as follows:]

Use of a standard for mapping language for RDB to RDF may even allow use of a single mapping specification in the context of "mirroring" of schema and (possibly some or all of the) data in various databases, possibly from different vendors (e.g., Oracle database, MySQL, etc.) and located at various sites. Similarly structured data (that is, data stored using same schema) is useful in many different organizations often located in different parts of the world. These organizations may employ databases from different vendors due to one or more of many possible factors (such as, licensing cost, resource constraints, availability of useful tools and applications and of appropriate database administrators, etc.). Presence of a standard RDB2RDF mapping language allows creation and use of a single mapping specification against each of the hosting databases to present a single (virtual or materialized) RDF view of the relational data hosted in those databases and this RDF view can then be queried by applications using SPARQL query or protocol.

Another major reason for a standard is to allow easy migration between different systems. Just as a single web-page in HTML can be viewed by
two different Web browsers from different vendors, a single RDB2RDF mapping standard should allow a user from one database to expose their data as RDF, and then, when they export their data to another database, allow the newly imported data to be queried as RDF without changing the mapping file.

For example, imagine that a database administrator is working on exposing weather data as Linked Data to be consumed by other applications. At first, this weather data is stored in a light-weight database (such as MySQL). However, as more and more weather data is collected, and more and more users access the Web data, the light-weight database may have difficulty scaling. Therefore, the database administrator migrates their database to a more heavy-weight database (such as Oracle Database 11g). Of course, the database administrator does not want to re-create the ability to view the data as RDF using a vendor-specific mapping file, but instead wants to seamlessly migrate the view of their data as RDF.

A standardized mapping between relational data and RDF allows the database administrator to migrate the view of their data as RDF across databases, allowing the vendors to compete on functionality and features rather than forcing database administrators to rewrite their entire relational data to RDF mapping when they want to migrate their data from one database to another.
Editorial note
another motivation for a standard is that for certain classes of systems (such as CMS) a 'default' mapping could be defined which can be deployed no matter what underlying RDB is used

1.3 Scope

Per the RDB2RDF charter, the scope of the mapping language we consider here is limited to read-only access to the RDB.
Editorial note
This section should be moved in a proper position; dunno how to make XMLSpec do it, hence I have it here for now

~~1.4 Glossary~~

~~identifier[Definition: TBD]~~

~~label[Definition: TBD]~~

2 Use Cases

~~When mapping RDB data to RDF data, one can consider different 'target source types':~~

~~* RDF that comes from a structured data source (such as other RDB, XLS, CSV, etc.);~~
~~* Native RDF on the Web (in RDF/XML, RDFa, etc.);~~
~~* RDF that comes from unstructured sources (HTML, PDF, etc)~~

~~With this in mind~~ With integration of relational data from one RDB with various kinds of data (such as, relational, XLS, CSV, unstructured text, etc.), the use cases presented in the following fall into one or more of the following categories:

1. I want to integrate my RDB with another structured source (RDB, XLS,CSV, etc), so I'll convert my RDB to RDF and assume my other structured source can also be in RDF. See UC1, UC3.
2. I want to integrate my RDB with existing RDF on the web (Linked Data), so I'll convert my RDB to RDF and then I'm able to link and integrate. See UC2, UC4.
3. I want to integrate my RDB with unstructured data (HTML, PDF, etc), so I'll convert my RDB to RDF and assume my other unstructured source can also be in RDF.
4. I want my RDB data to be available for SPARQL or other RDF-based querying, and/or for others to integrate with other data sources (structured, rdf, unstructured). See UC2, UC4.

2.1 UC1 - Patient Recruitment

The Semantic Web for Health Care and Life Sciences Interest Group has created several demonstrators using SPARQL to query clinical and biological relational databases. Included is the database structure, sample data, and a SPARQL query ~~with results, and an equivalent SQL query~~.
Following are six tables of sample diabetic patient data extracted from the University of Texas Health Science Center.
Some columns have been omitted from this use case for brevity.

Accompanying each table ~~is~~ are two RDF views (represented in Turtle) corresponding to RDF HL7/RIM and CDISK SDTM data structures
While there are many motivations for providing a common interface to administratively distinct databases (access to patient history, shared rules for clinical decision support, etc), in this case, SPARQL queries (following the table description) were used to find candidates for clinical studies.
For these RDF graphs, the following namespaces apply:

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>                    .
@prefix hl7: <http://www.hl7.org/v3ballot/xml/infrastructure/vocabulary/vocabulary#>  .
@prefix spl: <http://www.hl7.org/v3ballot/xml/infrastructure/vocabulary/vocabulary#>  .
```

2.1.1 Person

ID	MiddleName	SexDE	DateOfBirth	LastEditedDTTM
1234561		2	1983-01-02 00:00:00	2007-11-13 15:49:20
1234562		3	1963-12-27 00:00:00	2008-01-30 17:08:42
1234563		2	1983-02-25 00:00:00	2007-03-10 06:01:55

<http://hospital.example/DB/Person/ID.1234561#record> a hl7:Person ;
  hl7:entityName "" ; hl7:administrativeGenderCodePrintName Sex_DE:M ;
  hl7:livingSubjectBirthTime "1983-01-02T00:00:00Z"^^xsd:dateTime .
<http://hospital.example/DB/Person/ID.1234562#record> a hl7:Person ;
  hl7:entityName "" ; hl7:administrativeGenderCodePrintName Sex_DE:F ;
  hl7:livingSubjectBirthTime "1963-12-27T00:00:00Z"^^xsd:dateTime .

<http://hospital.example/DB/Person/ID.1234563#record> a hl7:Person ;
  hl7:entityName "" ; hl7:administrativeGenderCodePrintName Sex_DE:M ;
  hl7:livingSubjectBirthTime "1983-02-25T00:00:00Z"^^xsd:dateTime .

<http://hospital.example/DB/Person/ID.1234561#record> a sdtm:Patient ;
 stdm:middleName "" ;
 stdm:dateTimeOfBirth "1983-01-02T00:00:00Z"^^xsd:dateTime .
<http://hospital.example/DB/Person/ID.1234562#record> a sdtm:Patient ;
 sdtm:middleName "" ;
 sdtm:dateTimeOfBirth "1963-12-27T00:00:00Z"^^xsd:dateTime ;
 sdtm:sex <http://hospital.example/DB/Sex_DE#F> .

<http://hospital.example/DB/Person/ID.1234563#record> a sdtm:Patient ;
 sdtm:middleName "" ;
 sdtm:dateTimeOfBirth "1983-02-25T00:00:00Z"^^xsd:dateTime ;
 sdtm:sex <http://hospital.example/DB/Sex_DE#F> .

2.1.2 Sex_DE

ID	EntryCode	EntryName	EntryMnemonic
2	1	Male	M
3	2	Female	
	F		

<http://hospital.example/DB/Zex_DE/ID.2#record>
   hl7:administrativeGenderCodePrintName "Male"@en-us ;
   Sex_DE:EntryMnemonic "M"@en-us .
<http://hospital.example/DB/Zex_DE/ID.3#record>
   hl7:administrativeGenderCodePrintName "Female"@en-us ;
   Sex_DE:EntryMnemonic "F"@en-us .

<http://hospital.example/DB/Person/ID.1234561#record>

   stdm:sex <http://hospital.example/DB/Sex_DE#M> .
<http://hospital.example/DB/Person/ID.1234562#record>
   stdm:sex <http://hospital.example/DB/Sex_DE#F> .
<http://hospital.example/DB/Person/ID.1234563#record>
   stdm:sex <http://hospital.example/DB/Sex_DE#M> .

2.1.3 Item_Medication

ID	PatientID	ItemType	PerformedDTTM
99999999002	1234561	ME	2007-09-28 00:00:00
99999999003	1234562	ME	2007-09-28 00:00:00
99999999004	1234562	ME	2008-07-28 00:00:00

<http://hospital.example/DB/Person/ID.1234561#record>
hl7:substanceAdministration
<http://hospital.example/DB/Item_Medication/ID.99999999002#record> .
<http://hospital.example/DB/Item_Medication/ID.99999999002#record>
a hl7:SubstanceAdministration ;
hl7:effectiveTime _:t1 .
_:t1 hl7:start "2007-09-28T00:00:00"^^xsd:dateTime .
<http://hospital.example/DB/Person/ID.1234562#record>

hl7:substanceAdministration
<http://hospital.example/DB/Item_Medication/ID.99999999003#record> ;
hl7:substanceAdministration
<http://hospital.example/DB/Item_Medication/ID.99999999004#record> .
<http://hospital.example/DB/Item_Medication/ID.99999999003#record>
a hl7:SubstanceAdministration ;
hl7:effectiveTime _:t2 .
_:t2 hl7:start "2007-09-28T00:00:00"^^xsd:dateTime .
<http://hospital.example/DB/Item_Medication/ID.99999999004#record>
a hl7:SubstanceAdministration ;
hl7:effectiveTime _:t3 .
_:t3 hl7:start "2008-07-28T00:00:00"^^xsd:dateTime .

2.1.4 Medication

ID	ItemID	Dose	Refill	QuantityToDispense	DaysToTake	PrescribedByID	MedDictDE
88888888002	99999999002	2	6	180	45	1004682	132139
88888888003	99999999002	2	0	180	45	1004683	132139
88888888004	99999999003	2	6	180	45	1004682	132139
88888888005	99999999004	4	6	180	45	1004682	132139

<http://hospital.example/DB/Item_Medication/ID.99999999002#record>
   hl7:consumable <http://hospital.example/DB/Medication_DE/ID.132139#record> .
_:t1 hl7:durationInDays 45 .
<http://hospital.example/DB/Item_Medication/ID.99999999003#record>

   hl7:consumable <http://hospital.example/DB/Medication_DE/ID.132139#record> .
<http://hospital.example/DB/Item_Medication/ID.99999999004#record>
   hl7:consumable <http://hospital.example/DB/Medication_DE/ID.132139#record> .
_:t2 hl7:durationInDays 45 .
<http://hospital.example/DB/Item_Medication/ID.99999999005#record>
   hl7:consumable <http://hospital.example/DB/Medication_DE/ID.132139#record> .
_:t3 hl7:durationInDays 45 .

2.1.5 Medication_DE

ID	Entry	EntryCode	EntryName	NDC	Strength	Form
	UnitOfMeasure		DrugName	DisplayName		
132139	131933	98630	GlipiZIDE-Metformin HCl 2.5-250 MG Tablet			
	54868079500	2.5-250				
	TABS	MG	GlipiZIDE-Metformin HCl	GlipiZIDE-Metformin HCl 2.5-250 MG Tablet		

<http://hospital.example/DB/Medication_DE/ID.132139#record>
   hl7:displayName "GlipiZIDE-Metformin HCl 2.5-250 MG Tablet" .

2.1.6 NDCcodes

ingredient	RxCUI	labelType	name	NDC
6809	351273	Clinical	Glipizide 2.5 MG / Metformin 500 MG Oral Tablet	54868079500

<http://hospital.example/DB/Medication_DE/ID.132139#record>

   spl:activeIngredient _:i1 .
_:i1 spl:classCode 54868079500 .

[a sdtm:ConcomitantMedication ;
   sdtm:subject <http://hospital.example/DB/Person/ID.1234561#record> ;
   sdtm:standardizedMedicationName "GlipiZIDE-Metformin HCl 2.5-250 MG Tablet" ;
   hl7:activeIngredient [hl7:classCode 54868079500] ;
   sdtm:startDateTimeOfMedication "2007-09-28 00:00:00"^^xsd:dateTime] .

[a sdtm:ConcomitantMedication ;
   sdtm:subject <http://hospital.example/DB/Person/ID.1234562#record> ;
   sdtm:standardizedMedicationName "GlipiZIDE-Metformin HCl 2.5-250 MG Tablet" ;
   hl7:activeIngredient [hl7:classCode 54868079500] ;

```
    sdtm:startDateTimeOfMedication "2007-09-28 00:00:00"^^xsd:dateTime ] .

[ a sdtm:ConcomitantMedication ;
    sdtm:subject <http://hospital.example/DB/Person/ID.1234562#record> ;
    sdtm:standardizedMedicationName "GlipiZIDE-Metformin HCl 2.5-250 MG Tablet" ;
    hl7:activeIngredient [ hl7:classCode 54868079500 ] ;
    sdtm:startDateTimeOfMedication "2008-07-28 00:00:00"^^xsd:dateTime ] .
```

2.1.7 Queries Over the RDF Graph.

The RDF graphs place the relational data into the Semantic Web. There are many ways to consume RDF data, integration with other data sources, inference according to OWL or RIF rules, browsing with a linked data browser like Bubbles or Tabulator. This patient recruitment use case can be realised as a SPARQL query over the RDF graphs. Following is a query which extracts patients taking a particular class of medication (an anticoagulant, in this example) and not another (weight loss, here).

```
PREFIX sdtm: <http://www.sdtm.org/vocabulary#>
PREFIX spl: <http://www.hl7.org/v3ballot/xml/infrastructure/vocabulary/vocabulary#>

SELECT ?patient ?dob ?sex # ?takes ?indicDate ?indicEnd ?contra
 WHERE {
  ?patient a sdtm:Patient ;
        sdtm:middleName ?middleName ;
        sdtm:dateTimeOfBirth ?dob ;
        sdtm:sex ?sex .

  [         sdtm:subject ?patient ;
            sdtm:standardizedMedicationName ?takes ;

            spl:activeIngredient [ spl:classCode 6809 ] ;
        sdtm:startDateTimeOfMedication ?indicDate
  ] .
  OPTIONAL {
  [         sdtm:subject ?patient ;
            sdtm:standardizedMedicationName ?disqual ;
            spl:activeIngredient [ spl:classCode 11289 ]
        sdtm:startDateTimeOfMedication ?indicDate
  ] }
} LIMIT 30
```

2.2 UC2 - Web applications (Wordpress)

In order to make the Semantic Web useful to ordinary Web users, RDF and OWL have to be deployed on the Web on a much larger scale. Web applications such as Content Management Systems, online shops or community applications (e.g. Wikis, Blogs, Fora) already store their data in relational databases. Providing a standardized way to map the relational data structures behind these Web applications into RDF, RDF-Schema and OWL will facilitate broad penetration and enrich the Web with RDF data and ontologies and facilitate novel semantic browsing and search applications.

By supporting the long tail of Web applications and thus counteracting the centralization of the Web 2.0 applications the planned RDB2RDF standardization will help to give control over data back to end-users and thus promote a democratization of the Web.

To support this usecase scenario, the mapping language should be easily implementable for lightweight Web applications and have a shallow learning curve to foster early adoption by Web developers.

We illustrate this use case with the example of Wordpress. Wordpress is a popular blogging Web application and installed on tens of thousands of Web servers. Wordpress used a relational database (MySQL) with a relatively simple schema:
Wordpress SQL Schema

Wordpress SQL Schema

*

SQL Schema

*

SQL Schema and complete data of blog.aksw.org

* NTriples export of the above database created using Triplify

A mapping should be able to reuse existing vocabularies.

Mapped to RDF the resulting ontology should contain the classes post, attachment, tag, category, user and comment. An example instance of the post class, for example, should look like:

@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix dc11: <http://purl.org/dc/elements/1.1/> .
<http://blog.aksw.org/triplify/post/8> a sioc:Post .

<http://blog.aksw.org/triplify/post/8> sioc:has_creator
<http://blog.aksw.org/triplify/user/5> .
<http://blog.aksw.org/triplify/post/8> dc:created "2007-02-27T17:23:36"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://blog.aksw.org/triplify/post/8> dc11:title "Submissions open: 3rd Workshop on Scripting for the Semantic Web" .

<http://blog.aksw.org/triplify/post/8> sioc:content "The submissions web-site is now open for the 3rd Workshop on Scripting for the Semantic Web..." .
<http://blog.aksw.org/triplify/post/8> dc:modified "2008-02-22T21:41:00"^^<http://www.w3.org/2001/XMLSchema#dateTime> .

An example SPARQL query on the resulting ontology could be:

```
SELECT ?name, ?title
WHERE {
  ?post   rdf:type       sioc:Post .
  ?post   dc:title       ?title .
  ?post   dc:has_creator ?author
  ?author foaf:name      ?name
}
```

2.3 UC3 - Integrating Enterprise Relational databases for tax control

Goal: Integrating relational databases and exposing them on the web or intranet based on the final RDB2RDF XG 1.1.3 and 1.1.2 use cases) through the use of unique identifiers. This approach consist of integrating and interlinking data about entities on different databases.

Problem:

a. Join between entities described in different databases.

b. Join structured data (SQL) to structured data, from incompatible schema, or where data is dirty, poorly normalized, lacking proper keys/indices.

Requirements:

* mapping RDB2RDF (need to write User friendly tools that implement the mappings);
* federated SPARQL queries (SPARQL2SQL...);
* unique identifiers ;

Use Case Description:
This use case is a pilot project for the Trentino region tax agency. Trentino is an autonomous region in the north of Italy. The region has a population of 1 mil. People and more than 200 municipalities with their on information systems. The goal of the project is to integrate and link tax related data about people, organizations, buildings etc. This data come from different databases especially from the region's many municipalities, each with their own individual data structures, and other sources.
With our methodology we will provide a lightweight method for aggregating the data. In this way we are providing the user, a tax agent in our case an intelligent tool for navigating through the data present in the many different databases. The tool aggregates data and creates a profile for each tax payer. Each user profile shows different type of information , with links to other entities such as the buildings owned , payments made , location of residence etc.

Example

Supposed that we have two tables (Anagrafe and Urban_Cadastre) from different databases, we select some typical attributes for the two tables to explain our conversion method.
Table Angrafe includes the information about two type of entities, persons and locations (person's residence place), and some other information which are not properties of persons or locations:

Firstname	Lastname	City_Residence_Place	Country_Residence_Place	Other_Info
Paolo	Bouquet	Trento	Italy	xyz...

Table Urban_Cadastre contains the information about buildings and their owners:

Owner_Name	Building_LocalID	Building_Address	Building_Type
Paolo Bouquet	123456	VIA G.LEOPARDI	3

DDL

Image:ddl.JPG

Using traditional RDB2RDF translation methods the RDF representation for the two example tables coming from two different databases is shown below:

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:database_anagrafe="http://www.database1.org/anagrafe/">

 <rdf:Description rdf:about="http://www.database1.org/anagrafe/entry_row1">
   <database_anagrafe:Other_Info>xyz</database_anagrafe:Other_Info>

<database_anagrafe:Country_Residence_Place>Italy</database_anagrafe:Country_Resid
ence_Place>

<database_anagrafe:City_Residence_Place>Trento</database_anagrafe:City_Residence_
Place>

   <database_anagrafe:Last_Name>Bouquet</database_anagrafe:Last_Name>
   <database_anagrafe:First_Name>Paolo</database_anagrafe:First_Name>
 </rdf:Description> </rdf:RDF>

<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:database_urbano="http://www.database1.org/Urban_Cadastre/">

 <rdf:Description rdf:about="http://www.database1.org/Urban_Cadastre/entry_row1">
   <database_urbano:Building_Type>3</database_urbano:Building_Type>
   <database_urbano:Building_Address>VIA
G.LEOPARDI</database_urbano:Building_Address>
   <database_urbano:Building_LocalID>123456</database_urbano:Building_LocalID>

   <database_urbano:Owner_Name>Paolo Bouquet</database_urbano:Owner_Name>
 </rdf:Description> </rdf:RDF>

Querying

In SQL there is no way to create a query which joins data of two tables coming from different databases. *(I am not sure this is a true statement. In Oracle, for example, one can use DBLINK to join tables from different Oracle databases.)*
For solving the identity problem which is required by the use case creating a RDF representation is not enough.
The use case demands the use of unique identifier to refer to entities in order to join descriptions about the same entity coming from different
datasources.

2.4 UC4 - rCAD: RNA Comparative Analysis Database

rCAD - RNA Comparative Analysis using SQLServer: The tremendous increase in available biological information creates opportunities to decipher the structure, function and evolution of cellular components while presenting new computational challenges for performance and scalability. To fully utilize this large increase in knowledge, it must be organized for efficient retrieval and integrated for multi-dimensional analysis. Given this, biologists are able to invent new comparative sequence analysis protocols that will yield new and different structural and functional information. Based on Microsoft SQL-server, we have designed and implemented the RNA Comparative Analysis Database -rCAD which supports comparative analysis of RNA sequence and structure, and unites, for the first time in a single environment, multiple dimensions of information necessary for alignment viewing, sequence metadata, structural annotations, structure prediction studies, structural statistics of different motifs, and phylogenetic analysis. This system provides a queryable environment that hosts efficient updates and rich analytics.

The rCAD consists of different schema: Sequence Metadata, Evolutionary Relationships, Structural Relationships and Sequence Alignment.

For this use-case, we will be presenting only the Sequence Alignment schema. The SQL-DDL for Microsoft SQL Server can be found here: rCAD Sequence Alignment SQL DDL

This use-case presents the issue of a schema that does not have an existing domain ontology in which it can be mapped to. The closest domain ontology is the Multiple Alignment Ontology (MAO) which can only be mapped to the Sequence Alignment part of the entire rCAD database. However MAO is in the OBO language. Nevertheless, OBO ontologies can be translated to OWL (and back).

2.4.1 Expose Relational Data as RDF when there is no existing Domain Ontology to map the relational schema to

Rob, from the RNA lab would like to expose the Sequence Alignment data from the rCAD database as RDF. However, there is no existing domain ontology in which the relational schema can be mapped to. Therefore, an ontology should be derived automatically from the relational schema. The RDF data will become instance of this automatically generated ontology.

The Alignment table from the rCAD database is the following:

```
CREATE TABLE [AlignmentClassic].[Alignment]  (
   [AlnID]                int NOT NULL,
```

```
   [SeqTypeID]          tinyint NOT NULL,
   [AlignmentName]              varchar(max) NULL,
   [ParentAlnID]        int NULL,
   [NextColumnNumber]       int NOT NULL,
   CONSTRAINT [PK_Alignment] PRIMARY KEY([AlnID])
)
ON [PRIMARY]
GO
ALTER TABLE [AlignmentClassic].[Alignment]
   ADD CONSTRAINT [FK_Alignment_SequenceType]
         FOREIGN KEY([SeqTypeID])
         REFERENCES [dbo].[SequenceType]([SeqTypeID])
         ON DELETE NO ACTION
         ON UPDATE NO ACTION
GO
ALTER TABLE [AlignmentClassic].[Alignment]
   ADD CONSTRAINT [FK_Alignment_Alignment]
         FOREIGN KEY([ParentAlnID])
         REFERENCES [AlignmentClassic].[Alignment]([AlnID])
         ON DELETE NO ACTION
         ON UPDATE NO ACTION
GO
```

and the desired ontology that is generated automatically from the relational schema is the following:

```
PREFIX rcad: <http://rcad.org/vocabulary/rcad.owl#>
rcad:Alignment rdf:type owl:Class .
rcad:AlignmentName rdf:type owl:DatatypeProperty.
rcad:AlignmentName rdfs:domain rcad:Alignment.
rcad:AlignmentName rdfs:range xsd:string.
rcad:NextColumnNumber rdf:type owl:DatatypeProperty.
rcad:NextColumnNumber rdfs:domain rcad:Alignment.
rcad:NextColumnNumber rdfs:range xsd:integer.
rcad:ParentAlnID rdf:type owl:ObjectProperty
rcad:ParentAlnID rdfs:domain rcad:Alignment
rcad:ParentAlnID rdfs:range rcad:Alignment
```

and the desired RDF triples, which are instances of the automatically generated ontology are the following:

```
PREFIX rcad: <http://rcad.org/vocabulary/rcad.owl#>

PREFIX rcad-data: <http://rcad.org/vocabulary/rcad-data.rdf#>

rcad-data:alignment1000 a rcad:Alignment;
```

```
        rcad:AlignmentName "My Alignment 1000"^^xsd:string;
        rcad:NextColumnNumber "123"^^xsd:int;
        rcad:ParentAlnID rcad-data:alignment2000
```

2.4.2 Expose Mapping a Relational Database to an OWL DL ontology

Rob, from the RNA lab would like to expose the Sequence Alignment data from the rCAD database as RDF. Just recently the Multiple Alignment Ontology (MAO) has been released, which is an "ontology for data retrieval and exchange in the fields of multiple DNA/RNA alignment, protein sequence and protein structure alignment." However, this ontology has been developed in OBO. Nevertheless, OBO ontologies can be translated to OWL ontologies, specifically OWL DL. Therefore, Rob will like to map his rCAD database to the MAO ontology

The Alignment and Alignment Column tables from the rCAD database is the following:

```
CREATE TABLE [AlignmentClassic].[Alignment]  (
   [AlnID]              int NOT NULL,
   [SeqTypeID]          tinyint NOT NULL,
   [AlignmentName]              varchar(max) NULL,
   [ParentAlnID]        int NULL,
   [NextColumnNumber]       int NOT NULL,
   CONSTRAINT [PK_Alignment] PRIMARY KEY([AlnID])
)
GO
ALTER TABLE [AlignmentClassic].[Alignment]
   ADD CONSTRAINT [FK_Alignment_SequenceType]
         FOREIGN KEY([SeqTypeID])
         REFERENCES [dbo].[SequenceType]([SeqTypeID])
         ON DELETE NO ACTION
         ON UPDATE NO ACTION
GO
ALTER TABLE [AlignmentClassic].[Alignment]
   ADD CONSTRAINT [FK_Alignment_Alignment]
         FOREIGN KEY([ParentAlnID])
         REFERENCES [AlignmentClassic].[Alignment]([AlnID])
         ON DELETE NO ACTION
         ON UPDATE NO ACTION
GO

---

CREATE TABLE [AlignmentClassic].[AlignmentColumn]  (
   [AlnID]              int NOT NULL,
```

```
    [ColumnNumber]    int NOT NULL,
    [ColumnOrdinal]    int NOT NULL,
    CONSTRAINT [PK_AlignmentColumn] PRIMARY KEY([AlnID],[ColumnNumber])
)
GO
ALTER TABLE [AlignmentClassic].[AlignmentColumn]
    ADD CONSTRAINT [FK_AlignmentColumn_Alignment]
        FOREIGN KEY([AlnID])
        REFERENCES [AlignmentClassic].[Alignment]([AlnID])
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
GO
```

This is just part of the Multiple Alignment Ontology in OWL DL

(What should the reader get out of this ontology excerpt below? There is only one common element between the excerpt below and the schema above: the "alignment_column" table and correspondingly labeled OWL class. Specifically, the excerpt below just shows two classes: one related to the table "alignment_column" and a second class with a label "sub_alignment". Each individual in "sub_alignment" class must be "part_of" at least one individual from class ro:RO_000000 class, and each individual in "alignment_column" class must be "part_of" at least one individual from "sub_alignment" class. Not sure if this will be helpful to the reader.)

(Did some reordering to make it clear. Also, class ro:RO_0000000 definition is not present in this excerpt, which makes it unclear.)

```
PREFIX ro: <http://purl.org/obo/owl/RO#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX _global: <http://purl.org/obo/owl/_global#>

_global:part_of         rdf:type          owl:ObjectProperty
_global:part_of         rdf:type          owl:TransitiveProperty
_global:part_of         rdfs:label         "part_of"

genid:A129876           rdf:type          owl:Restriction
genid:A129876           owl:onProperty          _global:part_of
genid:A129876           owl:someValuesFrom ro:RO_0000000

ro:RO_0000031           rdf:type          owl:Class
ro:RO_0000031           rdfs:label         "sub_alignment"
ro:RO_0000031           rdfs:subClassOf         genid:A129876

genid:A129906           rdf:type          owl:Restriction
```

genid:A129906 owl:onProperty _global:part_of
genid:A129906 owl:someValuesFrom ro:RO_0000031

ro:RO_0000084 rdf:type owl:Class
ro:RO_0000084 rdfs:label "alignment_column"
ro:RO_0000084 rdfs:subClassOf genid:A129906

3 Approaches

The RDB2RDF working group would like to call out, without specifically endorsing, three emergent approaches to mapping relational data to RDF.

For all examples:

Please do **not** use of WG members' names in these examples. Replace with more generic names (as shown below or something similar):
Juan => PersonA
Lee => PersonB
Eric => PersonC
Michael => PersonD

Use different colors for the nodes: one for the classes and datatypes and one for the individuals.

3.1 Direct Mapping

Supplying a relational database (schema and data) plus a stem URI defines an RDF graph which emulates the relational structure.

3.2 Database to Ontology Mapping

The RDB2RDF may define a mapping semantics as a single step process from database to an RDF graph in a final ontology.

The diagram is currently **not** showing the subgraph corresponding to the instance triples.

We may want to point out that the exact subclasses of the uv:Teacher class depends upon the content of the "Classification" column of the "Teacher" table and hence may vary if DML operations modify the "Classification" column's content in the host database.

3.3 Direct Mapping Plus Ontology Mapping

Alternatively, the RDB2RDF may define a mapping semantics as a mapping to a direct graph, followed by the application of RDF graph transformations into an RDF graph in a final ontology.

For the intended triple <ex:Student1 foaf:name “Lee”>: the subject node in the diagram should change to ex:Student1 (the ‘:’ is missing), and predicate label should change to foaf:name (instead of ex:name).
Also, the predicate label rdf:type is not shown in the diagram for the (intended) triple <ex:Student1 rdf:type uv:Student>.

4 Requirements

Following is a set of proposed requirements for R2RML.
The RDB2RDF Working Group will prioritize the proposed requirements and accept those that enable useful mappings without excessively delaying development of the specification or implementations of the specification.

4.1 Core Requirements

The analysis of the above use cases yields a set of core requirements for the R2RML.

4.1.1 DIRECT - Direct Mapping

I think we are writing too much here and if time permits, we should try to simplify it. We can just show one cell of a table, along with pkey col for the row and name of the column, to explain its correspondence to an edge and corresponding RDF triple involving a OWL datatype property and a pair of cells – one from child table and one from parent table (assuming single-col fkey) – to explain its correspondence to an OWL object property. We can also mention that each table itself is mapped to a class.

Relational graph of clinical data

Relational data, with it's foreign and primary keys, is a potentially cyclic graph.
Edges in this relational graph, realised as join constraints in SQL queries, correspond to foreign/primary key declarations in the relational schema.
This graph may be visualized as arrows connecting tuples, following foreign keys in the relational schema.
This illustration shows the foreign keys between five tuples in the UC1 - Patient Recruitment database.

Direct graph of clinical data

Edges in the relational graph can be expressed as RDF triples, as can all of the attributes in a tuple.
This composes a direct graph, a graph representing exactly the information in the relation database.
This graph can be used when it is desired to let the database structure determine the effective ontology of the RDF view.

A minimal configuration SHOULD provide a (virtual) RDF graph representing the attributes and relationships between tuples in the relational database.

4.1.2 TRANSFORM - Transformative Mapping
Editorial note
There are many types of graph transformation, representable by SQL views, SPARQL CONSTRUCTs, Horn logic, etc. The RDB2RDF working group would like feedback to decide if, for instance, the transformations expressed by RIF Basic Logic Dialect are appropriate. Use case contributions to this document will help determine the exact expressivity required.

4.1.2.1 SHAPE - Non-isomorphic Graph Shape Transformations
HL7-like RDF graph of clinical data

It is good Semantic Web practice to use shared ontologies.
Using popular ontologies, or even ontologies which represent information in multiple domain databases, usually requires transformations [GraphTransform] into a graph which is not isomorphic to the direct graph.

Continuing the example above, this HL7-like representation of a prescription has a subject (a patient), a name, and an ingredient.
Coercing the relational graph into this pattern requires graph transformation.

SDTM-like RDF graph of clinical data

In this graph, a patient has a substance administration, which has a consumable, which in turn, has a name and an ingredient.

The R2RML language MUST express transformations of the relational graph to produce the (virtual) RDF graph.

4.1.2.2 LABELGEN - Label Generation (suggestion: URIGEN or IDGEN - generating URIs for identifying the resources?)

RDF identifiers for objects in the conceptual model can, in some cases, be generated from a transformation of the schema and data in a tuple representing that conceptual model.

For example, it may be sufficient to identify a patient in a clinical database with primary key patientID and value 1234561 as http://myclinic.example/patient/patientID.12334561#x, while but if the patient IDs are shared with another database, it will be necessary to transform one or both of these identifiers into a common form, e.g. http://allclinics.example/sharedRecords/patient.12334561#y.

These use cases are labeled custom-identifier.

Given a row in a protein database with a primary key attribute "ID" and another unique attribute "uniProt":

ID	uniProtname	seqLength
18	68250 YYHAB	246 AA

The RDB2RDF would like to ask the world which of the following subject mappings are likely to meet the most use cases:

<http://mydb.example/prots/ID=18> db:name "YYHAB" . # consistent function of the primary key
<http://mydb.example/prots18/more/path> db:name "YYHAB" . # user-defined function of the primary key
<http://www.uniprot.org/uniprot/P68250> db:name "YYHAB" . # user-defined function of arbitrary attributes

The former uses a potentially hard-coded formula, the middle uses a user-supplied function of the primary key and the latter uses a function of a different attribute to produce a common proteomic node label.

The expression of the SQL String "YYHAB" is, in the above examples, expressed directly as an RDF Plain Literal (perhaps the SQL draft suggests "YYHAB"^^xsd:string, I don't recall). Expressing the seqLength would be an opportunity for micro-parsing as it encodes both the length (and integer) and the metric.

4.1.2.3 DATATYPES - Datatypes

Relational data types should be treated consistently with RDF datatypes per SQL-XSD mapping ISO IWD 9075-14:2011(E) Subclause 9.5, "Mapping SQL data types to XML Schema data types", possibly including XML datatypes defined at http://standards.iso.org/iso/9075/2003/sqlxml.
Editorial note
Need to find minutes/Wiki page - Juan?

4.1.3 SQLGEN - Query Translation

One use of R2RML is to materialize RDF views of data.
Another is to define virtual RDF views, enabled by translating SPARQL queries over the RDF view into SQL queries over the original database.

The R2RML language MUST allow mapping specification to have sufficient information to enable transformation of SPARQL queries over the RDF view into efficient SQL queries over the relational database.

4.1.4 CONNECTION - Database Connection

The mapping language SHOULD allow specification of a database connection, which e.g. can be established using a DSN or connection string.

4.1.5 MICROPARSING - User defined output processing functions

The mapping language SHOULD enable the use of user defined output processing functions in order to transform values from the physical database representation into some domain representation. For example, a database row might contain Wiki text, which should be transformed into HTML - this can be achieved, for example, by using standard or user defined SQL functions or using XQuery/XPath Functions and Operators.
Editorial note
Microparsing can come in many flavors. The RDB2RDF would like feedback as to which, if any, functions are needed in version 1 and what expressivity should be available to user-defined functions withing R2RML.

4.1.6 TABLEPARSING - Attribute-Value Tables

In some cases relational database schema already contain attribute value tables. The mapping language SHOULD also allow mapping of such attribute-value tables to RDF by designating such tables accordingly (e.g., saying “UNPIVOTED”).
Editorial note
Example? Do you think of something like a cell of type STRING containing key1=val1, key2=val which you want to handle? Again microparsing?

4.1.7 NAMEDGRAPH - Named Graphs

The mapping language SHOULD enable the creation of multiple Named Graphs within one mapping definition.
Editorial note
This seems not widely deployed. How exactly should this work? (We have a detailed example we have in: http://www.w3.org/2001/sw/rdb2rdf/wiki/Example_of_SQL-Query_based_Approach_%28Part_2:_Data%29:_RDB_%28relational%29_Data_and_corresponding_%28virtual%29_RDF_Graphs)

4.1.8 MANYTOMANY - Exposing many-to-many join tables as simple triples

Please replace the names below with PersonA, PersonB, PersonC, etc.

TeacherID	TeacherName
1	Michael
2	Ahmed
3	Marcelo
...	...

StudentId	TeacherID
1	1
1	2
2	3
3	2
...	...

StudentId	StudentName
1	Lee
2	Eric
3	Juan
...	...

The community's school district maintains an RDB with basic student and personnel information, including a STUDENTS and a TEACHERS table. The relationship between the two is given in a STUDENT_TEACHER table:

The SQL DDL for these tables are the following:

```
CREATE TABLE student (
                                  studentID int PRIMARY KEY,
                                  studentName varchar
                                 )

                                 CREATE TABLE teacher (
                                  teacherID int PRIMARY KEY,
                                  teacherName varchar
                                 )

                                 CREATE TABLE student_teacher (
                                  studentID int,
                                  teacherID int,
                                  PRIMARY KEY(studentID, teacherID),
                                  FOREIGN KEY(studentID) REFERENCES
student(studentID),
                                  FOREIGN KEY(teacherID) REFERENCES
teacher(teacherID)
                                 )
```

SemantEducaTrix, the most recent Semantic Web company to burst into the educational software market, is mapping the school system's relational database to RDF / SPARQL. They'd like to access the relationships modeled with this join table as simple links between students and teachers:

```
ex:student1 ex:studentName "Lee".
                                  ex:student2 ex:studentName "Eric".
                                  ex:student2 ex:studentName "Juan".


ex:teacher1 ex:teacherName "Michael".
                                  ex:teacher2 ex:teacherName "Ahmed".
                                  ex:teacher3 ex:teacherName "Marcelo".


ex:student1 ex:has_teacher ex:teacher1, ex:teacher 2 ;
                                  ex:student2 ex:has_teacher ex:teacher3 ;
                                  ex:student3 ex:has_teacher ex:teacher2 ;
                                  ...
```

4.1.9 VALUETYPE - Value-based type specification

The TEACHERS table, see above, has a Classification column:

TeacherId	Classification ...
1	History
2	Physics
3	Music

SemantEducaTrix wants to instantiate these teachers as different {{{rdf:type}}}s depending on the value in the Classification column:

```
ex:teacher1 a ex:HistoryTeacher .
                                    ex:teacher2 a ex:PhysicsTeacher .
                                    ex:teacher3 a ex:MusicTeacher .
```

4.2 Non-Core Requirements

4.2.1 NSDECL - Namespace declaration

The mapping language SHOULD enable the declaration and use of namespace prefixes.
Editorial note
What exactly does this mean? Considered syntactic sugar for now; what about XML-NS?

4.2.2 METADATA - Static Metadata

The mapping language SHOULD enable the attachment of static metadata (such as licensing information) to all RDF entities or instances of a certain class. This is in particular important, when RDF is published as Linked Data on the Web.
Editorial note
See also next requirement which could be considered a special case of static metadata.

4.2.3 PROVENANCE - Provenance

The mapping language SHOULD support the preservation of provenance by generating additional RDF triples according to a provenance vocabulary, such as http://sourceforge.net/apps/mediawiki/trdf/index.php?title=Guide_to_the_Provenance_Vocabulary

4.2.4 UPDATES - Update Logs

The mapping language SHOULD provide extension points to support the creation of update logs of relational data.
Editorial note
Not really in the scope of R2RML, hence a nice-to-have feature.
Then why are we including it here? I would suggest removing Sec 4.2.4 altogether to avoid unnecessary confusion.

4.3 Application Requirements
Editorial note
See http://www.w3.org/2001/sw/rdb2rdf/wiki/Use_Cases_and_Requirements#R2ML_Application_Use_Case_Requirements but unsure how this fits in - needs heavy reformulation
Again, we should not include it in this version.

5 Acknowledgments

The editors gratefully acknowledge contributions from the members of the W3C RDB2RDF Working Group: Marcelo Arenas, Sören Auer, Samir Batla, Richard Cyganiak, Daniel Daniel Miranker, Souripriya Das, Alexander de Leon, Alexander de Leon, Orri Erling, Ahmed Ezzat, Lee Feigenbaum, Angela Fogarolli, Enrico Franconi, Howard Greenblatt, Wolfgang Halb, Harry Halpin, Michael Hausenblas, Nuno Lopes, Li Ma, Ashok Malhotra, Ivan Mikhailov, Eric Prud'hommeaux, Juan Sequeda, Seema Sundara, Ben Szekely, Edward Thomas, Boris Villazón-Terrazas.

6 References

DynaWebSites
A survey on dynamic Web content generation and delivery techniques, Jayashree Ravi, Zhifeng Yu, Weisong Shi, 2009.
Editorial note
we need a better reference for 'majority of dynamic Web content is RDB-backed'
(See http://www.cs.wayne.edu/~weisong/papers/ravi09-dynamic-content.pdf.)
LinkedData
Linked Data, Tim Berners-Lee, Design Issue Note, 2006. (See http://www.w3.org/DesignIssues/LinkedData.html.)
SPARQL
SPARQL Query Language for RDF, Eric Prud'hommeaux and Andy Seaborne 2008. (See http://www.w3.org/TR/rdf-sparql-query/.)
SQL
SQL.
Editorial note
need a better reference here
(See http://en.wikipedia.org/wiki/SQL.)

RDF-RDB
RDF Access to Relational Databases (See http://www.w3.org/2003/01/21-RDF-RDB-access/.)
RDB-RDF
Relational Databases on the Semantic Web, Tim Berners-Lee, Design Issue Note, 1998. (See http://www.w3.org/DesignIssues/RDB-RDF.html.)
RDBMSMapping
SWAD-Europe Deliverable 10.2: Mapping Semantic Web Data with RDBMSes, Dave Beckett and Jan Grant, 2003. (See http://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report/#sec-mapping.)
RDB2RDFSurvey
A Survey of Current Approaches for Mapping of Relational Databases to RDF, Satya S. Sahoo, Wolfgang Halb, Sebastian Hellmann, Kingsley Idehen, Ted Thibodeau Jr, Sören Auer, Juan Sequeda, Ahmed Ezzat, 2009. (See http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf.)
GraphTransform
Wikipedia Article on Graph rewriting (See http://en.wikipedia.org/wiki/Graph_rewriting.)
RDF
Resource Description Framework (RDF): Concepts and Abstract Syntax, G. Klyne, J. J. Carroll, Editors, W3C Recommendation, 10 February 2004 (See http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.)
ReuseableIDs
Reusable Identifiers in the RDB2RDF mapping language, Michael Hausenblas and Themis Palpanas, 2009. (See http://esw.w3.org/topic/Rdb2RdfXG/ReusableIdentifier.)
URI
RFC3986 - Uniform Resource Identifier (URI): Generic Syntax (See http://tools.ietf.org/html/rfc3986.)

A CVS History

$Log: Overview.xml,v $
Revision 1.39 2010/05/11 17:21:15 eric
+ ack section

Revision 1.38 2010/05/11 17:16:23 eric
~ tweak termdefs

Revision 1.37 2010/05/11 17:09:46 eric
- SQL query

Revision 1.36 2010/05/11 16:59:24 eric
+ glossary

Revision 1.35 2010/05/11 16:57:14 eric
~ update per WG decision 2010-05-11T16:55:22Z <cygri> RESOLUTION: name 4.1.1 Direct Mapping, 4.1.2 Transformative Mapping

Revision 1.34 2010/05/11 16:40:36 eric
~ cleaned up UC3 a bit

Revision 1.33 2010/05/11 03:01:08 eric
~ trying out Juan Sequeda's illustrations

Revision 1.32 2010/05/11 02:59:38 eric
~ moved UC5 an UC6 to requirements per Sören Auer's request

Revision 1.31 2010/05/11 02:42:12 eric
~ minor wording

Revision 1.30 2010/05/10 00:14:47 eric
~ reworded DIRECT a bit

Revision 1.29 2010/05/09 23:23:33 eric
~ improve image captions

Revision 1.28 2010/05/09 18:04:05 eric
~ per http://www.w3.org/mid/20100509175536.GA14557@w3.org

Revision 1.27 2010/05/04 17:31:52 eric
~ followed MacTed's categorization advice

Revision 1.26 2010/05/04 17:19:20 eric
~ s/functional requirements/core requirements/

Revision 1.25 2010/05/04 17:16:03 eric
~ don't presume a mapping from the direct graph

Revision 1.24 2010/04/27 15:11:11 mhausenb
added categorisation to UC section and cross-linked UC; cleaned up UC5 and UC6 to show up in the correct position (where in div3 earlier)

Revision 1.23 2010/04/23 16:52:27 eric
+ proposed requirements text

Revision 1.22 2010/04/23 16:45:48 eric
+ SQLGEN requirement

Revision 1.21 2010/04/23 16:24:31 eric

~ new names and titles for requirements

Revision 1.20 2010/04/23 12:19:36 eric
~ plural error

Revision 1.19 2010/04/23 11:34:51 eric
+ discussion of shared ontologies

Revision 1.18 2010/04/22 14:38:08 eric
+ distinction between direct and non-isomorphic graphs

Revision 1.17 2010/04/22 13:52:27 eric
~ narrowed wordpress turtle

Revision 1.16 2010/04/21 15:08:17 mhausenb
integrated http://www.w3.org/2001/sw/rdb2rdf/wiki/Requirements/Functional

Revision 1.15 2010/04/21 11:45:27 eric
~ fixed bibl entry markup

Revision 1.14 2010/04/21 11:04:12 eric
+ http://www.w3.org/2003/01/21-RDF-RDB-access/

Revision 1.13 2010/04/21 04:46:17 eric
- some headings at the top of Patient Recruitment
+ use case plea for different node mappings
+ bit on datatypes

Revision 1.12 2010/04/20 15:36:38 mhausenb
minor fixes in CSS and doc

Revision 1.11 2010/04/20 14:26:34 eric
~ overhauled tables

Revision 1.10 2010/04/20 14:02:58 eric
~ fixed s and
</br>s

Revision 1.9 2010/04/20 13:56:50 mhausenb
XMLSpec conformancy fixing

Revision 1.8 2010/04/20 13:44:03 mhausenb
updated RDB2RDF principle and added scope (read-only)

Revision 1.7 2010/04/20 11:59:28 mhausenb
added RDB2RDF principle, restructured doc to 1. intro, 2. UC, 3. Reqs, added authors, added WP schema figure from Wiki

Revision 1.6 2010/04/20 04:05:43 eric
~ a bit more integration

Revision 1.5 2010/04/19 13:53:08 mhausenb
new version of XMLSpec XSLT