

<http://latc-project.eu>



## **D3.1.2 First Release of Documentation and Tutorials**

<b>Project GA No.</b>	FP7-256975
<b>Project acronym</b>	LATC
<b>Start date of project</b>	2010-09-01
<b>Document due date</b>	2011-02-28
<b>Actual date of delivery</b>	2011-02-28
<b>Lead Partner</b>	FUB
<b>Reply to</b>	Anja Jentsch, <a href="mailto:mail@anjajentsch.de">mail@anjajentsch.de</a>
<b>Document status</b>	DRAFT



<b>Project GA No.</b>	FP7-256975
<b>Project acronym</b>	LATC
<b>Project full title</b>	Linking Open Data Around The Clock
<b>Dissemination level</b>	PU
<b>Number of pages</b>	21
<b>Task responsible</b>	FUB
<b>Other contributors</b>	Christophe Guéret
<b>Author(s)</b>	Anja Jentzsch
<b>EC Project Officer</b>	Stefano Bertolo
<b>Keywords</b>	tools, recommendation, publication, consumption

# Table of Contents

<b>1</b>	<b>EXECUTIVE SUMMARY</b>	<b>4</b>
<b>2</b>	<b>TOOL DOCUMENTATION</b>	<b>4</b>
2.1	Neologism	4
2.2	D2R Server	5
2.3	RDF Extension for Google Refine	6
2.4	Pubby	6
2.5	Any23	7
2.6	LDSpider	7
2.7	Sindice	8
2.8	R2R Framework	9
2.9	Silk Link Discovery Framework	9
2.10	Sig.ma	10
2.11	RelFinder	10
2.12	OntoWiki	10
2.13	SPARQL Views	11
<b>3</b>	<b>EXAMPLE</b>	<b>12</b>
3.1	Prerequisites	13
3.2	Neologism: Modeling the CORDIS ontology	13
3.3	D2R Server: Publishing the CORDIS data set as Linked Data	17
3.4	CKAN: Find thematically overlapping Linked Data sets	18
3.5	Silk Link Discovery Framework: Interlinking the CORDIS Linked Data set	18
3.6	RelFinder: Explore the CORDIS Linked Data set	21

## 1 Executive Summary

This deliverable documents the usage of the tools that are recommended in the LATC Data Publication & Consumption Tools Library<sup>1</sup>. It furthermore describes an example usage of the library.

The process of publishing the Community Research and Development Information Service (CORDIS)<sup>2</sup> as Linked Data as well as consuming it, will be documented with screencasts in future versions of D3.1.2. The screencasts will cover the following tools:

- Neologism
- D2R Server
- Silk Link Discovery Framework
- OntoWiki
- SPARQL Views

The screencasts will be made available online and accessible via the Tools Library website<sup>3</sup>.

## 2 Tool Documentation

### 2.1 Neologism

Neologism is a web application that is based on the content manager Drupal<sup>4</sup>. As such, it requires to have a working web server with some minimum requirements<sup>5</sup>:

- A Web server that can run PHP 5.2 or higher scripts and supports Drupal Clean URLs (the usage of mod\_rewrite for Apache is recommended)
- A MySQL server version 5.0.15 or higher
- The PDO database extension for PHP

All-in-one packages such as WAMP<sup>6</sup> and MAMP<sup>7</sup> can be used to install all of the above on either a Windows or MacOS system.

Installation procedure:

- Uncompress the downloaded<sup>8</sup> release of neologism into a directory accessible from the Web server. For instance, if the Web server is configured to access content in the “srv” directory, unpack the content of the archive for Neologism into “srv/neologism”.
- Create an empty database and name it with a properly name. For instance, “neologism-db”.
- Open the address <http://localhost/neologism> in a web browser to start the installation procedure. In that address, replace “neologism” by the name of

---

<sup>1</sup> <http://www4.wiwiss.fu-berlin.de/latc/toollibrary/>

<sup>2</sup> <http://cordis.europa.eu/>

<sup>3</sup> <http://www4.wiwiss.fu-berlin.de/latc/toollibrary/>

<sup>4</sup> <http://www.drupal.org>

<sup>5</sup> <http://neologism.deri.ie/requirements>

<sup>6</sup> <http://www.wampserver.com/en/>

<sup>7</sup> <http://www.mamp.info/en/index.html>

<sup>8</sup> <http://neologism.googlecode.com/files/neologism-0.4.10.zip>

the directory you unpacked neologism into. The installer is a friendly procedure asking several questions about the configuration to generate. Screenshots of the different steps are available on the Drupal web site<sup>9</sup>.

## 2.2 D2R Server

D2R Server is a tool for publishing the content of relational databases on the Web as Linked Data. D2R Server uses a customizable D2RQ mapping to map database content into. The D2RQ mapping language<sup>10</sup> is a declarative language for describing the relation between a relational database schemata and RDFS vocabularies or OWL ontologies. A D2RQ map is an RDF document.

In order to run D2R Server you will need:

- Java 1.4 or newer on the path.
- A supported database. D2R Server works with Oracle, MySQL, PostgreSQL, Microsoft SQL Server, and any SQL-92 compatible database. Microsoft Access can be used with some restrictions. More information is available in the D2RQ manual<sup>11</sup>.
- A modern browser like Firefox, Opera or Safari for using D2R Server's AJAX SPARQL Explorer. Internet Explorer can only browse the HTML pages, but not use the SPARQL Explorer.
- Optionally, a J2EE servlet container as a deployment target. D2R Server can be run either as a stand-alone web server or inside an existing servlet container.

What to do:

1. Download<sup>12</sup> and extract the latest D2R Server version into a suitable location.
2. Download a JDBC driver from your database vendor<sup>13</sup>. Place the driver's JAR file into D2R Server's /lib directory. Also take note of the driver class name (e.g. org.postgresql.Driver for PostgreSQL or oracle.jdbc.driver.OracleDriver for Oracle) and JDBC URL pattern (e.g. jdbc:mysql://servername/database for MySQL) from the driver's documentation. Drivers for MySQL and PostgreSQL are already included with D2R Server.
3. Generate a mapping file for your database schema. Mappings are to be defined by D2RQ Mapping Language. Change into the D2R Server directory and run:

```
generate-mapping -o mapping.n3 -d driver.class.name
-u db-user -p db-password jdbc:url:...
```

<sup>9</sup> <http://drupal.org/documentation/install/run-script>

<sup>10</sup> <http://www4.wiwiw.fu-berlin.de/bizer/d2rq/spec/#specification>

<sup>11</sup> <http://www4.wiwiw.fu-berlin.de/bizer/d2rq/spec/#compatibility>

<sup>12</sup> <http://www4.wiwiw.fu-berlin.de/bizer/d2r-server/#download>

<sup>13</sup>

[http://www.java2s.com/Tutorial/Java/0340\\_Database/AListofJDBCDriversconnectionstringdrivername.htm](http://www.java2s.com/Tutorial/Java/0340_Database/AListofJDBCDriversconnectionstringdrivername.htm)

mapping.n3 is the name for the new mapping file. -d can be skipped for MySQL.

4. Start the server:

```
d2r-server mapping.n3
```

5. Test the Server: Open `http://localhost:2020/` in a web browser. Replace 2020 by the port you specified in your mapping file.

You can browse the database content or use the SPARQL Explorer to execute queries and display results in a number of formats.

## 2.3 RDF Extension for Google Refine

The RDF Extension for Google Refine<sup>14</sup> adds a graphical user interface for exporting data of Google Refine<sup>15</sup> projects in RDF format. The export is based on mapping the data to a template graph using the GUI.

1. You need to have Google Refine<sup>16</sup> (version 2.0 or later) installed on your machine.
2. Download the latest RDF Extension for Google Refine.
3. If it does not exist, create a folder named extensions under your user workspace directory for Google Refine. The latter should be findable in the following places depending on your operating system:

- Linux

```
~/.local/share/google/refine/extensions
```

- Windows

```
C:/Documents and Settings/<user>/Application  
Data/Google/Refine/extensions
```

```
C:/Documents and Settings/<user>/Local Settings/Application  
Data/Google/Refine/extensions
```

- Mac OSX

```
~/Library/Application Support/Google/Refine/extensions/
```

4. Unzip the downloaded archive into the extensions folder created.
5. It is recommended that you have internet connection during the first running of the extension, as it will try to download a set of predefined vocabularies (rdf, rdfs and owl). You still can download them later anyway.
6. Restart Google Refine.
7. Open an existing project or create a new one. You should see a small RDF button in the upper right corner.

## 2.4 Pubby

Pubby<sup>17</sup> makes it easy to turn a SPARQL endpoint into a Linked Data server by offering a Linked Data and HTML interface. Pubby provides dereferenceable URIs by rewriting URIs found in the SPARQL-exposed dataset into the Pubby server's

<sup>14</sup> <http://lab.linkeddata.deri.ie/2010/grefine-rdf-extension/>

<sup>15</sup> <http://code.google.com/p/google-refine/>

<sup>16</sup> <http://code.google.com/p/google-refine/wiki/Downloads?tm=2>

<sup>17</sup> <http://www4.wiwiss.fu-berlin.de/pubby/>

namespace. Furthermore it takes care of handling 303 redirects and content negotiation.

In order to set up Pubby, you need to:

1. Download the latest version of Pubby<sup>18</sup>.
2. If you haven't already, download and install a servlet container. Pubby has been tested with Tomcat and Jetty.
3. Unzip the Pubby distribution and copy the webapp directory into the servlet container's webapps folder. If Pubby is the only web application you want to run in the container, then rename the webapp directory to root. Otherwise, rename it to something like mydataset. Assuming your server is set up to run at `http://myserver/`, this will change the Pubby root to `http://myserver/mydataset/`.
4. Modify the configuration file<sup>19</sup> to suit your needs. It is located within Pubby's webapp directory, at `/WEB-INF/config.ttl`.

## 2.5 Any23

Any23 is a command line tool made using the programming language Java. In order to use it, you will need to have a recent version of the Java environment installed on your machine. Java is available for download on the web site from Oracle<sup>20</sup>. You will also need maven to build the software<sup>21</sup>.

Installation procedure:

- Download Any23 from its homepage<sup>22</sup> and unpack it in a directory of your choice
- Any23 gets run by typing the command `./any23-core/bin/any23` from the directory in which you unpacked it. You need to execute that command once to have the software compiled.

Usage:

- To get the RDF content in a web page, type `./any23-core/bin/any23` followed by the address of the page to fetch the content from. For instance:

```
./any23-core/bin/any23 http://dbpedia.org/resource/Amsterdam
./any23-core/bin/any23 http://librdf.org
```

## 2.6 LDSpider

LDSpider<sup>23</sup> is a web crawling framework for the Web of Data. It can be used through a command line application as well as through a flexible API for a usage within another application.

Requirements:

- Java Runtime Environment

Usage:

---

<sup>18</sup> <http://www4.wiwiss.fu-berlin.de/pubby/download>

<sup>19</sup> <http://www4.wiwiss.fu-berlin.de/pubby/#configuration>

<sup>20</sup> <http://www.java.com/inc/BrowserRedirect1.jsp?locale=en&host=www.java.com>

<sup>21</sup> <http://maven.apache.org/download.html>

<sup>22</sup> <http://any23.googlecode.com/files/any23-0.4.1.zip>

<sup>23</sup> <http://code.google.com/p/ldspider/>

1. Download the latest LDSpider version.
2. Run:

```
$ java -jar ldspider-1.1dev.jar
usage: [-a <file>] [-b <depth uri-limit> | -c <max-uris>] [-h] [-n]
      [-o <file>] [-r <redirects>] [-s <file> | -u <uri>]
      [-t <threads>] [-y]

-a <file>                name of access log file
-b <depth uri-limit>    do strict breadth-first
-c <max-uris>           use load balanced crawling strategy
-h,--help              print help
-n                    do not extract links, just follow redirects
-o <file>              name of NQuad file with output
-oe <uri>              URI of endpoint that supports SPARQL/Update
-r <redirects>        write redirects.nx file
-s <file>              location of seed list
-t <threads>          number of threads (default 2)
-u <uri>              uri of an instance
-y                    stay on domains of seed uris
-follow <uri>        only follow a specific predicate
                     e.g. http://www.w3.org/2002/07/owl#sameAs
```

## 2.7 Sindice

Sindice is a web application so there is nothing to install. The only requirement is to have a modern Web browser to access to the main site of Sindice<sup>24</sup>. They are essentially two ways to use Sindice: either as a search engine to find data on the Web of Data or as an inspector tool to see how some data published is seen on the Web of Data. The first usage is similar to that of Google, the second usage is an alternative to using tools like any23.

Search engine:

- Click on the “search” tab on the main page of Sindice and enter some keyword corresponding to the searched information in the dedicated entry box, then press enter. The result is a list of resources matching the search criteria, and the possibility to filter them.

Data inspector:

- Click on the “inspector tool” tab on the main page of Sindice and enter the address of a page on the Web. The document can be a resource from the Web of Data (such as <http://dbpedia.org/resource/Amsterdam>) or a Web document (such as <http://librdf.org>). The result is the list of triples found at that address.

---

<sup>24</sup><http://sindice.com>



## 2.8 R2R Framework

The R2R Framework<sup>25</sup> enables Linked Data applications which discover data on the Web, that is represented using unknown terms, to search the Web for mappings and apply the discovered mappings to translate Web data to the application's target vocabulary.

The R2R Framework currently consists of:

- the R2R Mapping Language<sup>26</sup>, a declarative mapping language for representing correspondences between terms from different RDF vocabularies.
- the R2R Java API<sup>27</sup>, a modular API which makes use of the mappings to transform Web data to a target vocabulary.

To use the R2R Framework, you need:

- Java 1.5 or higher to run R2R (check with `java -version` if you're not sure),

What to do:

1. Download and extract the latest R2R version<sup>28</sup> into a suitable location.
2. Test your installation with an example from the Examples section.
3. Write your own mappings using the R2R Mapping Language.

## 2.9 Silk Link Discovery Framework

The Silk Link Discovery Framework<sup>29</sup> is a tool for discovering relationships between data items within different Linked Data sources. It is available in different editions suiting different use cases.

Silk Single Machine<sup>30</sup> can be used to produce link sets once and publish them along with the data. Silk Server<sup>31</sup> can be set up to allow for linking to often changing link targets.

In order to use Silk Single Machine, you need:

- The latest version of the Silk Link Discovery Framework<sup>32</sup>.
- The most recent Java Runtime Environment.

What to do:

1. Write a Silk-LSL<sup>33</sup> configuration file to specify which resources should be interlinked.
2. Run Silk Single Machine:

```
java -DconfigFile=<Silk-LSL file> [-DlinkSpec=<Interlink ID>] [-Dthreads=<number of threads>] [-Dreload=(true/false)] -jar silk.jar
```

<sup>25</sup><http://www4.wiwiss.fu-berlin.de/bizer/r2r/>

<sup>26</sup><http://www4.wiwiss.fu-berlin.de/bizer/r2r/spec/#specification>

<sup>27</sup> <http://www4.wiwiss.fu-berlin.de/bizer/r2r/spec/#usingR2RAPI>

<sup>28</sup><http://sourceforge.net/projects/r2r/files/R2R%20API/>

<sup>29</sup><http://www4.wiwiss.fu-berlin.de/bizer/silk/>

<sup>30</sup><http://www4.wiwiss.fu-berlin.de/bizer/silk/singlemachine/>

<sup>31</sup><http://www4.wiwiss.fu-berlin.de/bizer/silk/server/>

<sup>32</sup><http://sourceforge.net/projects/silk2/>

<sup>33</sup>

3. Review Results: Open the output files designated in the Silk-LSL configuration and review the generated links.

## 2.10 Sig.ma

Sig.ma is a web application that does not require anything special to be installed, apart from a modern Web browser. To use it, first go to its website<sup>34</sup> and then enter either a list of keywords or the address of a resource in the search box. Sig.ma will query Sindice to find data about the queried words / resources and aggregate all the results into a data list. The controls on the right of the page allow for validating the data sources and wrongly ones can be filtered out. The final, filtered, set of data about the resource can then be saved for later use.

## 2.11 RelFinder

RelFinder display the relations between resources on the Web of Data, it is available online<sup>35</sup> and there is nothing to install for using. The main interface of the application has input boxes on the left part used to give a set of resources to look for relations among them. Just enter two or more keywords and press the button “find relations” to see which other concepts connect them. The data comes from a pre-defined set of SPARQL endpoints, if an endpoint is missing it can be added by clicking on the wrench icon.

If you want to set up your own RelFinder instance, you need:

- A Webserver
- the latest PHP version

Then download the following files from the RelFinder Google Code repository<sup>36</sup>:

- RelFinder.swf - the Flash file that is executed
- crossdomain.xml - the crossdomain specification
- Proxy2.php - if you do not use the solution with the crossdomain file
- config/Config.xml - the file that configures the datasets
- config/examples.xml - a file that defines some examples

Copy them into a RelFinder folder in your webserver's public directory.

Make sure that you use the same folder structure and set the access rights properly. Note also that using the proxy may destroy HTTP sessions, so better use the option with the crossdomain file instead.

## 2.12 OntoWiki

OntoWiki<sup>37</sup> is a Semantic Data wiki which enables the collaborative creation and publication of RDF knowledge bases as Linked Data.

How to set up OntoWiki:

1. First go to your web directory:

```
cd /var/www
```

2. Then clone our default repository:

<sup>34</sup><http://sig.ma>

<sup>35</sup><http://relfinder.dbpedia.org/relfinder.html>

<sup>36</sup> <http://code.google.com/p/relfinder/>

<sup>37</sup> <http://ontowiki.net/Projects/OntoWiki>

```
hg clone https://ontowiki.googlecode.com/hg/ myow
```

- If you do not already have a Zend in your include-path you need to download it. For this, you can run:

```
make zend
```

in your `myow` base directory.

If you can't use `make`, you can download Zend manually too:

```
rm -rf libraries/Zend
curl -O http://framework.zend.com/releases/ZendFramework-1.11.1/ZendFramework-1.11.1-minimal.tar.gz || wget
http://framework.zend.com/releases/ZendFramework-1.11.1/ZendFramework-1.11.1-minimal.tar.gz
tar xzf ZendFramework-1.11.1-minimal.tar.gz
mv ZendFramework-1.11.1-minimal/library/Zend libraries
rm -rf ZendFramework-1.11.1-minimal.tar.gz ZendFramework-1.11.1-minimal
```

Your OntoWiki installation is now available under: <http://localhost/myow/>.

Configuration:

- The most important configuration step is the database configuration. First you have to copy these files:

```
cd /path/to/myow
cp config.ini-dist config.ini
```

- Then you need to change the database configuration in your `config.ini`: `%%store.zenddb.dbname = ontowiki ; your configured DB name`  
`store.zenddb.username = ow ; the username to your database server`  
`store.zenddb.password = ow ; the pass for this user`  
`store.zenddb.dbtype = mysql ; at the moment only mysql tested ;`  
`store.zenddb.host = localhost ; default is localhost %%`
- After that, go to your OntoWiki URL in your browser (e.g. `'http://localhost/myow/'`).
- There is a First Steps tutorial online<sup>38</sup>.

## 2.13 SPARQL Views

The SPARQL Views plugin for Drupal enables the integration of data available at SPARQL end points into Drupal powered Web sites. Because it is a plugin for Drupal, a working installation of Drupal is needed. The plugin is available on GitHub<sup>39</sup>, it requires Drush<sup>40</sup> to be compiled and Git to be downloaded.

Installation procedure:

- Clone the GitHub repository

```
git clone https://github.com/linclark/sparql_views
```

- Compile the plugin

```
drush make --no-core sites/all/modules/sparql_views/sparql_views.make
```

<sup>38</sup> <http://ontowiki.net/Projects/OntoWiki/FirstSteps>

<sup>39</sup> [https://github.com/linclark/sparql\\_views](https://github.com/linclark/sparql_views)

<sup>40</sup> <http://drupal.org/project/drush>

- In your Drupal Web site, enable SPARQL Views, RDFx, Views, and Views UI. If you want to use the SPARQL Views drag-and-drop interface, enable SPARQL Views UI and its dependencies.
- Go to the Tools tab in Views and Disable the Views data cache. If you want to use SPARQL Views UI, you unfortunately have to turn off Views JavaScript currently as well.

Usage:

To create a query, and get its associated results as a view, go to the Settings pane in Views, in left hand column under Advanced Settings. Queries can be input in two ways:

- Enter your prefixes and select query directly in the Settings pane. The select query should be in the form `SELECT * WHERE {?s ?p ?o} LIMIT 100`. You will want to limit your query if it returns a large result, otherwise you might get the white screen of death because of PHP timeout.
- Use the drag-and-drop interface by clicking on 'Build query'. This is currently very limited and can only create conjunctive queries.

After entering the query, any variables used will show up as fields. If you do not see fields, check to ensure your query saved. If it did, ensure that you have the Views data cache turned off.

### 3 Example

The following example demonstrates the publication and consumption process for the Community Research and Development Information Service (CORDIS)<sup>41</sup>.

The example makes use of the following tools:

- Neologism
- D2R Server
- Silk Link Discovery Framework
- RelFinder

Prerequisites:

- The CORDIS data set is originally available in a relational database.

Outline of the step by step publication process:

1. Model the ontology for CORDIS using Neologism.
2. Publish the data set as Linked Data using D2R Server.
3. Find Linked Data sets that thematically overlap with CORDIS and that can be linked to on CKAN<sup>42</sup>.
4. Use the Silk Link Discovery Framework to interlink CORDIS with other data sets and publish the links along with CORDIS.

Each publication step will be described in detail.

The Linked Data version of CORDIS can be consumed in different ways. We will show how to set up RelFinder to explore relations between entities in the CORDIS Linked Data set.

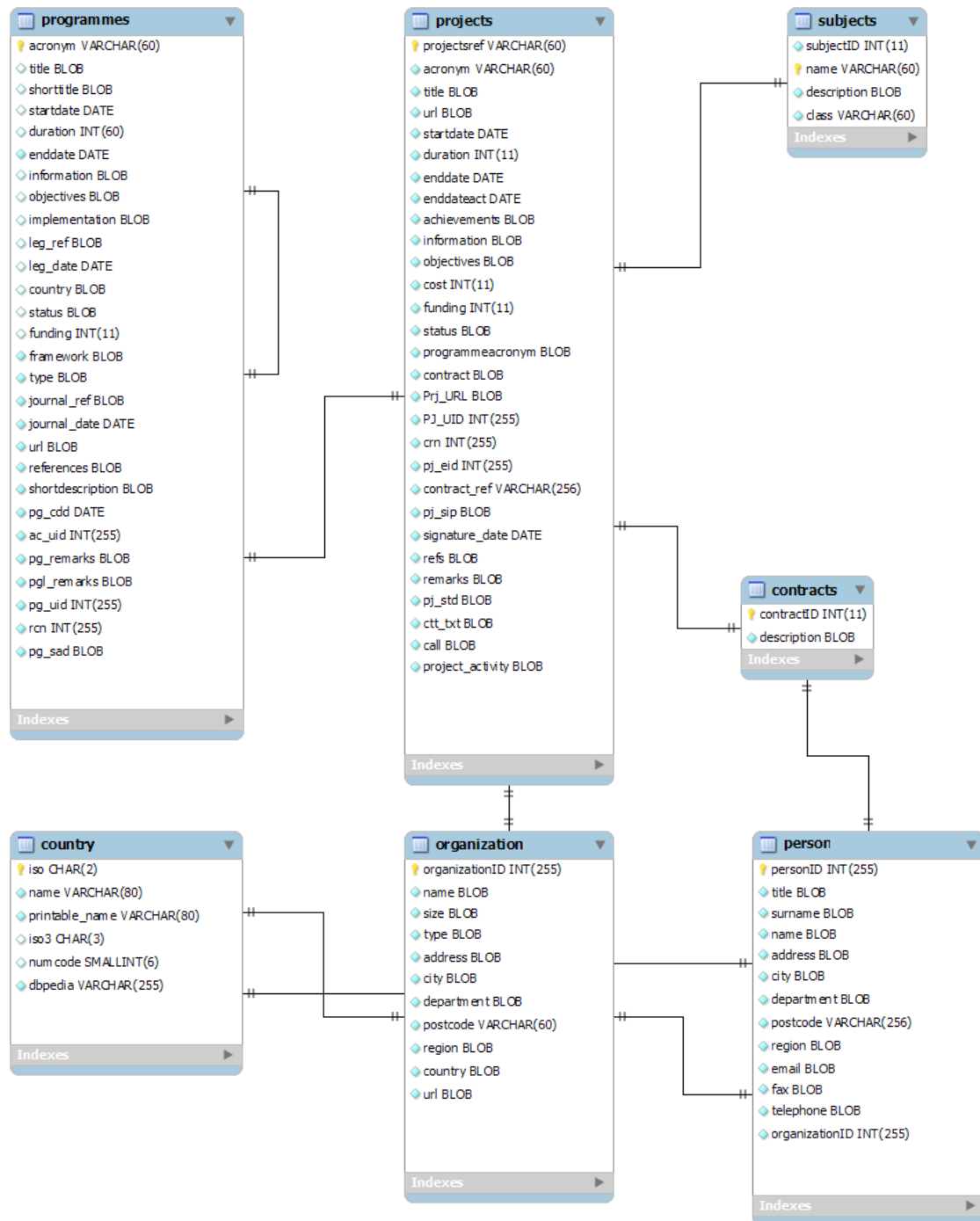
---

<sup>41</sup> <http://cordis.europa.eu/>

<sup>42</sup> <http://ckan.net/group/lodcloud>

### 3.1 Prerequisites

The CORDIS data set originally available in a relational database with the following database schema:

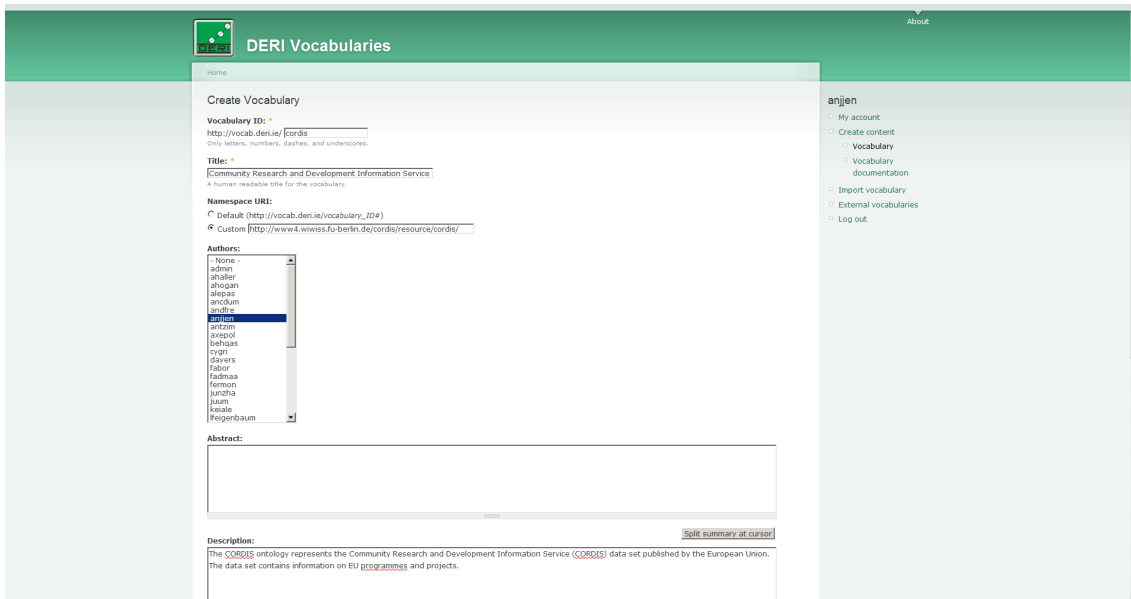


### 3.2 Neologism: Modeling the CORDIS ontology

Set up Neologism as described in section 2.1.

Login to your Neologism instance.

Create a new vocabulary for CORDIS (*Create content > Vocabulary*).

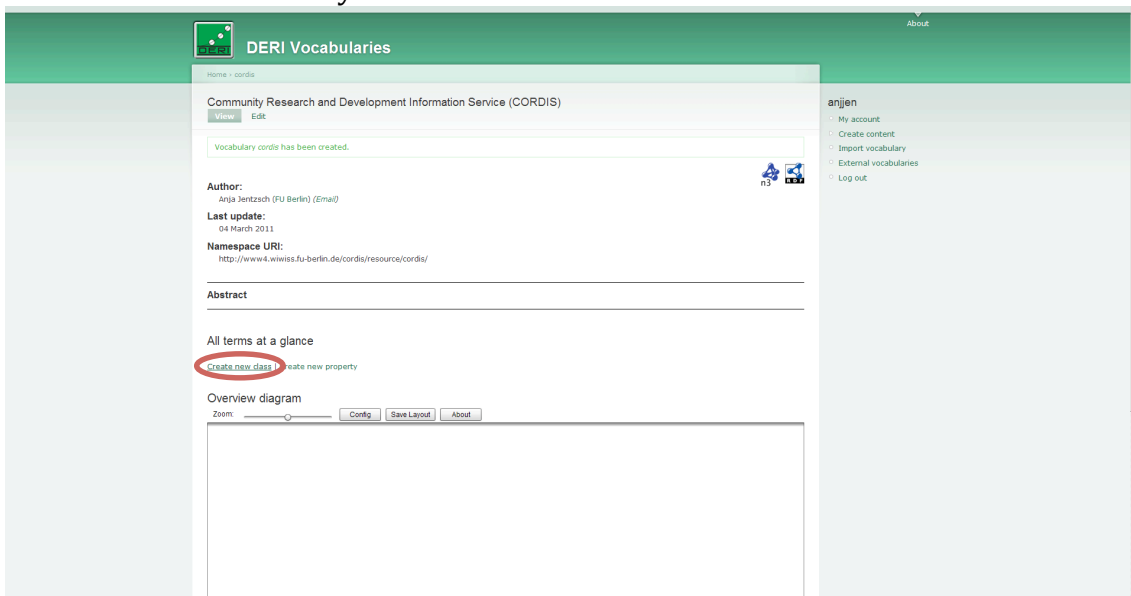


Provide at least the vocabulary ID, title, namespace URI, authors and description of the vocabulary.

For CORDIS provide e.g. the following parameters:

- Vocabulary ID: cordis
- Title: Community Research and Development Information Service (CORDIS)
- Description: The CORDIS ontology represents the Community Research and Development Information Service (CORDIS) data set published by the European Union. The data set contains information on EU programmes and projects.
- Namespace URI: http://www4.wiwiss.fu-berlin.de/cordis/resource/cordis/

Then save the vocabulary.

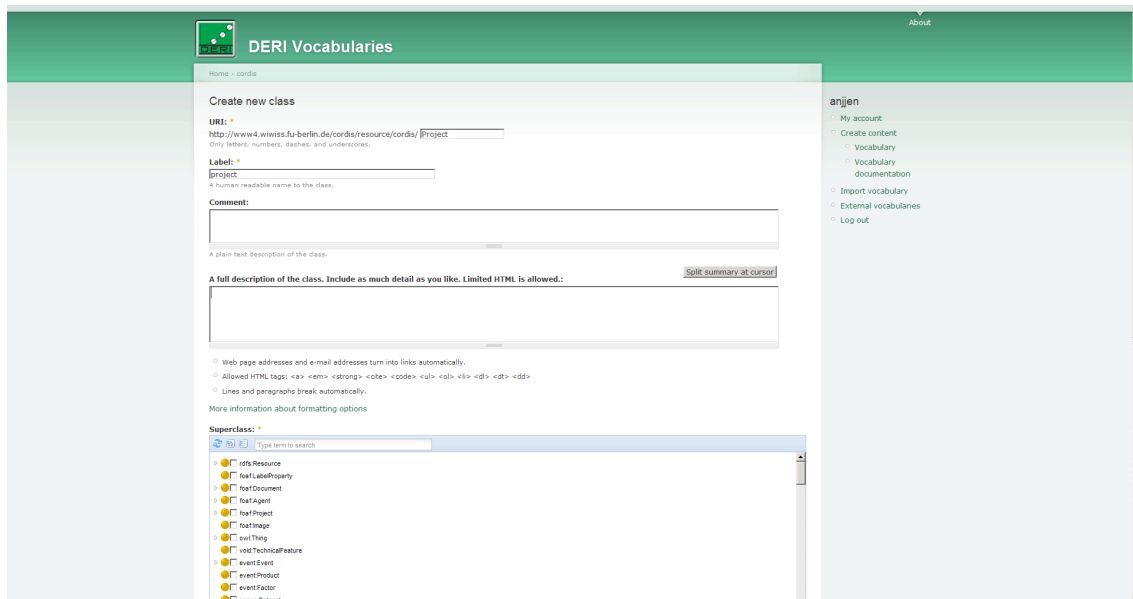


Create classes for each concept in the data set (*Create new class* button in vocabulary view).

For CORDIS these are e.g.:

- Project
- Programme

- Organization
- Person

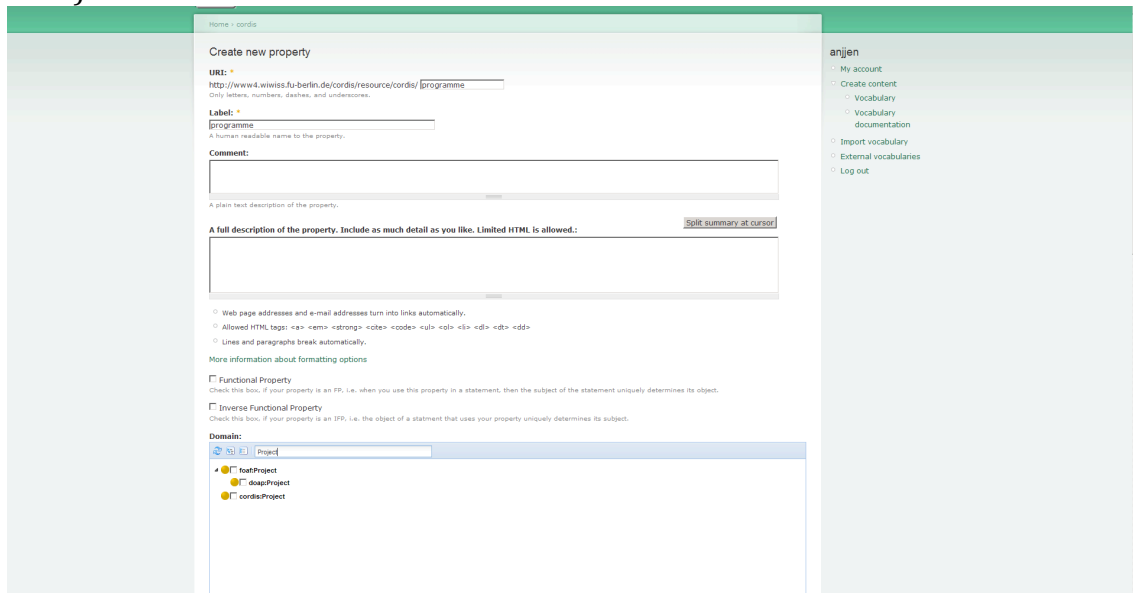


When adding a class, provide an URI and a label. Furthermore, a description of the class can be added and super classes can be chosen.

For the class Project provide the following parameters:

- URI: Project (the namespace is already given by the vocabulary namespace)
- Label: project
- Comment: A project is a collaborative enterprise involving several project partners.

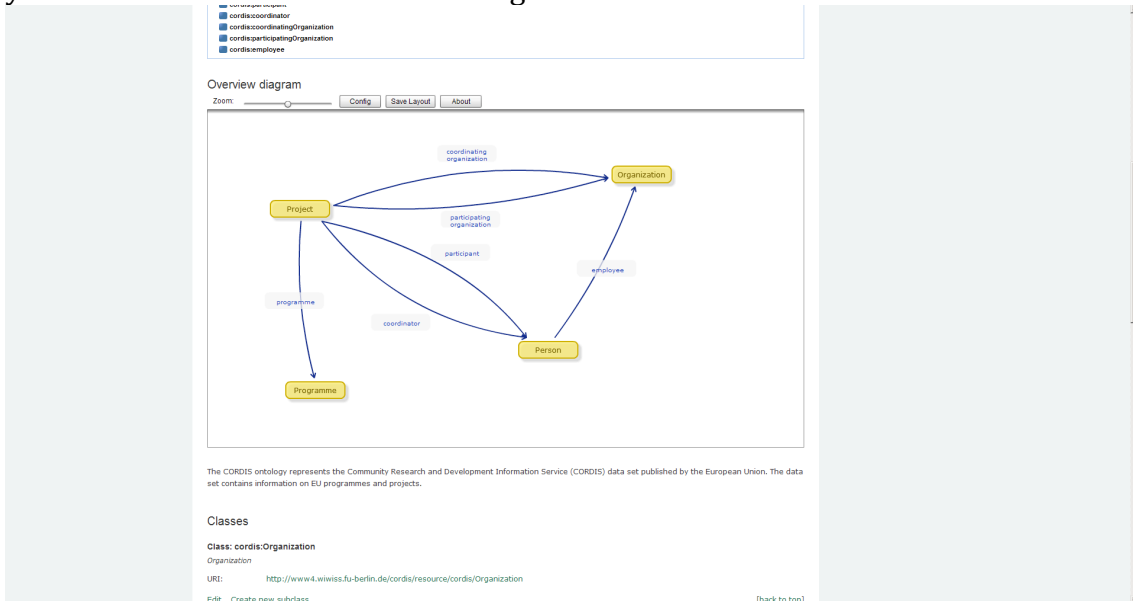
Then add properties to your classes (*Create new property* button in vocabulary view). These are class relations as well as class attributes.



An example for a property is the relation between projects and programmes. Therefore define the programme property with the domain cordis:Project and the range cordis:Programme.

Make sure to reuse vocabularies whenever possible. For links to websites containing more information on a concept, use *foaf:page* instead of defining your own website property.

When the concepts and their properties of the database (or data set) are modelled you can see them in the overview diagram:



You then can save the modelled vocabulary as N3 or RDF/XML (N3 or RDF/XML icons in vocabulary view).

The screenshot shows the DERI Vocabularies interface for the CORDIS vocabulary. Key elements include:

- Header:** DERI Vocabularies
- Page Title:** Community Research and Development Information Service (CORDIS)
- Message:** Property employee has been updated.
- Author:** Anja Jentzsch (FU Berlin) (Email)
- Last update:** 04 March 2011
- Namespace URI:** http://www4.wiwiss.fu-berlin.de/cordis/resource/cordis/
- Abstract:** (Empty)
- All terms at a glance:**
  - Classes: Organization | Person | Programme | Project
  - Properties: coordinatingOrganization | coordinator | employee | participant | participatingOrganization | programme
- Classes:**
  - cordis:Organization
  - cordis:Person
  - cordis:Programme
  - cordis:Project
- Properties:**
  - cordis:programme
  - cordis:participant
  - cordis:coordinator
  - cordis:coordinatingOrganization
  - cordis:participatingOrganization
  - cordis:employee
- Right Sidebar:** anjen, My account, Create content, Import vocabulary, External vocabularies, Log out.

Part of the CORDIS vocabulary in N3:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix vann: <http://purl.org/vocab/vann/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix cordis: <http://www4.wiwiss.fu-berlin.de/cordis/resource/cordis/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://vocab.deri.ie/cordis> a owl:Ontology;
dc:title "Community Research and Development Information Service (CORDIS)";
vann:preferredNamespaceUri "http://www4.wiwiss.fu-berlin.de/cordis/resource/cordis/";
vann:preferredNamespacePrefix "cordis";
dc:creator <http://vocab.deri.ie/cordis#anjen> .

<http://vocab.deri.ie/cordis#anjen> a foaf:Person;
foaf:name "Anja Jentzsch";
```



```

foaf:mbox <mailto:mail@anjajentzsch.de> .

<http://vocab.der.i.e/cordis#FU%20Berlin> a foaf:Organization;
foaf:member <http://vocab.der.i.e/cordis#anjjen>;
foaf:name "FU Berlin";
foaf:homepage <http://www.fu-berlin.de/> .

cordis:Project a rdfs:Class, owl:Class;
rdfs:isDefinedBy <http://vocab.der.i.e/cordis>;
rdfs:label "Project" .

cordis:Programme a rdfs:Class, owl:Class;
rdfs:isDefinedBy <http://vocab.der.i.e/cordis>;
rdfs:label "Programme" .

cordis:programme a rdf:Property;
rdfs:isDefinedBy <http://vocab.der.i.e/cordis>;
rdfs:label "programme";
rdfs:domain cordis:Project;
rdfs:range cordis:Programme .

...

```

### 3.3 D2R Server: Publishing the CORDIS data set as Linked Data

Set up D2R Server as described in section 2.2 until you reach step 3.

Write a D2RQ mapping file<sup>43</sup> to map your database to the previously defined ontology.

The following excerpt from the CORDIS D2RQ map defines prefixes, the server and database properties as well as the mappings for the `cordis:Project` and `cordis:Programme` classes and the ontology property `cordis:programme` which connects a project with a programme.

```

@prefix map: <file:/C:/Cordis/cordis.n3#> .
@prefix cordis: <http://www4.wiwiss.fu-berlin.de/cordis/resource/cordis/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix d2r: <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-server/config.rdf#> .
@prefix vocabClass: <http://www4.wiwiss.fu-berlin.de/cordis/vocab/resource/class/> .
@prefix vocabProperty: <http://www4.wiwiss.fu-berlin.de/cordis/vocab/resource/property/> .

map:Server a d2r:Server;
rdfs:label "D2R Server publishing the CORDIS data set";
d2r:baseURI <http://www4.wiwiss.fu-berlin.de/cordis/>;
d2r:port 2038;
.

map:database a d2rq:Database;
d2rq:jdbcDriver "com.mysql.jdbc.Driver";
d2rq:jdbcDSN "jdbc:mysql://127.0.0.1/cordis?autoReconnect=true";
d2rq:username "d2r";
.

map:Projects a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:class cordis:Project;
d2rq:uriPattern "Project/@@projects.projectsref@@";
d2rq:classDefinitionLabel "project"@en;
.

map:title a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:Projects;
d2rq:column "projects.title";
d2rq:property rdfs:label;
.

...

map:project2programme a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:Projects;
d2rq:property cordis:programme;
d2rq:propertyDefinitionLabel "Programme Acronym"@en;
d2rq:uriPattern "Programme/@@projects.programmeacronym|urlify@@";
d2rq:join "programmes.acronym = projects.programmeacronym"
.

```

<sup>43</sup> <http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/#specification>

```
map:Programmes a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:class cordis:Programme;
d2rq:uriPattern "Programme/@@programmes.acronym|urlify@@";
d2rq:classDefinitionLabel "Programme"@en;
.
...
```

Save the D2RQ file as *cordis.n3* to your D2R Server directory.

Then test the mapping by starting the server from inside this directory:

```
d2r-server.bat cordis.n3
```

You should see the server running at <http://localhost:2038> now.

If the mapping is complete, you can set up D2R Server as a service. On Windows use the install-service script:

```
install-service cordis cordis.n3
```

Find the example D2R Server publishing the CORDIS data set as Linked Data online<sup>44</sup>.

### 3.4 CKAN: Find thematically overlapping Linked Data sets

Even if you already know Linked Data sets CORDIS can be interlinked with, you should explore the data sets on CKAN<sup>45</sup>.

As an example we choose DBpedia<sup>46</sup> as a link target. Since DBpedia covers a lot of domains, we can interlink many entities in CORDIS and DBpedia. Suitable entity types for interlinking are e.g.: EU projects, EU programmes, organizations, countries, persons.

### 3.5 Silk Link Discovery Framework: Interlinking the CORDIS Linked Data set

Set up Silk Single Machine as described in section 2.9.

As an example we write a Silk link specification<sup>47</sup> file for EU projects in CORDIS and DBpedia. In both data sets we restrict the data set to EU projects with the `<RestrictTo>` element. We then require the label or acronym to match as well as either the website or EU project reference number. The minimum similarity of two data items which is required to generate a link between them is set to 95% in the `<Filter>` element, while only links between items with a similarity higher than 98% is written to the resulting link set, while the links below 98% are written to a verify file.

The following listing is the resulting Silk link specification:

```
<?xml version="1.0" encoding="utf-8" ?>
<Silk>
  <Prefixes>
    ...
  </Prefixes>

  <DataSources>
    <DataSource id="dbpedia" type="sparqlEndpoint">
      <Param name="endpointURI" value="http://dbpedia.org/sparql" />
      <Param name="graph" value="http://dbpedia.org" />
    </DataSource>

    <DataSource id="cordis" type="sparqlEndpoint">
      <Param name="endpointURI" value="http://www4.wiwiss.fu-berlin.de/cordis/sparql" />
    </DataSource>
  </DataSources>
</Silk>
```

<sup>44</sup> <http://www4.wiwiss.fu-berlin.de/cordis/>

<sup>45</sup> <http://ckan.net/group/lodcloud>

<sup>46</sup> <http://dbpedia.org>

<sup>47</sup> <http://www4.wiwiss.fu-berlin.de/bizer/silk/spec/>

## FP7-256975 LOD Around The Clock (LATC)

```

</DataSources>

<Interlinks>
  <Interlink id="projects">
    <LinkType>owl:sameAs</LinkType>

    <SourceDataset dataSource="dbpedia" var="a">
      <RestrictTo>
        ?a dbpedia-prop:wikiPageUsesTemplate &lt;http://dbpedia.org/resource/Template:Research-
Project&gt;
      </RestrictTo>
    </SourceDataset>

    <TargetDataset dataSource="cordis" var="b">
      <RestrictTo>
        ?b rdf:type cordis:Project
      </RestrictTo>
    </TargetDataset>

    <LinkCondition>
      <Aggregate type="average">
        <Aggregate type="max">
          <Compare metric="levenshtein">
            <TransformInput function="lowerCase">
              <Input path="?a/rdfs:label" />
            </TransformInput>
            <TransformInput function="lowerCase">
              <Input path="?b/rdfs:label" />
            </TransformInput>
          </Compare>
          <Compare metric="levenshtein">
            <TransformInput function="lowerCase">
              <TransformInput function="replace">
                <TransformInput function="stripUriPrefix">
                  <Input path="?a\dbpedia-prop:redirect" />
                </TransformInput>
                <Param name="search" value="_" />
                <Param name="replace" value=" " />
              </TransformInput>
            </TransformInput>
            <TransformInput function="lowerCase">
              <Input path="?b/rdfs:label" />
            </TransformInput>
          </Compare>
          <Compare metric="levenshtein">
            <TransformInput function="lowerCase">
              <Input path="?a/dbpedia-prop:title" />
            </TransformInput>
            <TransformInput function="lowerCase">
              <Input path="?b/rdfs:label" />
            </TransformInput>
          </Compare>
          <Compare metric="levenshtein">
            <TransformInput function="lowerCase">
              <Input path="?a/rdfs:label" />
            </TransformInput>
            <TransformInput function="lowerCase">
              <Input path="?b/cordis:acronym" />
            </TransformInput>
          </Compare>
          <Compare metric="levenshtein">
            <TransformInput function="lowerCase">
              <TransformInput function="replace">
                <TransformInput function="stripUriPrefix">
                  <Input path="?a\dbpedia-prop:redirect" />
                </TransformInput>
                <Param name="search" value="_" />
                <Param name="replace" value=" " />
              </TransformInput>
            </TransformInput>
            <TransformInput function="lowerCase">
              <Input path="?b/cordis:acronym" />
            </TransformInput>
          </Compare>
          <Compare metric="levenshtein">
            <TransformInput function="lowerCase">
              <Input path="?a/dbpedia-prop:title" />
            </TransformInput>
            <TransformInput function="lowerCase">
              <Input path="?b/cordis:acronym" />
            </TransformInput>
          </Compare>
        </Aggregate>
      </Aggregate type="max">
        <Compare metric="equality">
          <TransformInput function="stripPostfix">
            <Input path="?a/dbpedia-prop:website" />
            <Param name="postfix" value="/" />
          </TransformInput>
          <TransformInput function="stripPostfix">

```

```

    <Input path="?b/foaf:page" />
    <Param name="postfix" value="/" />
  </TransformInput>
</Compare>
<Compare metric="equality">
  <TransformInput function="regexReplace">
    <Input path="?a/dbpedia-prop:projectreference" />
    <Param name="regex" value="^([\^~])*~*0*([0-9]*)\s*$" />
    <Param name="replace" value="$2" />
  </TransformInput>
  <TransformInput function="regexReplace">
    <Input path="?b/cordis:reference" />
    <Param name="regex" value="^0*" />
    <Param name="replace" value="" />
  </TransformInput>
</Compare>
<Compare metric="equality">
  <Input path="?a/dbpedia-prop:projectreference" />
  <Input path="?b/cordis:reference" />
</Compare>
</Aggregate>
</Aggregate>
</LinkCondition>

<Filter threshold="0.95" />

<Outputs>
  <Output maxConfidence="0.98" type="file" >
    <Param name="file" value="cordis_dbpedia_projects_verify_links.xml"/>
    <Param name="format" value="alignment"/>
  </Output>
  <Output minConfidence="0.98" type="file">
    <Param name="file" value="cordis_dbpedia_projects_links.xml"/>
    <Param name="format" value="ntriples"/>
  </Output>
</Outputs>
</Interlink>
</Interlinks>
</silk>

```

Save the link specification as *silk\_cordis\_dbpedia.xml*. Then got to the directory where you put the Silk Single Machine jar file and run:

```
java -DconfigFile=silk_cordis_dbpedia.xml -jar silk.jar
```

The resulting 17 links are written into the *cordis\_dbpedia\_projects\_links.xml* file:

```

<http://dbpedia.org/resource/DAIDALOS> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/71242> .

<http://dbpedia.org/resource/SeCSE> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/72081> .
<http://dbpedia.org/resource/IMAGUQUANIM> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/75943> .

<http://dbpedia.org/resource/SALERO> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/79378> .

<http://dbpedia.org/resource/Stasis_%28EU_project%29> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/79477> .

<http://dbpedia.org/resource/BEINGRID> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/79512> .

<http://dbpedia.org/resource/SUPER> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/79373> .

<http://dbpedia.org/resource/AssessGrid> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/79340> .

<http://dbpedia.org/resource/EMANICS> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/80625> .

<http://dbpedia.org/resource/DAIDALOS> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/80687> .

<http://dbpedia.org/resource/UNIC_%E2%80%93Universal_satellite_home_Connection>
<http://www.w3.org/2002/07/owl#sameAs> <http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/80627> .

<http://dbpedia.org/resource/RESERVOIR> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/85304> .

<http://dbpedia.org/resource/TREAT-NMD> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/84926> .

<http://dbpedia.org/resource/SOA4All> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/85536> .

<http://dbpedia.org/resource/SecureChange> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/89030> .

```

```
<http://dbpedia.org/resource/ONTORULE> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/89260> .
<http://dbpedia.org/resource/ProVisG> <http://www.w3.org/2002/07/owl#sameAs>
<http://www4.wiwiss.fu-berlin.de/cordis/resource/Project/89375> .
```

Import the links into a database table (e.g. *cordis\_dbpedia\_projects*) by separating source and target into different columns and add the according property mapping to your D2RQ map:

```
map:ProjectDBpediaLink a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:Projects;
  d2rq:property owl:sameAs;
  d2rq:propertyDefinitionLabel "DBpedia link"@en;
  d2rq:join "projects.projectsref = cordis_dbpedia_projects.project_id";
  d2rq:uriColumn "cordis_dbpedia_projects.dbpedia_url";
  .
```

Your D2R Server for CORDIS now also serves the links for EU projects in DBpedia as Linked Data.

### 3.6 RelFinder: Explore the CORDIS Linked Data set

For exploring the CORDIS Linked Data set we use RelFinder<sup>48</sup>. Instead of using the showcase version as described in section 2.11, we set up a separate RelFinder version especially for CORDIS. The instructions on setting up the RelFinder are online at <http://relfinder.dbpedia.org/integrating.html>.

Try your RelFinder version by pointing your web browser to the URL where you set up RelFinder. Then add 2 to n entity names into the RelFinder “between” form. Then define the maximum path length between these entities and click the *Find Relations* button.

You can find the RelFinder for CORDIS online at: <http://www4.wiwiss.fu-berlin.de/cordis/relfinder/RelFinder.swf>.

---

<sup>48</sup> <http://relfinder.dbpedia.org/>