UNIVERSIDAD DE CASTILLA-LA MANCHA



Analyzing Linked Data tools for SHARK

Technical Report

Cristina Roda, Elena Navarro, Carlos E. Cuesta

September 2013

Architectural Knowledge (AK) has been an integral part of Software Architecture specification since its original inception, but it has been explicitly managed only recently. It can be described as a computational structure of design decisions and rationales; recent research emphasizes that availability must be complemented by an effective use of this information. Therefore, when systems are complex and large, an effective searching method becomes critical. We propose the use of Linked Data techniques to define and query AK, thus achieving a flexible storage and scalable search. Our approach suggests storing the network of decisions in RDF format to be retrieved efficiently by means of SPARQL queries. This format is, by definition, general and extensible. As a side effect, many different AK structures can be described this way, which then becomes a general format to describe AK. In that way, this work analyzes some significant features regarding to AK of several Linked Data tools in order to determine which ones are the best/worst for handling AK as Linked Data.

1 What is Open Data?

According to [1], *Open Data* can be defined as "data that can be freely used, reused and redistributed by anyone". So, this type of information is available to everyone in a computational format.

This idea of *openness* has some requirements in relation to data which are [1]:

- 1. Availability and Access, data must be available in a convenient form, preferably through the Internet as nowadays everybody have open and freely access to it everywhere.
- 2. *Reuse and Redistribution,* data must be provided allowing its easy reuse and redistribution.
- 3. *Universal Participation*, everyone must be able to use, reuse and redistribute this kind of data, without discrimination against any person or group of people.

We may find lots of types of Open Data, such as:

- 1. *Cultural Data*, information about cultural works, like titles or authors, normally stored by libraries, archives and museums.
- 2. *Science Data*, data produced by means of scientific research in any scientific field.
- 3. *Financial Data*, data from government or financial markets, like accounts, stocks, shares, etc.
- 4. *Statistic Data*, data produced by statistical offices.
- 5. *Geodata*, information used to make any type of maps.
- 6. *Weather Data*, data used to predict and understand the weather.
- 7. Transport Data, such as timetables, routes or on-time statistics.
- 8. *Environment Data*, data related to the natural environment (rivers, seas, etc.).

And *why should data be open?* These are the main reasons:

- Transparency. In a well-functioning, democratic society, citizens need to know what their government is doing. To do that, they must be able freely to access government data and information and to share that information with other citizens. Transparency isn't just about access, it is also about sharing and reuse — often, to understand material it needs to be analysed and visualized and this requires that the material be open so that it can be freely used and reused.
- 2. *Releasing social and commercial value*. In a digital age, data is a key resource for social and commercial activities. Everything from finding your local post office to building a search engine requires access to data, much of which is created or held by government. By opening up data, government can help drive the creation of innovative business and services that deliver social and commercial value.
- 3. Participation and engagement participatory governance or for business and organizations engaging with your users and audience. Much of the time citizens are only able to engage with their own governance sporadically maybe just at an election every 4 or 5 years. By opening up data, citizens are enabled to be much more directly informed and involved in decision-making. This is more than transparency: it's about making a full "read/write" society, not just about knowing what is happening in the process of governance but being able to contribute to it.

2 What is Linked Data?

As its name indicates, *Linked Data* implies using the Web in order to create typed links between data from different sources, i.e. it refers to a set of best practices for publishing and connecting structured data on the Web [2]. All this information produces a global data space, commonly called the *Web of Data* or *Semantic Web*.

Until now, we have been used the term Web of Documents due to the fact that the information connected on the Web was made up of simply documents, but today this paradigm is changing. Lots types of data can be interconnected through the Web, not only documents, so the architecture of the Web of Data tends to be very similar to the WWW one.

In this sense, the *Semantic Web* isn't just about putting data on the web [3]. It is about making links, so that a person or machine can explore the Web of Data. With Linked Data, when you have some of it, you can find other, related, data.

Tim Berners-Lee outlined the Linked Data principles in [3]:

- 1. Use URIs as names for things.
- 2. Use *HTTP URIs* so that people can look up those names.
- 3. If a URI is looked up, provide useful information, using the standards *RDF* and *SPARQL*.
- 4. Include *links* to other URIs so that they can discover more things.

Namely, a *URI* identify real world objects, abstract concepts, web documents and digital content. This URI must be a *HTTP URI* which can identify the same as URIs and enable these URIs to be dereferenced (or look up) over the HTTP protocol into a description of the identified object or concept. This description should follow the standard *RDF* which is a single data model for publishing structured data on the Web, but not a data format, so RDF data can be serialized in different formats, like RDF/XML, RDFa, Turtle, etc. Finally, this information represented as RDF has to be connected using *RDF links* that are hyperlinks in the Linked Data context which connect disparate data into a single global data space [4].

In summary, like the Web of Hypertext, the Web of Data is constructed with documents on the web. However, unlike the Web of Hypertext, where links are relationships anchors in hypertext documents written in HTML, the Web of Data has links between arbitrary things described by RDF [3].

3 What is Linked Open Data?

At this point, we should know what is Open Data and Linked Data as two separate things so that now it is easier to define what is *Linked Open Data* (LOD). It is [3] Linked Data released under an open license, which does not impede its reuse for free.

In this sense, Tim Berners-Lee [3] proposes a star scheme in order to know how good and available is our Linked Open Data. Under this scheme, you get one star if the information has been made public at all, i.e. if it has an open licence. The more stars you get, as you make it progressively more powerful, the easier for people to use.

- \star Available on the web (whatever format) but with an open licence, to be Open Data.
- ★★ Available as machine-readable structured data (e.g. excel instead of image scan of a table).

- $\star \star \star$ As (2) plus non-proprietary format (e.g. CSV instead of excel).
- $\star \star \star \star$ All the above plus use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff.

 $\star \star \star \star \star$ All the above plus link your data to other people's data to provide context.

4 Linked Data Tools

This section is intended to show how to use some Linked Data tools in order to manage Architectural Knowledge (AK). More specifically, Fig 1 displays the AK decision network that we are going to exploit within the different Linked Data tools.

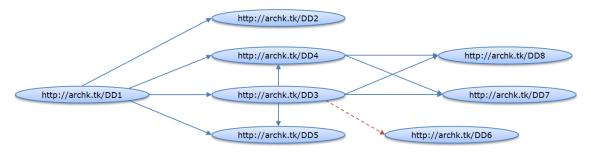


Fig 1 - AK decision network

Each http://archk.tk/DDX node represents a Design Decision (DD) in our network. All relationships between these nodes are constrains type, except for the connection between http://archk.tk/DD3 and http://archk.tk/DD6, which is a relation of exclusion.

The following RDF/XML code lines represent our AK decision network, written in an *.rdf* file.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rdf:RDF[
       <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
       <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
       <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
       <!ENTITY prop 'http://archk.tk/property#'>
       <!ENTITY kind 'http://archk.tk/dd/kinds#'>
       <!ENTITY conn 'http://archk.tk/dd/relation#'>
       <!ENTITY att 'http://archk.tk/dd/attributes#'>
       <!ENTITY rat 'http://archk.tk/dd/attributes/rationale#'>
]>
<rdf:RDF xml:base="http://archk.tk/DD3" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:conn="http://archk.tk/dd/relation#" xmlns:prop="http://archk.tk/property#"
xmlns:kind="http://archk.tk/dd/kinds#" xmlns:att="http://archk.tk/dd/attributes#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rat="http://archk.tk/dd/attributes/rationale#">
       <rdf:Description rdf:about="http://archk.tk/DD3">
               <att:author>Rafael</att:author>
               <att:category>Structural</att:category>
               <att:decision>Common Business Logic is Packaged as a Layer</att:decision>
               <att:risk rdf:datatype="&xsd;integer">0</att:risk>
               <att:scope>Universal</att:scope>
               <att:state>Approved</att:state>
               <att:rationale> The business logic of the application must be
encapsulated in a set of components for both the mobile application and the web server;
the common set of components is defined as a separate layer.
               </att:rationale>
               <rat:Layer>Business</rat:Layer>
```

```
<att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-
01T13:55:33.280368+01:00</att:timeStamp>
               <kind:classification>Existence</kind:classification>
               <conn:constrains rdf:resource="http://archk.tk/DD4" />
               <conn:constrains rdf:resource="http://archk.tk/DD5" />
               <conn:constrains rdf:resource="http://archk.tk/DD7" />
               <conn:constrains rdf:resource="http://archk.tk/DD8" />
               <conn:excludes rdf:resource="http://archk.tk/DD6" />
               <prop:type>DesignDecision</prop:type>
       </rdf:Description>
       <rdf:Description rdf:about="http://archk.tk/DD1">
               <att:author>Cris</att:author>
               <att:category>Structural</att:category>
               <att:decision>Using a three-layered architecture</att:decision>
               <att:risk rdf:datatype="&xsd;integer">0</att:risk>
               <att:scope>Universal</att:scope>
               <att:state>Approved</att:state>
               <att:rationale>Rationale1</att:rationale>
               <rat:Layer>Business</rat:Layer>
               <att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-
01T13:55:33.280368+01:00</att:timeStamp>
               <kind:classification>Existence</kind:classification>
               <conn:constrains rdf:resource="http://archk.tk/DD2" />
               <conn:constrains rdf:resource="http://archk.tk/DD3" />
               <conn:constrains rdf:resource="http://archk.tk/DD4" />
               <conn:constrains rdf:resource="http://archk.tk/DD5" />
               <prop:type>DesignDecision</prop:type>
       </rdf:Description>
       <rdf:Description rdf:about="http://archk.tk/DD2">
               <att:author>Cris</att:author>
               <att:category>Structural</att:category>
               <att:decision>Using a User-Interface layer</att:decision>
               <att:risk rdf:datatype="&xsd;integer">0</att:risk>
               <att:scope>Universal</att:scope>
               <att:state>Approved</att:state>
               <att:rationale>Rationale2</att:rationale>
               <rat:Layer>Business</rat:Layer>
               <att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-
01T13:55:33.280368+01:00</att:timeStamp>
               <kind:classification>Existence</kind:classification>
               <prop:type>DesignDecision</prop:type>
       </rdf:Description>
       <rdf:Description rdf:about="http://archk.tk/DD4">
               <att:author>Cris</att:author>
               <att:category>Structural</att:category>
               <att:decision>Defining classes of the mobile application for managing the
on-site ticketing process</att:decision>
               <att:risk rdf:datatype="&xsd;integer">0</att:risk>
               <att:scope>Universal</att:scope>
               <att:state>Approved</att:state>
               <att:rationale>Rationale4</att:rationale>
               <rat:Layer>Business</rat:Layer>
               <att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-
01T13:55:33.280368+01:00</att:timeStamp>
               <kind:classification>Existence</kind:classification>
               <conn:constrains rdf:resource="http://archk.tk/DD7" />
               <conn:constrains rdf:resource="http://archk.tk/DD8" />
               <prop:type>DesignDecision</prop:type>
       </rdf:Description>
       <rdf:Description rdf:about="http://archk.tk/DD5">
               <att:author>Cris</att:author>
               <att:category>Structural</att:category>
               <att:decision>Defining classes for managing the main services offered by
the web application</att:decision>
               <att:risk rdf:datatype="&xsd;integer">0</att:risk>
               <att:scope>Universal</att:scope>
               <att:state>Approved</att:state>
```

```
<att:rationale>Rationale5</att:rationale>
               <rat:Layer>Business</rat:Layer>
               <att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-
01T13:55:33.280368+01:00</att:timeStamp>
               <kind:classification>Existence</kind:classification>
               <prop:type>DesignDecision</prop:type>
       </rdf:Description>
       <rdf:Description rdf:about="http://archk.tk/DD6">
               <att:author>Cris</att:author>
               <att:category>Structural</att:category>
               <att:decision>Using a Data Management layer</att:decision>
               <att:risk rdf:datatype="&xsd;integer">0</att:risk>
               <att:scope>Universal</att:scope>
               <att:state>Approved</att:state>
               <att:rationale>Rationale6</att:rationale>
               <rat:Layer>Business</rat:Layer>
               <att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-
01T13:55:33.280368+01:00</att:timeStamp>
               <kind:classification>Existence</kind:classification>
               <prop:type>DesignDecision</prop:type>
       </rdf:Description>
       <rdf:Description rdf:about="http://archk.tk/DD7">
               <att:author>Cris</att:author>
               <att:category>Structural</att:category>
               <att:decision>Connecting the mobile application and the web
application</att:decision>
               <att:risk rdf:datatype="&xsd;integer">0</att:risk>
               <att:scope>Universal</att:scope>
               <att:state>Approved</att:state>
               <att:rationale>Rationale7</att:rationale>
               <rat:Layer>Business</rat:Layer>
               <att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-
01T13:55:33.280368+01:00</att:timeStamp>
               <kind:classification>Existence</kind:classification>
               <prop:type>DesignDecision</prop:type>
       </rdf:Description>
       <rdf:Description rdf:about="http://archk.tk/DD8">
               <att:author>Cris</att:author>
               <att:category>Structural</att:category>
               <att:decision>Packaging separately the API that connects the Business
Logic to the Data Layer</att:decision>
               <att:risk rdf:datatype="&xsd;integer">0</att:risk>
               <att:scope>Universal</att:scope>
               <att:state>Approved</att:state>
               <att:rationale>Rationale8</att:rationale>
               <rat:Layer>Business</rat:Layer>
               <att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-
01T13:55:33.280368+01:00</att:timeStamp>
               <kind:classification>Existence</kind:classification>
               <prop:type>DesignDecision</prop:type>
       </rdf:Description>
</rdf:RDF>
```

The Linked Data tools that we are going to talk about are *Virtuoso* (section 4.1), *Linked Media Framework* (section 4.2), *Apache Jena and Fuseki* (section 4.3), *TopBraid Suite* (section 4.4), *Sesame* (section 4.5), *Mulgara* (section 4.6), *RedStore* (section 4.7), *Callimachus* (section 4.8). In section 4.9, we are going to present other Linked Data tools that are not going to take part of the feature analysis in section 5, but we consider that they have to be mentioned.

4.1 Virtuoso

Virtuoso is a single data server, developed by *OpenLink Software*, which offers functionality from between traditional Relational data management to Linked Data server. In

particular, this hybrid product allows you to deal with the following areas [5], of which we are particularly interested in the italics marked ones:

- Relational data management
- RDF data management
- XML data management
- Free text content management and full text indexing
- Document Web server
- Linked Data server
- Web application server
- Web services deployment

In order to exploit this server with Linked Data, we may use *OpenLink Data Spaces* (ODS) [6] that is an expanding line of Virtuoso-based applications for establishing and managing data on the web and the emerging Semantic web. This application suite includes a web based file sharing platform, called *ODS Briefcase* [7], which allows users to control file access rights, search based on content and metadata. In addition, all resources are exposed as RDF data sets so file server functionality can be exploited by means of SPARQL query language for Semantic Web.

Some key features of ODS Briefcase are [6]:

- Automatic Metadata Management and Extraction (automatic extraction of file metadata from many file types)
- Powerful Full-Text Search (by metadata, path, filename, content words, mime-type, etc.)
- Flexible Data Access (via SPARQL, among others)
- Open Data Access (easy and transparent integration within any environment)
- Security
- Unified Central storage and access point
- Resource tagging (for categorize the content by user-defined tags)
- Shared Folders View (shows all resources that can be accessed by the user)
- Version Control

Especially for managing Linked Data, *ODS Briefcase* offers several characteristics as upload RDF files, validating their format previously according to a particular syntax, like XML; edit these files, consume uploaded data to the server by means of SPARQL [8], showing query results in various formats (HTML, XML, JSON, Javascript, NTriples, RDF/XML or spreadsheet).

In Fig 2, we may see the *OpenLink Briefcase* interface and the *textual_example_LD.rdf* file stored in the *AK_data* folder.

®ods	Profile edit Com	nmunities 🔻	Connectio	ons (o) 🔻 Me	essages (o) 🔻								Application Se	ttings Miss cris roda Logout Help
Applications edit			1.											Search Advanced
AddressBook 🗄 Bookmarks	((Metadata	Subscrip	1									Preferences About
Calendar														
(IIII) Weblog Sriefcase Mail More	Refresh	○ ^{Up}	Home	Shared Resources	New Folder	Rename	Сору	Move	Delete	Edit Properties	Tag	Upload	Bookmarklet	
	Path DAV/ho	ome/cristina	a.roda@alu.i	uclm.es/AK	_data		🕤 View	Details v	Filter Patte	rn		7 😣		
		Name		ags Size	Date Mo			Туре		Kind			wner	Group
	🗆 🗋 textu	ual_examp	le_LD.rdf	9 KB	2013-05-22	07:16:27	applicatio	on/rdf+xml	RDF Data (R	DF-XML, N3, 1	TURTLE)	cristina.rod	a@alu.uclm.es	cristina.roda@alu.uclm.es
		5												► A Software trivacy FAQ Developers

Fig 2 – OpenLink Briefcase

In Fig 3 and Fig 4**¡Error! No se encuentra el origen de la referencia.**, we can see the *OpenLink Virtuoso SPARQL Query* interface and the query results in HTML format, respectively.

	OpenLink Virtuoso SPARQL Query
This query page is designed to help you test OpenLink Virtuoso SP&RQL protocol endpoint. Consult the <u>Virtuoso Wiki page</u> describing the service or the <u>Online Virtuoso Documentation</u> section <u>RDF Database and SPARQL</u>	
There is also a rich Web based user interface with sample queries. You can access it at: fisparql.	
Query- Default Graph URI Use only local data (including data retrieved before), but do not retrieve more Query text SELECT * FROM <urstpm: catca="" securities=""> WHERE {?s ?p ?o}</urstpm:>	
Display Results As: HTML v II Rigorous check of the query Run Query Reset	.4

Fig 3 – OpenLink Virtuoso SPARQL Query

s	р	0
http://archk.tk/DD4	http://archk.tk /dd/attributes#timeStamp	2012-05-01T13:55:33.280368+01:00
http://archk.tk/DD4	http://archk.tk /dd/kinds#classification	Existence
http://archk.tk/DD4	http://archk.tk /dd/relation#constrains	http://archk.tk/DD7
http://archk.tk/DD4	http://archk.tk /dd/relation#constrains	http://archk.tk/DD8
http://archk.tk/DD4	http://archk.tk/property#type	DesignDecision
http://archk.tk/DD4	http://archk.tk /dd/attributes#author	Cris
http://archk.tk/DD4	http://archk.tk /dd/attributes#category	Structural
http://archk.tk/DD4	http://archk.tk /dd/attributes#decision	Defining classes of the mobile application for managing the on-site ticketing process
http://archk.tk/DD4	http://archk.tk/dd/attributes#risk	0
http://archk.tk/DD4	http://archk.tk /dd/attributes#scope	Universal
http://archk.tk/DD4	http://archk.tk/dd/attributes#state	Approved
http://archk.tk/DD4	http://archk.tk /dd/attributes#rationale	Rationale4
http://archk.tk/DD5	http://archk.tk /dd/attributes#timeStamp	2012-05-01T13:55:33.280368+01:00
http://archk.tk/DD5	http://archk.tk /dd/kinds#classification	Existence
http://archk.tk/DD5	http://archk.tk/property#type	DesignDecision
http://archk.tk/DD5	http://archk.tk /dd/attributes#author	Cris
http://archk.tk/DD5	http://archk.tk /dd/attributes#category	Structural
http://archk.tk/DD5	http://archk.tk /dd/attributes#decision	Defining classes for managing the main services offered by the web application
http://archk.tk/DD5	http://archk.tk/dd/attributes#risk	0
http://archk.tk/DD5	http://archk.tk /dd/attributes#scope	Universal
http://archk.tk/DD5	http://archk.tk/dd/attributes#state	Approved
	1	

Fig 4 – SPARQL query results in HTML format

In addition to this, we can create SPARQL queries based on a graph representation and run them using *OpenLink iSPARQL* [9]. Fig 5 shows the *OpenLink iSPARQL* interface. Notice that the data source (URL) has to be given, in our case *urn:ak_data*.

File Help Login		
QBE	Advanced Results	
📄 🚰 💾 🔀 🕟 🔍 🌋 🗑 💌 🕥 🤝 🕙 🐼 🖉 Data Source (URL):	urn:ak_data	+
		Inspector Please select a node or edge to view its properties. Schemas B C Schemas B C Schemas B C Schemas
Ouery options Ouery options Distinct Type: SELECT V Result size limit: 50 rows Leave empty for server maximum Query Metadata Endpoint Options (Virtuoso)	setting.	http://xmlns.com/fost/0.1/
Ubiquity Bookmarklet - drag this link to your browser's bookmark bar: <u>ISPAROL</u>	Copyright © 2009 OpenLink	: Software, Inc. OAT Version 2.8 Build \$Date: 2009/12/08 17:35:49 \$

Fig 5 – OpenLink iSPARQL

4.1.1 Architectural Knowledge as Linked Data: step by step with Virtuoso

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using Virtuoso server and *ODS Briefcase*.

1. First of all, we have to download the open version of Virtuoso Universal Server and install it in our machine (<u>http://virtuoso.openlinksw.com/download/</u>). At the time of writing, the latest release is 7.0, and it is available for Windows, Linux and Mac OS. We have to be registered to make the download, but registration is totally free.

Making Technology Work For You [®] Corporate Products Services	Support Sales Blogs Download	Offers	Seleccionar idioma
 Home News FAQ Download Offers Features Benefits Presentations Tutorials Big Wiki Documentation Articles White Papers Open-Source Project 	OpenLink Virtuoso Downloo All "Virtuoso Universal Server" installe Standalone "Client Connectivity Kit" II Release 7.0 Mindows XP/200x/itst/78 x86_64 (64-bit) Linux glibc 212 x86_64 (64-bit) glibc 212 x86_64 (64-bit) glibc 212 x86_64 (64-bit) glibc 25 x86_64 (64-bit) Mac OS X 10.7 x and higher (Universal)	rs include the full "Client Connecti	
Ask a question? Contact Us Support Case System Support Forum Live Chat (Online)	VAD Packages Release 6.4		

Fig 6 – OpenLink Virtuoso Download Page

2. With our credentials, now we can go to http://my.openlinksw.com/ods/ and sign in. Then, we are allowed to use ODS Briefcase by clicking the Briefcase link from the left vertical navigator bar.

®ods	Profile edit Communit	ies 🔻 Connecti	ons (o) 🔻 Messages (o				Application Sett	ings Miss cris roda	Logout Help
	OPENLIN								
Applications edit	BRI	FFCASE						Search	Advanced
	cristina.roda@alu.u	clm.es_Briefcase	e (Miss cris roda)					Preference	es About
Bookmarks	Browse Group	s Metadata	Subscriptions						
Calendar									
Weblog	• •		~ -		_ ^^	1.0			_
Briefcase) ዀ				🕺 🔒	🜳 💢 -		
More	Refresh Up	Home	Shared New Folder Resources	Rename Copy	Move Delete	Edit Tag Properties	Upload Bookmarklet		
morom	Path DAV/home/c	ristina.roda@alu.	uclm.es	😔 View	Details v Filter Patte	rn	7 😣		
		Name	🛆 Tags Size	Date Modified	Mime Type Kind	Owner	Group	Permissions	Actions
	AK_data		-	2013-05-22 05:18:47			cristina.roda@alu.uclm.es		*
	🔲 🛅 Gems			2013-05-22 05:08:31	cris	stina.roda@alu.uclm.es	cristina.roda@alu.uclm.es	rw-rr	×
	🗆 🖻 Items			2013-05-21 06:40:29	cris	tina.roda@alu.uclm.es	cristina.roda@alu.uclm.es	rw-rr	*
	OpenLink DenLink DenLink	Licenses		2013-05-21 06:40:29 2013-05-21 06:40:32			cristina.roda@alu.uclm.es cristina.roda@alu.uclm.es		»×
[Copyright About Fri	© 2000 - 2009 Ope vacy FAQ Developed	nLink Software

Fig 7 - OpenLink Briefcase overall interface

3. We can create a new folder to upload our RDF file into it. Click in "New Folder" and fill in the required information, selecting the type "RDF Upload Folder", as you see in Fig 8. Go to tab "RDF Upload", and specify the "Graph name", for example, *urn:ak_data*, as you see in Fig 9. Finally, click the "Create" button.

®ods	Profile adde	Communities	 Connectiu 	ons (o) 🔻 Messages	(0) 🔻						Application Settin	as Missicris roda Logout Help
Applications edit	Ŕ	BRIE	FCASE	(Miss cris roda)								Search Advanced
🖀 AddressBook – 📠 Bookmarks 💷 Calendar	Browse	Groups	Metadata	Subscriptions								- Tratel check (notat
Weblog Briefcase Mail More	Refresh		Home	Shared New Fold	ler Rename Co	py Move	Delet	e Edit Properties	Tag	Upload B	ookmarklet	
				cristina.roda@a	lu.uclm.es							
	Main	RDF Uploa	d									
				Folder name (*) rsioning Content		,	•					
					cristina.roda@alu.u							
					cristina.roda@alu.u			Select				
				Permissions	OWNE Group r w x r w ✔ ✔ ✔ □ □	x r w x	:					
				Full Text Search								
				ions Inheritance								
			Me	etadata Retrieval	Direct members v							
						Create	Cancel					

Fig 8 – Create a new folder with ODS Briefcase (1)

Sods	Profile edit Communities ▼ Connections (0) ▼ Messages (0) ▼	Application Settings Miss cris roda Logout Help
Applications edit	BRIEFCASE	Search Advanced
AddressBook	cristina.roda@alu.uclm.es_Briefcase (Miss cris roda)	Preferences About
Bookmarks	Browse Groups Metadata Subscriptions	
Calendar		
Weblog		
Sriefcase 🔄	🕝 🕜 🔂 🭙 🖳 🧭 🖄	n 📭 💥 🕺 🐧 🖓 🙀 🖓
	Refresh Up Home Shared New Folder Rename Copy Resources	r Move Delete Edit Tag Upload Bookmarklet Properties
More	Create folder in DAV/home/cristina.roda@alu.uclm.es	roperato
	Main RDF Upload	
	Graph name um:ak_data	
	Sponger (on/off)	
		Create Cancel

Fig 9 – Create a new folder with ODS Briefcase (2)

4. Once the new folder has been created, you can access it by clicking its name link. Then you may upload a new file to it. Click in the "Upload" option and specify the required information, selecting the file source from your computer, as you see in Fig 10. Finally, click the "Upload" button and a new RDF file should be created with the specified metadata.

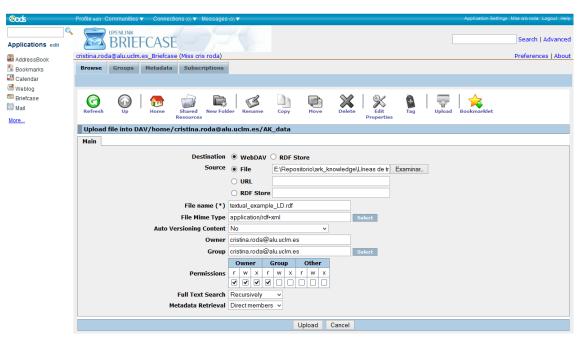


Fig 10 – Upload an RDF file with ODS Briefcase

5. We can check the inserted triples, going to *OpenLink Virtuoso SPARQL Query* endpoint: <u>http://my.openlinksw.com/sparql/</u>. There, we enter a query in the "Query text" area and press the "Run Query" button to execute it, as you see in Fig 11. The results of this query are the same as in Fig 4.

	OpenLink Virtuoso SPARQL Query
This query page is designed to help you test OpenLink Virtuoso SPARQL protocol endpoint. Consult the <u>Virtuoso Wirk page</u> describing the service or the <u>Online Virtuoso Documentation</u> section <u>RDF Database and SPARQL</u>	
There is also a rich Web based user interface with sample queries. You can access it at <u>fisparql</u> ,	
Query- Default Graph URI Use only local data (including data retrieved before), but do not retrieve more V Query text SELECT * FROM < vurn :ak_data>	
WHERE ((?# ?p ?o)	Ŀ
Display Results As: HTML V Rigorous check of the query Run Query Reset	

Fig 11 – A simple query using *OpenLink Virtuoso SPARQL Query*

For more information about these steps, you can visit this link.

4.2 Linked Media Framework (LMF)

Linked Media Framework [10] is an easy-to-setup server application that packages Semantic Web technologies to offer advanced services. This framework consists of two main elements: LMF Core and LMF Modules.

The LMF Core component is a Linked Data server that allows exposing data following the Linked Data Principles. In addition to the Linked Data Server, the LMF Core also offers a highly configurable Semantic Search service and a SPARQL endpoint.

Otherwise, we can enhance these LMF Modules from Linked Media Framework [10]:

- *LMF Semantic Search* that offers a highly configurable Semantic Search service.
- *LMF Linked Data Cache* that implements a cache to the Linked Data Cloud that is transparently used when querying the content of the LMF. In case a local resource links to a remote resource in the Linked Data Cloud and this relationship is queried, the remote resource will be retrieved in background and cached locally.

In addition to this, LMF implements its own authentication and authorization mechanism. In this manner, there are three pre-defined profiles: *simple* (allows read access from everywhere and write access only from localhost or other local interfaces), *standard* (allows read access from everywhere and write access only for authenticated users with manager role), and *restricted* (allows access only for authenticated users).

By default, LMF will use the *simple* profile, allowing only access from localhost. If we want to change this profile, we have to go to Security-profiles and then press *Save*.

We know that search over resources will always be domain-specific and will need to take into account the schema of the data. Therefore, the semantic search component of LMF provides only very simple and straightforward search functionality by default for three typical generic cases of metadata: *RDF* (rdfs:label, rdfs:comment), *DC* (dc:title, dc:description) and *SKOS* (skos:prefLabel, skos:altLabel, skos:definition).

However, in order to adapt the search component to our specific domain, LMF Admin Interface offers the possibility to define a *RDF Path Program*, i.e. a set of rules that map index fields to RDF properties or paths of RDF properties. For example, the following program (rdf) defines four fields (title, summary, tag, type):

```
@prefix hg : <http://www.holygoat.co.uk/owl/redwood/0.1/tags/> ;
title = rdfs:label :: xsd:string ;
summary = rdfs:comment :: lmf:text ;
tag = hg:taggedWithTag / hg:name :: xsd:string ;
type = rdf:type :: xsd:anyURI ;
```

In the most simple case (e.g. title), the rule maps an index field to exactly one RDF property. In more complex cases, the rule allows to follow a path of RDF properties; e.g. in the "tag" field, the rule would start at the current resource and follow the *hg:taggedWithTag* property, and from there it will follow the *hg:name* property and store it in the index. (Learn more about *RDF Path Program* in <u>http://code.google.com/p/lmf/wiki/ModuleSemanticSearch</u> and <u>http://code.google.com/p/ldpath/</u>).

4.2.1 Architectural Knowledge as Linked Data: step by step with LMF

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *Linked Media Framework*.

 First of all, we have to download LMF in <u>http://code.google.com/p/lmf/downloads/list</u>. At the time of writing, the latest standalone release is 2.3.1. It is recommended to download a standalone JAR file in order to install all LMF components and Apache Tomcat automatically.

- 2. Install the JAR file double-clicking on it, with default options. Notice that the installation path on windows systems may not contain whitespaces, which is a serious bug of Tomcat application server.
- 3. Once we have installed LMF, we have to start the LMF server. Go to Salzburg NewMediaLab folder, then Linked Media Framework folder and click in "Start Linked Media Framework" icon.



Fig 12 – Starting the LMF server

4. Now, we can access the administration interface by pointing our browser to http://{your_host_name}:8080/LMF and going to Administration-Linked Media Framework.

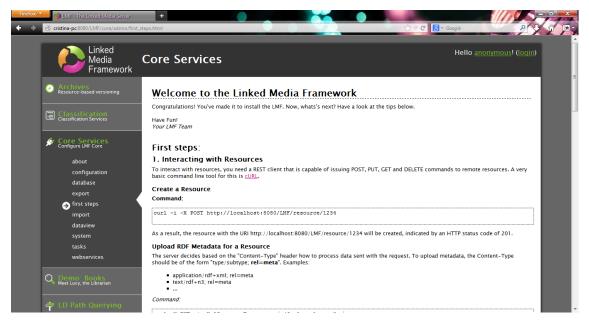


Fig 13 – The administration interface of LMF

5. Then, go to Core Services-import in order to upload an RDF file into LMF server. First, select the input source-type as *File*, choose the RDF file and click the *Import!* Button. A pop-up window notifies that the import was made successfully.

Firetox	+	
Linked Media Framework	Core Services	Hello <u>anonymous</u> ! (<u>login</u>)
Archives- Resource-based versioning Classification Services Configuration database export first steps import dataview system tasks webservices	Importer Import I. Select input source-type: File[URL 2. Select file: G\Universidad\Doctors Examins. 3. Import Source file Relation refite Re	
O Demo: Books Meet Lucy, the Librarian	-	-

Fig 14 – Adding an RDF file into LMF

Then, we can go to Core Services-dataview and see the RDF data already imported to LMF.

Firefox	🛀 🌍 Data Views	+		
÷ >	Cristina-pc:8080/LMF/core/admin/sgviz	ler/0.5/dataview.html		P 🗣 🔐 🖈 🗸
	ቍ LD Path Querying			Â
	Finked Data Caching Endpoints and Caches	100%		_
	O Reasoner sKWRL Programs			_
	C SKOS Thesaurus Management			_
	Queries and Updates			_
	Quartz Scheduling Integration	Resources Show I items without type relation		_
	O Converte	http://archk.tk/DD1		
	Access Control Management	http://archk.tk/dd/attributes#author	Cris	
		http://archk.tk/dd/attributes#scope	Universal	
		http://archk.tk/dd/attributes#decision	Using a three-layered architecture	
	Semantic Search Search Configuration and Cores	http://archk.tk/dd/attributes#rationale	Rationale1	
	Search Configuration and Cores	http://archk.tk/dd/attributes#category	Structural	
		http://archk.tk/dd/attributes#risk	0	
		http://archk.tk/property#type	DesignDecision	
	Social Module	http://archk.tk/dd/attributes/rationale#Layer	Business	
	social widgets and services	http://archk.tk/dd/attributes#state	Approved	
		http://archk.tk/dd/attributes#timeStamp	2012-05-01T11:55:33.000+0000	
	-A Charles	http://archk.tk/dd/relation#constrains	http://archk.tk/DD5	
_	Stanbol Interlinking and Enhancement	http://archk.tk/dd/relation#constrains	http://archk.tk/DD2	
		http://archk.tk/dd/relation#constrains	http://archk.tk/DD4	
		http://archk.tk/dd/relation#constrains	http://archk.tk/DD3	
		http://archk.tk/dd/kinds#classification	Existence	

Fig 15 – Data view with LMF

By clicking each resource, for example here *http://archtk.tk/DD1*, we can see its local description, as you see in Fig 16.

Local description of <u>http://archk.tk/D</u>	<u>D1</u> :		
property	has value	context	info
http://archk.tk/dd/attributes#author	Cris	knowledgespace:default	-
http://archk.tk/dd/attributes#category	Structural	knowledgespace:default	-
http://archk.tk/dd/attributes#decision	Using a three-layered architecture	knowledgespace:default	-
http://archk.tk/dd/attributes#rationale	Rationale1	knowledgespace:default	-
http://archk.tk/dd/attributes#risk	"0"^^xsd:integer	knowledgespace:default	-
http://archk.tk/dd/attributes#scope	Universal	knowledgespace:default	-
http://archk.tk/dd/attributes#state	Approved	knowledgespace:default	-
http://archk.tk/dd/attributes#timeStamp	"2012-05-01T11:55:33.000+0000"^^xsd:dateTime	knowledgespace:default	-
http://archk.tk/dd/attributes/rationale#Layer	Business	knowledgespace:default	-
http://archk.tk/dd/kinds#classification	Existence	knowledgespace:default	-
http://archk.tk/dd/relation#constrains	http://archk.tk/DD3	knowledgespace:default	-
http://archk.tk/dd/relation#constrains	http://archk.tk/DD4	knowledgespace:default	-
http://archk.tk/dd/relation#constrains	http://archk.tk/DD2	knowledgespace:default	-
http://archk.tk/dd/relation#constrains	http://archk.tk/DD5	knowledgespace:default	-
http://archk.tk/property#type	DesignDecision	knowledgespace:default	-

Fig 16 – LMF Linked Data Explorer

6. Finally, we can execute SPARQL queries in order to retrieve some RDF data from LMF server. Notice that LMF offers three different manners to make these queries.

We can use Flint SPARQL Editor, going to SPARQL-flint sparql editor.

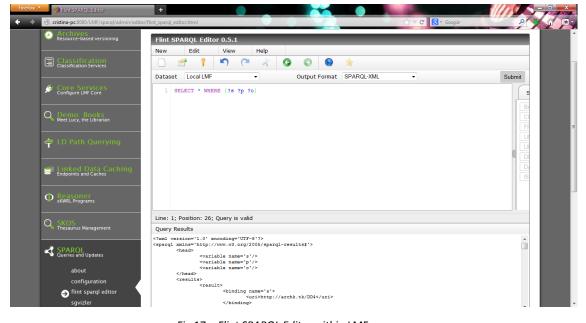


Fig 17 – Flint SPARQL Editor within LMF

Or we can use a Javascript SPARQL result set visualizer called *Sgvizler* that allows us to enter custom SPARQL queries and visualize their results in different kinds of charts. Go to SPARQL- sgvizler to use it. In Fig 18, we can see the output of the query at the bottom of the screen in a table format.

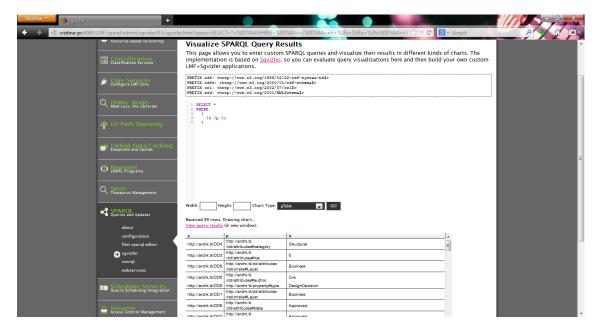


Fig 18 – Sgvizler SPARQL visualizer within LMF (1)

Notice that Sgvizler will show our data properly or report an error, depending on compatibility between our data and the chart type selected. Specifically, our Linked Data can only be represented with gOrgChart, gTable (Fig 18), sDefList, sList and sText chart types. In Fig 19, we can see an example of an error when we pretend to visualize our results as a bar chart (gBarChart type): "Data column(s) for axis #0 cannot be of type string".

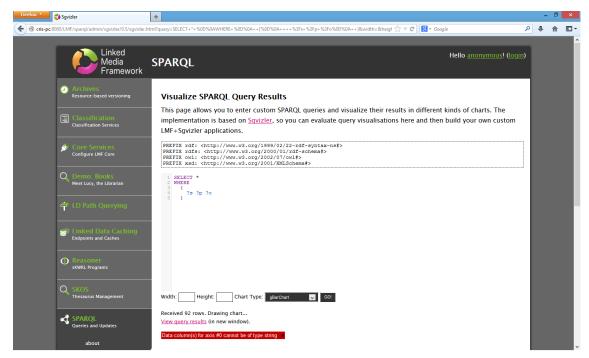


Fig 19 - Sgvizler SPARQL visualizer within LMF (2)

Finally, we can use *Snorql*, which is the SPARQL explorer for DBpedia¹, a project aiming to extract structured content from the information created as part of the Wikipedia project. Go to SPARQL-snorql to use Snorql.

¹ DBpedia official website: <u>http://dbpedia.org/About</u>

Firefox	uery=SELECT+*+WHERE+{%DD%DA++%3Fs+%3Fp+%3Fo%DD%DA	C ▼ C S- Google	
- Resource-based versioning	Exploring http://cristina-pc:8080/LMF/sp		
Classification Classification Services	SPARQL: PREFIX rdf: <http: 02="" 1999="" 22-rdf-syntax-ns#="" www.w3.org=""></http:>		
Configure LMF Core	<pre>PRETIX xdfs: chttp://www.wd.org/2000/01/zdf=schemai> PRETIX win: chttp://www.wd.org/2000/07/ouli> PRETIX wins: chttp://www.wd.org/2000/02/kow/coret> PRETIX ds: chttp://www.wd.org/2001/02/kow/coret> PRETIX foaf: chttp://www.on.org/doi/MacSchemai></pre>	Properties	
O Derro Books Meet Lucy, the Librarian	1 SELECT + WHERE (2 7s 7p ?o		
🛉 LD Path Querying			
Endpoints and Caching			
O Reasoner skwRL Programs			
Q SKOS Thesaurus Management			
SPARQL Queries and Updates	Results: Browse - Gol		
about	s p	0	
configuration	<http: archk.tk="" dd4="">@ <http: archk.tk="" attributes#category="" dd="">@</http:></http:>	"Structural"	
flint sparql editor	<http: archk.tk="" dd3=""> @ <http: archk.tk="" attributes#risk="" dd=""> @</http:></http:>	0	
sgvizler	<http: archk.tk="" dd8=""> @ <http: archk.tk="" attributes="" dd="" rationale#layer=""> @</http:></http:>		
A snorql	<http: archk.tk="" dd6=""> @ <http: archk.tk="" attributes#author="" dd=""> @</http:></http:>	"Cris"	
webservices	<htp: archk.tk="" dd8=""> タ <http: archk.tk="" property#type=""> タ <http: archk.tk="" dd1=""> タ <http: archk.tk="" attributes="" dd="" rationale#layer=""> ダ</http:></http:></http:></htp:>	"DesignDecision"	
	<http: archik.tk="" dd1="">E/ <http: archik.tk="" attributes="" dd="" rationale#layer="">E/ <http: archik.tk="" dd8="">E/ <http: archik.tk="" attributes#state="" dd="">E/</http:></http:></http:></http:>	"Business" "Approved"	
Quartz Scheduler Services	<http: archk.tk="" dd7=""> @ <http: archk.tk="" attributes#state="" dd=""> tP</http:></http:>	"Approved"	
Quartz Scheduling Integration	<http: archk.tk="" dd1="">@ <http: archk.tk="" attributes#risk="" dd="">@</http:></http:>	0	
	<http: archk.tk="" dd1=""> @ <http: archk.tk="" attributes#rationale="" dd=""> @</http:></http:>	"Rationale1"	
△ Security	<http: archk.tk="" dd3=""> @ <http: archk.tk="" dd="" relation#constrains=""> @</http:></http:>	<http: arohk.tk="" dd7=""> tP</http:>	
Access Control Management	<http: archk.tk="" dd2=""> @ <http: archk.tk="" attributes#rationale="" dd=""> @</http:></http:>	"Rationale2"	

Fig 20 – Snorql within LMF

4.3 Apache Jena and Fuseki

In general, *Apache Jena* [11] is a Java framework for building Semantic Web applications. In particular, *Jena* is a Java API for Semantic Web applications which can be used to create and manipulate RDF graphs. Therefore, Jena provides a collection of tools and Java libraries that allow you to develop semantic web and linked-data applications, tools and servers. This Jena framework includes the following features:

- An API for reading, processing and writing RDF data in XML, N-triples and Turtle formats
- An ontology API for handling OWL and RDFS ontologies
- A rule-based inference engine for reasoning with RDF and OWL data sources
- Stores to allow large numbers of RDF triples to be efficiently stored on disk
- A query engine compliant with the latest SPARQL specification
- Servers to allow RDF data to be published to other applications using a variety of protocols, including SPARQL

Thus, in order to provide support for RDF manipulation, Jena has object classes to represent properties, graphs (models), resources and literals. In addition, it provides constant classes for well-known schemas, such as RDF, RDFS, RDFa, Dublin Core or OWL, and also has some methods for reading and writing RDF as XML. Apart from that, Jena allows you to control the namespaces used on output with its prefix mappings; retrieve from a model the resource object, given its URI; access the properties of a resource, using some specific methods; query a model; or manipulate models as a whole by means of three operations: union, intersection and difference. So the primary use of Jena is to help you write Java code that handles RDF and OWL documents or descriptions.

On the other hand, *Fuseki* is a SPARQL server that offers the following services:

- SPARQL Query, SPARQL Update and file upload to a selected dataset
- Validators for SPARQL query and update and for non-RDF/XML formats

Notice that query outputs can be in several formats, such as JSON, XML, Text, CSV or TSV.

4.3.1 Architectural Knowledge as Linked Data: step by step with Apache Jena

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *Apache Jena*.

The aim of this tutorial is to provide a guide to prepare the Eclipse Platform for using Jena, not to explain in detail how to use their objects and classes themselves. To do this, visit <u>http://jena.apache.org/tutorials/rdf_api.html</u>.

- First of all, we have to download and install Eclipse. Go to <u>http://www.eclipse.org/downloads/</u> and download the latest version of the *Eclipse Classic* Platform. At the time of writing, the latest release is 4.2.2. Then, install it on your computer.
- 2. Open the Eclipse platform and create a new Java project. First, go to File-New-Project.

۵.				Java - Eclipse SDK	- 🗇 🗙
File	Edit Source Refactor Navigate Sear	h Proj	ect Run Window Help		
	New Alt+Shift-		Java Project		Quick Access 🗈 😫 Java
	Open File	Ľ	Project		E Outline 🛛 🗖
	Close Ctrl+	w 🌐	Package		An outline is not available.
	Close All Ctrl+Shift+		Class		
	Save Ctrl		Interface		
	Save As		Enum		
	Save All Ctrl+Shift		Annotation Source Folder		
	Revert		Java Working Set		
	Move		Folder		
		2			
		-5			
	Convert Line Delimiters To	<u> </u>			
6	Print Ctrl	P 🛃	Other Ctrl+N		
	Switch Workspace	→ T			
	Restart				
2	Import				
⊿	Export				
	Properties Alt+En	er			
	Exit				
			blems 🛛 🕼 Javadoc 😼 Declara	tion 🖳 Console 🖾	
		No cor	soles to display at this time.		

Fig 21 – Creating a new Java project with Eclipse (1)

Then select "Java Project" and click Next.

0			Java - Eclipse SDK		- ð ×
		rch Project Run Window Help			
(1)・日 原 白 × 参	- 0 - 9 -	₩ G • 🙋 🖋 • 🗄 • 🖗	× 5+ 6 × 6 × ≝		Quick Access
🛱 Package Explorer 🛿					 An outline is not available.
⊳ 😂 JenaTutorial ⊳ 鯞 SesameExample			New Project	- 🗆 🗙	
			Select a wizard Create a Java project Wizards:		
			type filter text		
			★ Java Project ★ Java Project from Existing Ant Buildfile ★ Deprint Project > ⇒ Creat > > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒ > ⇒		
		Problems @ Javadoc 😡 🛛 No consoles to display at this time		Finish Cancel	
			-		

Fig 22 - Creating a new Java project with *Eclipse* (2)

Give a name to the project, leaving the default options and click Finish.

•	Java - Eclipse SDK	- 0 ×
File Edit Source Refactor Navigate Search Project Run Window He		
C* ▼ B	THE STOP	Quick Access 🔡 🛱 Java
Il Package Explorer 🛛 🖓 🖓	😑 New Java Project – 🗆 🗙	□ 🗄 Outline 😫 🔍 □ An outline is not available.
> ₴ JenaTutorial > ⊉ SesameExample	Create a Java Project Create a Java project in the workspace or in an external location.	
	Project name: Tutorial	
	✓ Use default location	
	Location: C:\Users\Cris\workspace\Tutorial Browse	
	JRE	
	● Use an execution environment JRE: JavaSE-1.7 ∨	
	◯ Use a project specific JRE: jre7 v	
	Use default JRE (currently 'jre?') Configure JREs	
	Project layout	
	○ Use project folder as root for sources and class files	
	Create separate folders for sources and class files <u>Configure default</u>	
	Working sets	
😰 Problems @ Javadoc 🙆	Add project to working sets	ri ⊑ + ri + = o
No consoles to display at this tim		
	Reck Next > Finish Cancel	

Fig 23 – Creating a new Java project with Eclipse (3)

- Now, we have to download the Jena libraries in order to use them within Eclipse. Go to <u>http://sourceforge.net/projects/jena/files/Jena/Jena-2.6.4/</u> to download them and unzip the file downloaded to be ready to use in Eclipse.
- 4. Then, we can define a *user library* in Eclipse, i.e. a declaration of a library that we can reference from any project, which contains the Jena libraries. Go to Window-Preferences.

⊜			Java - Eclipse SDK	- 0	×
File Edit Source Refactor Navigate Sear	rch Project Run W	indow Help			
📑 • 🔛 🕲 🛆 🔌 🌼 • 💽 • 🏊 •	🕆 😋 🕈	New Window	f	Quick Access	Java
H Package Explorer ☆ → → Lena Tutorial → → → Tutorial → → Tutorial		New Editor Hiel Toolbar Open Perspective Open Perspective Save Prespective A Rest Perspective Close AI Perspective Close AI Perspective Close AI Perspective Close AI Perspectives Navigation > Preferences	8		
Tutorial					

Fig 24 – Adding Jena libraries into Eclipse (1)

Then go to Java-Build Path-User Libraries and click the "New..." button.

		Java - Eclipse SDK		- 0 ×
File Edit Source Refactor Navigate Search Project Run W	indow Help			
📑 • 🖩 🗟 🔌 🎄 • 🛛 • 🌯 • 🖶 🕫 • 💋 🛷	$ \underline{a} \cdot \overline{a} \cdot \overline{a} + \overline{a} + \overline{a} \underline{a}$		Quick Access	🔛 🐉 Java
File Edit Source Refactor Navigate Search Project Run W	ype filter text General General Ant Ant Ant Ant Appearance Appearance Appearance Casepath Variables Use tabaries Code Syle Code Syle Complet Debug Edud Path Casepath Res Junt Properties Files Editor Neroperties Files Editor Run/Debug Team Veroperties Files Editor Run/Debug Team Veroperties Files Editor Run/Debug Team	Preferences Image: Control of the state of the sta	E Outline ⊠ An outline is not available.	
	0	OK Cancel		
Tutorial				

Fig 25 - Adding Jena libraries into Eclipse (2)

Give a name to the user library, for example, *JenaLibs*, and click *OK*. Then, select the library that you have already created, and press "Add External JARs..." button.

● File Edit Source Refactor Navigate Search Pri		Java - Eclipse SDK				. 8 ×
	Project Run Window Help ⓒ ▼ 🍅 🖋 ▼ 🖢 ▼ 🎘 ▼ 🖘 ⇔ → ▼ 🔤			Qu	ick Access	🕄 🐉 Java
Pacatage Explorer ⊠ Pacatage Explorer ≅	Problems Supervision of the provided states o	Preferences User Libraries User Libraries User Libraries User Libraries	- C ×	- 🗆 🗄 Outline		
Tutorial						

Fig 26 - Adding Jena libraries into Eclipse (3)

Now we have to add all the .jar files from Jena libraries which are in the *lib/* directory of our Jena install directory.

0		Java - Eclipse SDK					- 8 ×
File Edit Source Refactor Navigate Search Pro							
📫 • 🗏 🕼 📥 🔌 🏇 • 💽 • 🂁 • 🕮 🧭	- 🙆 🖉 - 🖄 - 🖗 - 🗢 -	• S				Quick Access	📫 🐉 Java
🛱 Package Explorer 🛛 🗖 🗖					- 8	E Outline 🛛	
						An outline is not available.	
JenaTutorial	0	Preferences		_ 🗆 🗡			
SesameExample							
Victorial	۲	JAR Selection			×		
	🛞 🎯 💌 🕆 📕 🕨 Crit	s → workspace → Jena-2.6.4 → lib	♥ 🖒 Buscar er	ı lib	P		
	Organizar 👻 Nueva car	peta)III 🔹 🔟	0		
	🔶 Favoritos	Nombre	Fecha de modifica	Тіро	т, ^		
	🔒 Descargas	🙆 arq-2.8.7.jar	12/12/2010 13:17	Executable Jar File			
	Escritorio	▲ icu4j-3.4.4.jar	25/08/2010 19:03	Executable Jar File			
	Sitios recientes	🕌 iri-0.8.jar	25/08/2010 19:03	Executable Jar File			
		🍰 jena-2.6.4.jar	12/12/2010 15:57	Executable Jar File			
	🕽 Bibliotecas	🛓 jena-2.6.4-tests.jar	12/12/2010 15:57	Executable Jar File			
	Documentos	🗟 junit-4.5.jar	28/08/2010 12:11	Executable Jar File			
	📄 Imágenes	▲ log4j-1.2.13.jar	25/08/2010 19:14	Executable Jar File			
	🚽 Música 📃	lucene-core-2.3.1.jar	01/08/2009 18:21	Executable Jar File			
	Subversion	🛋 slf4j-api-1.5.8.jar	25/08/2010 19:03	Executable Jar File	- 11		
	🔠 Vídeos	🛃 slf4j-log4j12-1.5.8.jar	25/08/2010 19:03	Executable Jar File			
		💰 stax-api-1.0.1.jar	01/08/2009 18:21	Executable Jar File			
	🌏 Grupo en el hogar 🗸	🔹 wstx-asl-3.2.9.jar	01/08/2009 18:21	Executable Jar File	>		
		< re [*] arq-28.7 jar * "icu4j:34.4 jar * "iri-0.8 jar *	jena-2.6.4.ji v rjan,*zip Abi		~	ಡ	
Tutorial							

Fig 27 - Adding Jena libraries into Eclipse (4)

The result is that we have already created a user library called *JenaLibs* that contains the Jena libraries.

•	Preferences	- 🗆 🗙
type filter text General Ant Install/Update Install/Update Java Appearance Build Path Classpath Variables User Libraries Code Style Compiler Debug Editor Installed JREs JUnit Properties Files Editor Plug-in Development Run/Debug Team	User Libraries User libraries can be added to a Java Build path and bundle a number of extern libraries will be added to the boot class path when launched. Defined user libraries arq-2.8.7.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib arq-2.8.4.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib arq-2.6.4.lib arq-2.6.4.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib arq-2.6.4.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib arq-2.6.4.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib arq-2.6.4.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib arq-2.6.4.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib arq-2.6.4.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib arq-2.6.4.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib brace_junit-4.5.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib brace_junit-4.5.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib brace_junit-4.5.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib brace_junit-4.5.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib brace_junit-4.5.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib brace_junit-6.5.jar - C.\Users\Cris\workspace\Jena-2.6.4\lib brace_junit-6.jar - C.\Users\Cris\Workspace\Jena-2.6.4\lib brace_junit-	A a archives. System Add JARs Add JARs Add JARs Add External JARs Up Down Import Export
?	OK	Cancel

Fig 28 - Adding Jena libraries into Eclipse (5)

5. To use them into our Java project, we right-click on it in the explorer window and navigate to the Build Path menu option in order to add the *JenaLibs* library.

T	Java - Eclipse SDK	- D ×
File Edit Source Refactor Navigate Search Project	lun Window Help	
📫 • 🗒 🐚 🏝 🕅 🕸 • 🔕 • 💁 • 🔀 • 🕅 🞯 • 🗍	(A) + 2 + 2 + (+ + + +) ≤	Quick Access 🗈 🛱 🐉 Java
III Package Explorer ⊠ □ □ □ S ✓ → Ø JenaTutorial → Ø SesameExample	° 0	E Outline 🕄 🖓 🖓 🖓 🖓
⊳ 🔁 Tute New Go Into	•	
Open in New Window Open Type Hierarchy Show In Alt+Shi	F4 ₩ ≻	
Copy Qualified Name Paste C	+C -V ete	
Build Path	Re Link Source	
Source Alt+Sh Refactor Alt+Sh	+S > 🏟 New Source Folder	
import Z	Add External Archives	
 Refresh Close Project Close Unrelated Projects Assign Working Sets 	FS & Configure Build Path	± ₽ + 10 +
Run As Debug As Team Compare With Restore from Local History Configure	<pre>> / **** *****************************</pre>	
Properties Alt+	ter	
Tutorial		

Fig 29 - Adding Jena libraries into our Java project (1)

Then, select the option "User Library" and click Next.

⊜	Add Library	– 🗆 ×
Add Library		8
Select the library type to add.		
JRE System Library JUnit		
Plug-in Dependencies User Library		
? < Back	Next > Finish	Cancel

Fig 30 - Adding Jena libraries into our Java project (2)

Select JenaLibs and click Finish in order to add it to the classpath of our project.

Add Library	- • ×
User Library Select a library to add to the classpath.	5
User libraries:	
✓ ■JenaLibs	User Libraries
C Back Next > Finish	Cancel

Fig 31 - Adding Jena libraries into our Java project (3)

And now, we can use Jena within Eclipse (see this tutorial <u>http://www.iandickinson.me.uk/articles/jena-eclipse-helloworld/</u> for more detail).

4.3.2 Architectural Knowledge as Linked Data: step by step with Fuseki

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *Fuseki*.

- 1. First of all, we download the *Jena-Fuseki* package from <u>http://www.apache.org/dist/jena/binaries/</u>. It is a ZIP file that we have to unzip in order to use it.
- 2. Once we have this file unzipped, we can run the Fuseki server, writing the following line in the console: *fuseki-server --update --mem /ds*, where *--update* option allows us to update the dataset */ds*, and *--mem* option creates it as an empty and non-persistent dataset.

🔤 C:\Windows\system32\cmd.exe - fuseki-serverupdatemem /ds	
11/05/2013 22:05 14.273 LICENSE	λ.
11/05/2013 22:05 1.425 log4j properties	
11.05/2013 22:05 1.952 NOTICE	
11/05/2013 22:11 <dir> pages 11/05/2013 22:05 3.096 ReleaseNotes.txt</dir>	
11/05/2013 22:05 3.076 ReleaseNotes.txt 11/05/2013 22:05 17:997 s-delete	
11/05/2013 22:05 17.777 S-defete 11/05/2013 22:05 17.997 s-qet	
	1
11/05/2013 22:05 17.997 s-post	Ξ.
11/05/2013 22:05 17.997 s-put	
11/05/2013 22:05 17.997 s-query	
11/05/2013 22:05 17.997 s-update	
11/05/2013 22:05 17.997 s-update-form	
21 archivos 10.766.822 bytes	
4 dirs 79.217.184.768 bytes libres	
C:\Users\CRISTINA\Desktop\jena-fuseki-0.2.7>fuseki-serverupdatemem /ds	
17:10:47 INFO Server :: Dataset: in-memory 17:10:47 INFO Config :: Home Directory: C:\Users\CRISTINA\Desktop	
jena-fuseki-0.2.7	
17:10:48 INFO Server :: Dataset path = /ds 17:10:48 INFO Server :: Fuseki 0.2.7 2013-05-11T22:05:51+0100	
17:10:48 INFO Server :: rusekt 0.2.7 2013-05-11122:05:51-0100 17:10:48 INFO Server :: Started 2013/06/10 17:10:48 CEST on port	
3030	
2222	-

Fig 32 – Starting Fuseki server

Notice that if we run Fuseki with the --*loc=DIR* option instead of --*mem*, we can point it to an existing TDB dataset (and if it doesn't exist it will be created). As TDB datasets on disk are *persistent* anytime, when we run Fuseki with the --*loc* flag set to the same path (in this case *DIR*), it will use the same dataset and keep changes between sessions.

Since you'll have created the same dataset you can use Java code with the TDB jars to access this dataset from code directly without going via Fuseki.

3. As the server is started, we can go to <u>http://localhost:3030/</u> which is the Fuseki interface to manage our data.

Firefox 🕶 🔶 Fuseki - A SPARQL 1.1 Server +		
		P 4 6 10 -
	Fuseki	
Server Management		
Control Panel		
Documentation		
Fuseki documentation		
Validators		
SPARQL query validator SPARQL update validator RDF data validator RDF data validator IRI validator		
General SPARQL Service		
SPAROL query form		
Standards		
SPAROL 1.1 Query SPAROL 1.1 Update SPAROL 1.1 Protocol SPAROL 1.1 Protocol SPAROL 1.1 Uniform HTTP Protocol for Managing RDE Graphs		

- Fig 33 Fuseki interface
- 4. To upload our RDF data to the Fuseki server, we have to go to *Control Panel* and select our dataset, */ds*. Then, in the *Fuseki Query* interface, we select the RDF file to upload into the server and click *Upload*.

Fuseki Query						
Dataset: /ds	Jataset: /ds					
SPARQL Que	ry					
	.:					
	: JSON - tyle sheet (blank for none): /xml-to-html.xsl					
	ce the accept header to text/plain regardless					
Get R						
SPARQL Upd	ate					
	.:					
Perfor	m update					
File upload						
	!Universidad\Doctorado\Repositorio\Líi Examinar					
Graph: Uploa	default d					

Fig 34 – Uploading an RDF file into Fuseki server

5. Once our RDF file is uploaded into the Fuseki server, we can make queries to manage this data. Go to <u>http://localhost:3030/sparql.tpl</u> and write in the SPARQL Query text area the following query: "SELECT * WHERE { ?s ?p ?o }". Then select the output type and click Get Results.

🕑 localhost:3030/sparql.tpl		 🖁 🛪 Google	
	Fuseki Query		
: /ds			
RQL Query			
SELECT * WHERE { ?s ?p ?o }			
	.4		
Output: Text -			
XSLT style sheet (blank for none): /xml-to-html.xsl			
Force the accept header to text/plain regardless			

Fig 35 – Querying our data with Fuseki and SPARQL

Finally, the results of the query are presented, in this case, in text format:

3		q	0
		<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	
		<pre><http: archk.tk="" dd="" pre="" relation<=""></http:></pre>	
		<pre> <http: archk.tk="" dd="" pre="" relation<=""></http:></pre>	
		<pre> <http: archk.tk="" pre="" property#typ<=""></http:></pre>	
		<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	
<http: a<="" td=""><td>archk.tk/DD1></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td><pre>#timeStamp> "2012-05-01T13:55:33.280368+01:00"^^<http: 2001="" www.w3.org="" xmlschema#datetime=""></http:></pre></td></http:>	archk.tk/DD1>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	<pre>#timeStamp> "2012-05-01T13:55:33.280368+01:00"^^<http: 2001="" www.w3.org="" xmlschema#datetime=""></http:></pre>
<http: a<="" td=""><td>rchk.tk/DD1></td><td><pre> <http: archk.tk="" dd="" pre="" relation<=""></http:></pre></td><td>constrains> <http: archk.tk="" dd3=""></http:></td></http:>	rchk.tk/DD1>	<pre> <http: archk.tk="" dd="" pre="" relation<=""></http:></pre>	constrains> <http: archk.tk="" dd3=""></http:>
		<pre> <http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	
<http: a<="" td=""><td>archk.tk/DD1></td><td><pre><http: archk.tk="" dd="" pre="" relation<=""></http:></pre></td><td></td></http:>	archk.tk/DD1>	<pre><http: archk.tk="" dd="" pre="" relation<=""></http:></pre>	
<http: a<="" td=""><td>rchk.tk/DD1></td><td><pre> <http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td></td></http:>	rchk.tk/DD1>	<pre> <http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	
<http: a<="" td=""><td>archk.tk/DD1></td><td><pre> <http: archk.tk="" dd="" kinds#cla<="" pre=""></http:></pre></td><td>sification> "Existence"</td></http:>	archk.tk/DD1>	<pre> <http: archk.tk="" dd="" kinds#cla<="" pre=""></http:></pre>	sification> "Existence"
<http: a<="" td=""><td>archk.tk/DD1></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>##scope> "Universal"</td></http:>	archk.tk/DD1>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	##scope> "Universal"
<http: a<="" td=""><td>rchk.tk/DD1></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#author> "Cris"</td></http:>	rchk.tk/DD1>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#author> "Cris"
<http: a<="" td=""><td>rchk.tk/DD1></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#decision> "Using a three-layered architecture"</td></http:>	rchk.tk/DD1>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#decision> "Using a three-layered architecture"
<http: a<="" td=""><td>archk.tk/DD1></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td><pre>s/rationale#Layer> "Business"</pre></td></http:>	archk.tk/DD1>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	<pre>s/rationale#Layer> "Business"</pre>
<http: a<="" td=""><td>rchk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>sŧrisk> 0</td></http:>	rchk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	sŧrisk> 0
<http: a<="" td=""><td>rchk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td><pre>s/rationale#Layer> "Business"</pre></td></http:>	rchk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	<pre>s/rationale#Layer> "Business"</pre>
<http: a<="" td=""><td>archk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#scope> "Universal"</td></http:>	archk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#scope> "Universal"
<http: a<="" td=""><td>rchk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#author> "Cris"</td></http:>	rchk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#author> "Cris"
<http: a<="" td=""><td>rchk.tk/DD6></td><td><pre><http: archk.tk="" dd="" kinds#cla<="" pre=""></http:></pre></td><td>sification> "Existence"</td></http:>	rchk.tk/DD6>	<pre><http: archk.tk="" dd="" kinds#cla<="" pre=""></http:></pre>	sification> "Existence"
<http: a<="" td=""><td>archk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#category> "Structural"</td></http:>	archk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#category> "Structural"
<http: a<="" td=""><td>archk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>sfstate> "Approved"</td></http:>	archk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	sfstate> "Approved"
<http: a<="" td=""><td>rchk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#timeStamp> "2012-05-01T13:55:33.280368+01:00"^^<http: 2001="" www.w3.org="" xmlschema#datetime=""></http:></td></http:>	rchk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#timeStamp> "2012-05-01T13:55:33.280368+01:00"^^ <http: 2001="" www.w3.org="" xmlschema#datetime=""></http:>
<http: a<="" td=""><td>rchk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#rationale> "Rationale6"</td></http:>	rchk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#rationale> "Rationale6"
<http: a<="" td=""><td>archk.tk/DD6></td><td><pre><http: archk.tk="" pre="" property#tvg<=""></http:></pre></td><td>> DesignDecision"</td></http:>	archk.tk/DD6>	<pre><http: archk.tk="" pre="" property#tvg<=""></http:></pre>	> DesignDecision"
<http: a<="" td=""><td>rchk.tk/DD6></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#decision> "Using a Data Management laver"</td></http:>	rchk.tk/DD6>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#decision> "Using a Data Management laver"
<http: a<="" td=""><td>rchk.tk/DD4></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#decision> "Defining classes of the mobile application for managing the on-site ticketing process"</td></http:>	rchk.tk/DD4>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#decision> "Defining classes of the mobile application for managing the on-site ticketing process"
<http: a<="" td=""><td>archk.tk/DD4></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#rationale> "Rationale4"</td></http:>	archk.tk/DD4>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#rationale> "Rationale4"
<http: a<="" td=""><td>rchk.tk/DD4></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>#scope> "Universal"</td></http:>	rchk.tk/DD4>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	#scope> "Universal"
		<pre><http: archk.tk="" pre="" property#tyg<=""></http:></pre>	
		<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	
		<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	
		<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	
		<pre><http: archk.tk="" dd="" pre="" relation<=""></http:></pre>	
		<pre></pre>	
<http: a<="" td=""><td>rchk.tk/DD4></td><td><pre><http: archk.tk="" dd="" kinds#cla<="" pre=""></http:></pre></td><td>sification> "Existence"</td></http:>	rchk.tk/DD4>	<pre><http: archk.tk="" dd="" kinds#cla<="" pre=""></http:></pre>	sification> "Existence"
<http: a<="" td=""><td>rchk.tk/DD4></td><td><pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre></td><td>atriako I.0</td></http:>	rchk.tk/DD4>	<pre><http: archk.tk="" attribute<="" dd="" pre=""></http:></pre>	atriako I.0

Fig 36 – Query results with Fuseki and SPARQL

4.4 TopBraid Suite

TopBraid Suite [12] is a compendium of products that offers semantic technology to help user in several scopes of applicability, like connect data, systems and infrastructure or build flexible applications from linked data models.

All components of the suite work within an open architecture platform built specifically to implement W3C standards for integration and combination of data drawn from diverse sources (see Fig 37). These components are *TopBraid Composer*, for design data, *TopBraid Ensemble*, to assemble data, and *TopBraid Live*, to interact with data. They will be explained next.

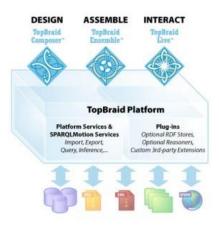


Fig 37 – TopBraid Suite Platform

TopBraid Composer (TBC) [13] is an Eclipse plug-in that offers an enterprise-class modelling environment for developing Semantic Web ontologies, building Semantic applications and converting data and models to/from RDF or OWL. Therefore, this platform provides complete support for developing and managing ontologies and Linked Data.

In that way, these are the main features of TopBraid Composer:

- Editing Ontologies and RDF Data
- Developing Semantic Web Applications
- Application Development Tools
- Importing Data Sources
- Exporting Data Sources
- Working with XML and Table files
- Developing Web Data (Microdata, RDFa)
- Data Integration

In addition, TopBraid Composer is fully compliant with W3C standards, such as RDF, RDFS, OWL or SPARQL. In this manner, all resources in the RDFS vocabulary are available in TopBraid Composer, as well as all OWL language constructs. TBC also ensures not to have syntactic errors in RDF data created in a text editor.

TBC provides an SPARQL tab to execute SPARQL queries and we can also save query results into a file. Furthermore, TBC can be used with Subversion (SVN) and allows opening some file formats within its environment, like RDF/XML, Turtle, N3, N-triples, Excel documents and XML.

On the other hand, we have *TopBraid Ensemble*, [12] a semantic web application assembly toolkit for rapid configuration and delivery of dynamic business applications, so it allows us to create model-driven applications. This tool provides a quick route from an ontology and RDF data to a working web application using pre-built model-driven *Rich Internet Application* (RIA) components implemented with Adobe Flex. Notice that to indicate how this application works is beyond the scope of this section, due to it is unlikely that we will use it.

Finally, we have *TopBraid Live* (TBL) [14] which is an enterprise SOA-capable Semantic Web application and Linked Data platform. It is a server for deploying flexible, model-driven applications and dynamic, on-demand integration of data from diverse sources.

So, it has so many benefits, such as:

- A unified, standards-based data integration process that combines data and schemas from internal, external, structured and unstructured sources using connectors to relational data, web services, XML files, JSON documents, spreadsheets, flat files, e-mail, content management systems and more.
- An orchestration engine for coordinating access to multiple sources (real time, ETL, etc.) including relational databases, XML, spreadsheets and web services.
- *Web Services* out-of-the-box RESTful Linked Data services plus a comprehensive library of ready-to-use modules and functions for rapid creation of custom composite web services from existing data and systems.
- Provenance and change management that completes audit trail of all changes, versioning.
- Business rules Flexible and powerful reasoning integrates RDFS and OWL inferencing with custom, domain-specific business rules.
- SPARQL Endpoint, given that all information managed by TopBraid is accessible through SPARQL. In addition to full SPARQL 1.1 Query Language support, TopBraid Live includes many extensions provided as SPARQL functions.
- *Linked Data*, given that all information managed by TopBraid is available as RESTful web services returning a choice of serializations.
- *Named graph architecture* TopBraid Live workspace is a quad store fully implementing named graph architecture.
- *Model-based* fully dynamic, flexible and agile.
- *Rapid application development* using pre-built UI components, application templates and modules.
- Support for the entire application development lifecycle seamless migration from development in TopBraid Composer to deployment in TopBraid Live.

There are two editions of this product. On the one hand, we have *TopBraid Live Enterprise Server* that provides full Internet and intranet access to the applications developed with TopBraid Suite. This server is deployed on a web container, such as Tomcat and supports multi-user access from distributed clients. Users may access it through web services and HTML and JavaScript applications, built using the TopBraid tools.

On the other hand, we have *TopBraid Live Personal Server* which is included in the Maestro Edition of TopBraid Composer (TBC-ME) in order to replicate Enterprise server features on a local machine within an IDE environment. In this manner, it executes in a single-user environment, providing a powerful tool for designing, developing and testing applications before deploying them to an Enterprise server. You can access the Personal server from any browser, running on the same machine as TBC. Once applications are designed with TBC-ME and the Personal Server, you can deploy them with a click of a button to the Enterprise Server for full Internet and intranet access.

Notice that TopBraid Live provides Federation Services in order to perform federated queries across multiple databases which is an essential feature for our aim.

Fig 38 represents the TopBraid *Enterprise Reference Architecture* (ERA) for the whole TopBraid Suite and its three components (TopBraid Composer, TopBraid Ensemble and TopBraid Live).

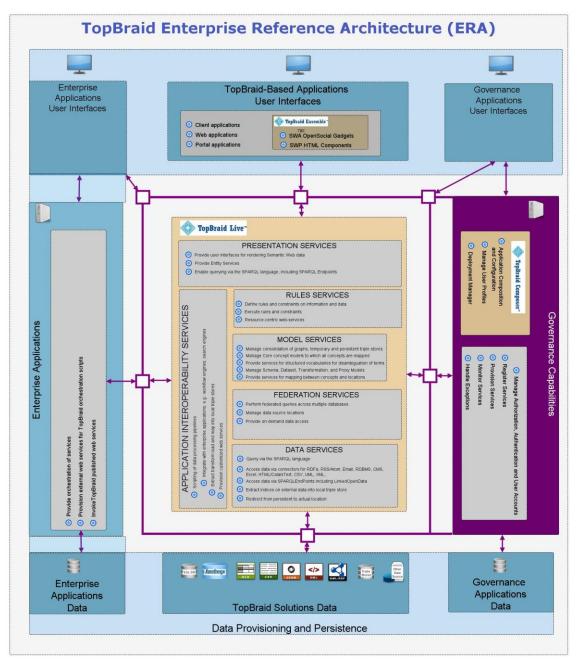


Fig 38 – TopBraid Enterprise Reference Architecture (ERA)

4.4.1 Architectural Knowledge as Linked Data: step by step with TopBraid Suite

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *TopBraid Suite*.

Let's start with TopBraid Composer.

1. First of all, we have to download TopBraid Composer from <u>http://www.topquadrant.com/products/TB install.php</u>. Three versions are available: *Maestro Edition, Standard Edition* and *Free Edition*, from the most comprehensive version of TBC to the lightest release that neither includes support and maintenance, nor the graph view to illustrate the RDF graph, respectively. Apart from the different versions, we can select the specific OS between Windows, Mac and Linux.

2. Once we have downloaded TBC, we have to unzip the corresponding file and execute *TopBraid Composer.exe* to open TBC environment.

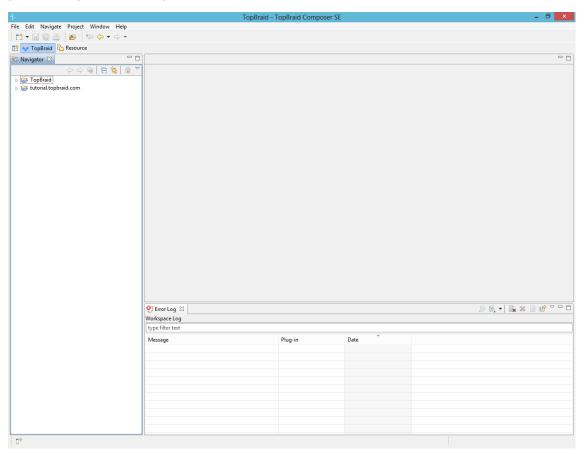


Fig 39 – TopBraid Composer overall interface

3. We can create a new project. Go to File-New-Project.

4			TopBraid -	TopBraid Compo	oser SE	- 8 ×	
File	Edit Navigate Project Window Help						
		Shift+N ▶ [🏫 Project				
	Open File Close Close All Ctrl+	Ctrl+W Shift+W	RDF/OWL File RDF/OWL/SPIN File RDF/OWL Jena SDB Database Connectio	n			
	Save Save As Save All Ctri- Revert Move Rename	+Shift+S			0F/OWL Oracle Database Connection 0F/OWL Sesame2 Repository Connection Ider ample		
۲	Refresh	F5					
ð	Print Switch Workspace Restart	Ctrl+P					
24 24	Import Export						
	Properties A	lt+Enter					
	1 pizza.ttl [TopBraid/Examples] 2 textual_example_LD.rdf [tutorial.to]						
	Exit						
		Error Lo Workspace				, 8 8, - 1 8, x 1 8 7 - 0	
		type filter					
		Message		Plug-in	Date		
				-			
	¢						

Fig 40 – Creating a new project with TBC (1)

Then, we choose *Project* from the General folder and click *Next*.

\$	New Project	- 🗆 ×
Select a wizard Create a new project resource		
Wizards: type filter text		
 a		
Sack	Next > Finish	Cancel

Fig 41 – Creating a new project with TBC (2)

Finally, we give a name to the project and click *Finish* to create it in our workspace.

\$	New Project	- 🗆 🗙
Project Create a new pro	oject resource.	
Project name:		
Location: C:\U	sers\Cris\TBCSEWorkspace\TBC	Browse
Working sets	to working sets	
Working sets:	~	Select
?	< Back Next > Finish	Cancel

Fig 42 – Creating a new project with TBC (3)

4. Once we have already created our TBC project, we may paste our RDF file into it, so we can manage its RDF data.

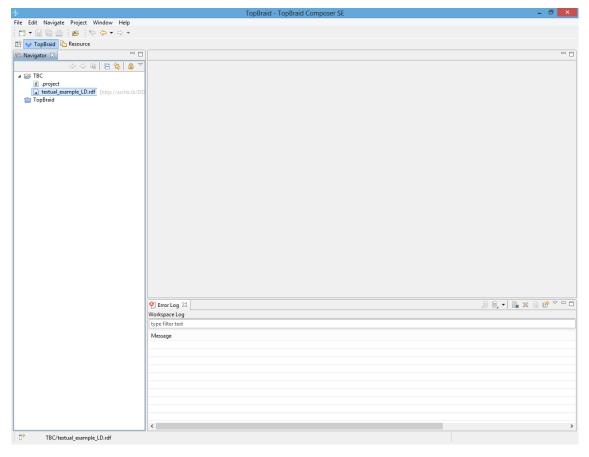


Fig 43 – Adding our RDF data into a TBC project

Then, we are allowed to open this RDF file by double-clicking it, using the editor provided by TBC.

	Imple_D.df Imple_D.Dr.df Imple_D.Dr<].	Image: Properties SS Image: Small State ■ owl:versionInfo ■ officionment
TopBraid Resource Conserse Image: Conserse of the second conserse	mple_LD.rdf 원 Overview ation); [http://archk.tk/DD3		owl:versionInfo
	mple_LD.rdf 원 Overview ation); [http://archk.tk/DD3		owl:versionInfo
Classes ⊠ Image: Classes ∑ Image: Classes ∑ ● ordfs:Resource (175) Ontology ● owk-AllOfferent Base URL (loc ▶ ordf:Property (90) Default Name ● ordf:Statement Namespas ▶ ordf:Class (82) Specify the g	Overview ation): http://archi.tk/DD3		owl:versionInfo
	Overview ation): http://archi.tk/DD3]a	owl:versionInfo
owkIluUnterent owkIluUnterenterent owkIluUnterenterenterenterenterenterenterenter	ation): http://archk.tk/DD3		rdfs:comment
▷ ● rdf:Property (90) Default Namu ● rdf:Statement > ● rdf:Class (82) > ● rdf:Statement > Namespa			
● rdf:Statement ▷ ● rdfs:Class (82)	espace:		rdfs:label rdfs:seeAlso
Specify the p			TUISISEEMISU
Deafin	ce Prefixes prefixes to abbreviate the URIs of the namespaces that are used in this model.		
Pretix	Namespace URI	^ Add	
att	http://archk.tk/dd/attributes#	Delete	
conn	http://archk.tk/dd/relation#		
kind	http://archk.tk/dd/kinds#	Clone	
owl prop	http://www.w3.org/2002/07/owl# http://archk.tk/property#	Goto	
rat	http://archk.tk/dd/attributes/rationale#	~	
	tology annotations View/Edit imported ontologies mespace conflicts Visualize imports in graph		
	The space connects		
Si textual example_LD.rdf [http://archk.tk/DD]			
Overview Pro	ofile Statistics Form Graph Source Code		
🔮 Error Log	🔶 Instances 🔳 Domain 🚦 Relevant Properties 🤸 SPARQL 🧐 Imports 🛙 🔒 🚱	— 🕀 🔍 🗖 🗖 🚼 Basket	X 🛛 🛛 X 🖓 🗖 🗆

Fig 44 – Opening our RDF file with TBC

5. Once we have opened this file in TBC, we can navigate through its different tabs: Overview, Profile, Statistics, Form, Graph or Source Code.

For example, the Form tab shows all properties regarding to a specific resource, in this case <http://archk.tk/DD3>.

4	TopBraid - TBC/textual_example_LD.rdf - TopBraid Composer SE – 🖻					
File	File Edit Navigate Project Model Inference Resource Window Help					
1 5	🝷 🔄 💿 📥 🜔 🝷 🔟 🛕 🛐 🏠 😪 🏠 🖘 🔶 👻 🔩 http://archk.tk/DD	8> 🟠 🤱				
₽Ŷ	TopBraid 🔁 Resource					
	■ textual_example_LD.rdf ⊠		- 4	-		
8	Resource Form		~ '	^ 🛓		
	Name: <http: archk.tk="" dd3=""></http:>		OK			
8	▼ Annotations	▼ Other Properties		8		
€5.	owl:backwardCompatibleWith 🌣	attauthor 🗢		1.16		
	owkincompatibleWith \bigtriangledown	Rafael	~			
		att:category 🗢				
	owl:priorVersion abla	Structural	$\overline{\nabla}$			
	 Incoming References 	att:decision 🗢				
	← conn:constrains ▽	Common Business Logic is Packaged as a Layer	\neg			
		att:rationale 🗢				
		The business logic of the application must be encapsulated in a set of components for both the mobile application and the web server, the common set of components is defined as a separate layer.				
		att:risk 🌣				
		I 0				
		att:scope att:scope				
		Universal				
		att:state 🗢				
		Approved				
		att:timeStamp 🗢				
		Q 2012-05-01T13:55:33.280368+01:00	∇			
		rat:Layer 🗢				
		Business				
		kind:classification				
		Existence				
		conn:constrains				
		(a) http://archk.tk/DD4	$\overline{\nabla}$			
		Image: state of the s	Ť			
		<pre>(http://archk.tk/DD/> () </pre>	Ť			
		connexcludes ▽				
		© (http://archk.tk/DD6>				
		proptype DesignDecision				
		rdf:type ▽				
		ow:Ontology				
		owlimports ~				
	Overview Profile Statistics Form Graph Source Code	owl:versionIRI マ		-		
4 U.		e 🔮 🔶 💻	X	۳ م		

Fig 45 – Form tab in TBC

Another useful component of TBC is the Graph tab (not available in Free Edition of TBC) which shows a graph view of the RDF data, in this case, the AK decision network, including relationships between design decisions.

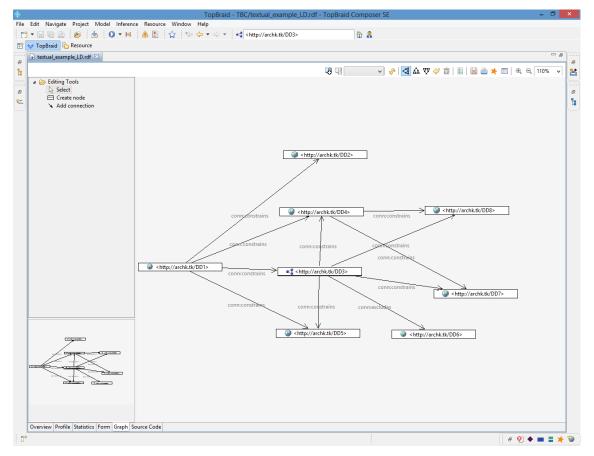


Fig 46 – Graph tab in TBC (1)

We can also expand these nodes in the graphic view in order to show their related information, as you see in Fig 47.

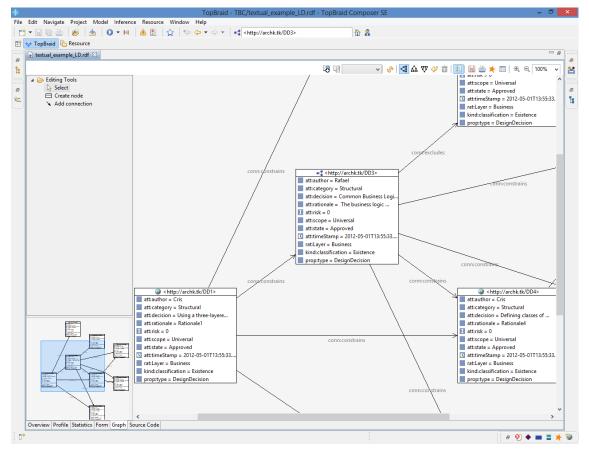


Fig 47 – Graph tab in TBC (2)

6. We may also be interested in execute some SPARQL queries against our RDF data. TBC offers another component which is a SPARQL view that allows us to run this type of queries.

Edit Navigate Project Window Help	- TopBraid Composer SE		
TopBraid Resource			
🕑 Error Log 🔶 Instances 🔳 Domain 🚆 Relevant Properties 🌟 SPARQL 🛛 🦃 Imports			🕒 🕨 🕘 📕 🛛 🛃 🗮 🛣 🗡
Query Editor Query Library	[s]	р	0
SELECT *	A State of the second secon	att:state	Approved
WHERE	<http: archk.tk="" dd1=""></http:>	conn:constrains	<http: archk.tk="" dd5=""></http:>
{ ?s ?p ?o	<http: archk.tk="" dd1=""></http:>	conn:constrains	<http: archk.tk="" dd4=""></http:>
15 (p 10 }	<http: archk.tk="" dd1=""></http:>	prop:type	DesignDecision
,	<http: archk.tk="" dd1=""></http:>	att:category	Structural
	<http: archk.tk="" dd1=""></http:>	att:timeStamp	2012-05-01T13:55:33.280368
	<http: archk.tk="" dd1=""></http:>	conn:constrains	<t <http:="" archk.tk="" dd3=""></t>
	<http: archk.tk="" dd1=""></http:>	att:risk	I 0
	<http: archk.tk="" dd1=""></http:>	conn:constrains	<http: archk.tk="" dd2=""></http:>
	<http: archk.tk="" dd1=""></http:>	att:rationale	Rationale1
	<http: archk.tk="" dd1=""></http:>	kind:classification	Existence
	<http: archk.tk="" dd1=""></http:>	att:scope	Universal
	<http: archk.tk="" dd1=""></http:>	att:author	Cris
	<http: archk.tk="" dd1=""></http:>	att:decision	Using a three-layered archit
	<http: archk.tk="" dd1=""></http:>	rat:Layer	Business
	<http: archk.tk="" dd2=""></http:>	att:timeStamp	2012-05-01T13:55:33.280368
	<http: archk.tk="" dd2=""></http:>	prop:type	DesignDecision
	<http: archk.tk="" dd2=""></http:>	rat:Layer	Business
	<http: archk.tk="" dd2=""></http:>	att:decision	Using a User-Interface layer
	<http: archk.tk="" dd2=""></http:>	att:state	Approved
	<http: archk.tk="" dd2=""></http:>	att:scope	Universal
	<http: archk.tk="" dd2=""></http:>	att:category	Structural
	<http: archk.tk="" dd2=""></http:>	att:author	Cris
	<http: archk.tk="" dd2=""></http:>	att:rationale	Rationale2
	<http: archk.tk="" dd2=""></http:>	att:risk	I 0
	<http: archk.tk="" dd2=""></http:>	kind:classification	Existence
	<http: archk.tk="" dd3=""></http:>	conn:constrains	<http: archk.tk="" dd8=""></http:>
	< <p><<p><<p><<p><<p></p></p></p></p></p>	att:state	Approved
	<pre><q <http:="" archk.tk="" dd3=""></q></pre>	att:decision	Common Business Logic is
	<\$ <http: archk.tk="" dd3=""></http:>	att:rationale	The business logic of the ap
	<1 <http: archk.tk="" dd3=""></http:>	att:timeStamp	2012-05-01T13:55:33.280368
	<1 <http: archk.tk="" dd3=""></http:>	conn:constrains	<http: archk.tk="" dd5=""></http:>
	<1 <http: archk.tk="" dd3=""></http:>	rat:Layer	Business
	< <p><<p><<p><<p><</p></p></p></p>	att:risk	1 0
	< <p><</p>	conn:constrains	<http: archk.tk="" dd7=""></http:>
	<pre><http: archk.tk="" dd3=""></http:></pre>	att:scope	Universal
	<\$ <http: archk.tk="" dd3=""></http:>	kind:classification	Existence
	<\$ <http: archk.tk="" dd3=""></http:>	conn:excludes	<http: archk.tk="" dd6=""></http:>
	<\$ <http: archk.tk="" dd3=""></http:>	att:category	Structural
	<\$ <http: archk.tk="" dd3=""></http:>	rdf:type	owl:Ontology
	<\$ <http: archk.tk="" dd3=""></http:>	att:author	Rafael
5	<	conn:constrains	<http: archk.tk="" dd4=""></http:>
	and chilling (caughd alls (DD2)	0	Desimo

Fig 48 – SPARQL view in TBC

At this time, we have to know that applications developed using TopBraid Composer can be deployed on TopBraid Live whether these applications use a TopBraid Ensemble web-based interface, a custom web-based interface or a programmatic RESTful interface.

Notice that only the Maestro Edition of TopBraid Composer includes the Personal Edition of TopBraid Live so that you can test web-based applications while developing them. When you are ready to deploy these applications for use by multiple users, the Enterprise Edition of TopBraid Live provides a scalable platform for this.

Now, we are going to explain the steps for using *TopBraid Live Personal Server*. Notice that you have to download the TBC Maestro Edition (TBC-ME) in order to use it. We are not going to explain how to use *TopBraid Live Enterprise Server* given that it is not available as free download, but you can contact <u>sales@topquadrant.com</u> for more information and go to <u>http://www.topquadrant.com/docs/tbl/42install/installation.html</u> to follow a complete installation guide.

So, let's continue with TopBraid Live Personal Server within TBC-ME.

- 1. First of all, open TBC-ME by double-clicking its *TopBraid Composer.exe* file.
- 2. Then, ensure that the Personal Server is started. It is started by default but if not, go to the *Help* menu and click on the option "Start Personal TopBraid Server". This option should change as "Stop Personal TopBraid Server" when this server is started.

 At this point, TBC-ME is launching the TopBraid Live Personal Server in background at <u>http://localhost:8083</u>. You can use <u>http://localhost:8083/tbl</u> to access the admin console (see Fig 49).

localhost 8083/tbl/		☆ ≠ C Soogle	P 🗄	: 1	10
	A				
	TopBraid Suite"				
	TopBraid Suite Console version 4.2.1 v20130426-09259	User logi	a		
	TopBraid Live Personal Server				
	TopBraid Live Server Administration				
		Send Error Log to TQ Support	1		

Fig 49 – TopBraid Live Personal Server admin interface

4. Then, we can click on the *TopBraid Live Server Administration* link in order to see the different options offered by this application.

Firefox Y TopBraid Live Personal Server - Server A +			- 0 ×
♦ localhost:8083/tbl/admin/		☆ ♥ C Scoogle	۹ 🗈 🕯 🗖 -
TopBraid Suite Console version			User login
TopBraid Live Per Administrative Function Base URI Manage Server Configuration EVN Session Man Cached Graphs Select Administrat Server Version Int OSCI Bundle Info OSCI Bundle Info OSCI Bundle Info Project Upload Project Upload Project Delete St Memory Manager	ment on Parameters Parameters agement 25 simation mation vices rage password		
		Send Errar Loa to TQ	Banod

Fig 50 – TopBraid Live Personal Server admin options

For example, if we go to the first one, *Base URI Management*, we can see the different projects that we have in our workspace and their own files:

Firefox 🔨 💠 TopBraid Live Personal Serve	r - Base URI +			- d ×
C Collect 8083/tbl/admin/baseURIMgmt			∰ ≠ C 🔀 + Google	P 🖡 🏦 🖸 •
	TopBraid Suite"		User logio	Î
	TopBraid Live Personal S Project - TopBraid Show	ierver — Base URI Management		
	Project - evn.topbraidlive.org			
	FileName	base URI		
	about.html	file://facet.topbraidlive.org/about.html		
	examplePage.html	file://facet.topbraidlive.org/examplePage.html		
	facet#i	http://topbraid.org/facet		
	facet.ui.ttl	http://facet.topbraidlive.org/facet.ui		
	Project - prueba Hide			
	FileName	base URI		
	textual_example_LD.rdf	http://archictlv/DD3		
	Project - sample.evn.topbraidliv	re.org Hide		
	FileName	base URI		
	geo.tch.tdb	um:x-evn-master:geo.tch		
	geo.tdb	um:x-evm-master:geo		
	iptc.lich.tdb	um x-evn-master.iptc.tch		
	(ptc.tdb	um x-evn-master.iptc		
	Project - server.topbraidlive.org	Show		

Fig 51 – Base URI Management (TBL Personal Server)

Another administrative function is *Server Configuration Parameters* that allows you to visualize and edit the current server configuration.

Firefox Thtp://localhost:808la	dmin:ConfigEditor +				٥
localhost:8083/tbl/swp?_viewClass=	tbladmin:ConfigEditor	☆ マ C ^I	م	+	⋒
TopBraid Suite" S	erver Configuration Parameters				
configuration (ServerConfiguration)					
Server Parameters					
Security On:	true				
	http://localhost:8083				
Show Hidden Files:	false				
General Database Parameters					
Cache All Enabled:	faise				
Max Connections per SDB:	1				
wax connections per SDB:	30				
SWP Parameters					
ui label function URI:					
ui link base:					
SMTP Server Parameters					
SMTP Authentication On:	true				
SMTP Server:	100				
SMTP Server Port Number:	587				
SSL Enabled:	false				
SMTP Username:					
SMTP Password:					
Advanced Parameters					
Disable Teamwork Permission Module:	faise				
Enable Fair Locks:	true				
Longer Stack Traces:					
Fail With Error on Design Inconsistency:					
Suppress Warnings of Possible Design					
Inconsistency:					
Max Time per Request:					
Temporary Graphs Time Out: Time before a TDB disk flush:	2 300				
Use Saxon:					
ogging and Tracing Parameters					
Tracing On:	false				
Tracing to Console:	false				
Max Log File Size:	1000				
Max Number of Backups Log.:	10				

Fig 52 – Server Configuration Parameters (TBL Personal Server)

As another example, we may go to *Project Delete* in order to eliminate one or more projects in our workspace. Notice that not all the projects can be removed, as you see in Fig 53.

Firefox * 🔹 💠 TopBraid L	ive Personal Server - Delete Pr +		- 🗇 🗙
e localhost:8083/tbl/adm	in/projectDelete	🏫 ⊽ 😋 🔀 ◄ Google	₽ 🖡 🏠 🖸 י
	TopBraid Suite" JoBRaid Suite Console version 4.2.1 v20130428-0926R	User: <u>Iogin</u>	
	TopBraid Live Personal Server — Delete Project		
	Project - TopBraid Show		
	Project - evn.topbraidlive.org Show		
	Project - facet.topbraidlive.org Show Delete		
	Project - prueba Show Delete		
	Project - sample.evn.topbraidlive.org Show Delete		
	Project - server.topbraidlive.org Show		
		Send Error Log to TQ Support	

Fig 53 – Project Delete (TBL Personal Server)

5. If we want to launch SPARQL queries using this server, we can go to <u>http://localhost:8083/tbl/sparql</u> and execute them, for example: SELECT * WHERE { ?s ?p ?o}. The output of this query is shown in Fig 54. Notice that the result format may be in XML, HTML, Text/CSV and Text/TSV.

	Braid Live SPARQL Endpoint + bl/swp?_viewClass=endpoint:HomePage	☆ マ C 8 • Go	pogle 🔎 🖡	俞
	ve SPARQL Endpoint		· · · · · · · · · · · · · · · · · · ·	
	SELECT * WHERE { ?s ?p ?o }			
	http://archk.tk/DD3	v	.u.	
Result Format	HTML v			
loculto				
s	p	0 Approved		
s ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state	Approved		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains	Approved http://archk.tk/DD5		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains	Approved http://archk.tk/DD5 http://archk.tk/DD4		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/property#type	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/property#type http://archk.tk/property#type	Approved http://archk.tk/DD5 http://archk.tk/DD4		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/attributes#category http://archk.tk /dd/attributes#timeStamp	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision Structural		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/attributes#category http://archk.tk /dd/attributes#timeStamp	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision Structural 2012-05-01T13:55:33.280368+01:00		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/attributes#category http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision Structural 2012-05-01T13.55:33.280368+01.00 http://archk.tk/DD3		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/attributes#fategory http://archk.tk/dd/attributes#fategory http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision Structural 2012-05-01T13:55:33.280368+01:00 http://archk.tk/DD3 0		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/attributes#category http://archk.tk/dd/attributes#category http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision Structural 2012-05-01T13:55:33.280368+01:00 http://archk.tk/DD3 0 http://archk.tk/DD2		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/attributes#category http://archk.tk/dd/attributes#category http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision Structural 2012-05-01T13:55:33.280368+01:00 http://archk.tk/DD3 0 http://archk.tk/DD2 Rationale1		
s ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1 ttp://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/groperty#type http://archk.tk/dd/attributes#category http://archk.tk/dd/attributes#rategory http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/attributes#risk http://archk.tk/dd/attributes#rategorates/ http://archk.tk/ http://ar	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision Structural 2012-05-01T13.55:33.280368+01.00 http://archk.tk/DD3 0 http://archk.tk/DD2 Rationale1 Existence		
ttp://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1 http://archk.tk/DD1	http://archk.tk/dd/attributes#state http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/groperty#type http://archk.tk/dd/attributes#category http://archk.tk/dd/relation#constrains http://archk.tk/dd/relation#constrains http://archk.tk/dd/attributes#risk http://archk.tk/dd/attributes#raionale http://archk.tk/dd/attributes#raionale http://archk.tk/dd/attributes#raionale http://archk.tk/dd/attributes#raionale http://archk.tk/dd/attributes#raionale	Approved http://archk.tk/DD5 http://archk.tk/DD4 DesignDecision Structural 2012-05-01T13:55:33.280368+01:00 http://archk.tk/DD3 0 http://archk.tk/DD2 Rationale1 Existence Universal		

Fig 54 – TopBraid Live SPARQL Endpoint

4.5 Sesame

Sesame [15] is an open source Java framework for storing and querying RDF data, similar to Jena (see section 4.3). This framework is fully extensible and configurable with respect to storage mechanisms, inferencers, RDF file formats, query result formats and query languages.

This framework isn't very useful without implementations of various APIs. The core of the Sesame framework is the *RDF Model API* which defines how the building blocks of RDF (statements, URIs, blank nodes, literals, graphs and models) are represented. Another API is the *Repository API* which is the central access point for Sesame repositories. Its purpose is to give a developer-friendly access point to RDF repositories, offering various methods for querying and updating the data in an easy way.

Therefore, Sesame supports SPARQL querying, a memory-based and a disk-based RDF store and RDF Schema inferencers. It also supports most popular RDF file formats and query result formats.

In addition, we can use the Web application *OpenRDF Workbench* that allows us to interact with Sesame. It is possible given that this Web application provides a web interface for querying, updating and exploring the repositories of a Sesame server.

4.5.1 Architectural Knowledge as Linked Data: step by step with Sesame

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *Sesame*.

In particular, we are going to explain how to use OpenRDF Workbench to manage RDF data.

- First of all, we have to download a Java servlet container, like Tomcat or Jetty, in order to set up Sesame as a standalone server. We choose Jetty as it has a very simple installation and is developed by Eclipse Foundation. Go to http://download.eclipse.org/jetty/stable-9/dist/ and select the latest distribution that will be a ZIP file. At the time of writing, the latest version is 9.0.3.v20130506.
- Next, we have to unzip this file and writing the following line in the console in order to start Jetty Java servlet: *java –jar start.jar*. Then, we can confirm that the installation of Jetty has been successful, going to <u>http://localhost:8080</u> through our browser.

Civ.	C:\Windows\system32\cmd.exe - java -jar start.jar – 🗖 💌
2013-06-11 13:48:58 TION! 2013-06-11 13:48:58 2013-06-11 13:48:58 [file:/E:/jetty-di: 2013-06-11 13:49:00 r found - if your v	ion-9.0.3.v20130506>java —jar start.jar 3.876:WARN::main: test-realm is deployed. DO NOT USE IN PRODUC 3.954:INFO:oejs.Server:main: jetty-9.0.3.v20130506 3.985:INFO:oejdp.ScanningAppProvider:main: Deployment monitor stribution-9.0.3.v20130506/webapps/1 at interval 1 4.471:INFO:oejpw.PlusConfiguration:main: No Transaction manage webapp requires one, please configure one. java:71: No CPUParser for this platform - looking for class: [Windows8]
BIGDATA(R)	
	Flexible Reliable Affordable nputing for the Enterprise
Copyright SYSTAP, I	LLC 2006-2013. All rights reserved.
Oracle Corporation freeMemory=11310453 buildVersion=1.2.3	4 del n, Stepping n, Undeterminable #CPU=1 1.7.0_21 36
Dependency	License
I CU 1	http://source.icu-project.org/repos/icu/icu/trunk/license.htm
bigdata-ganglia	http://www.apache.org/licenses/LICENSE-2.0.html
colt	http://acs.lbl.gov/software/colt/license.html
commons-codec	http://www.apache.org/licenses/LICENSE-2.0.html
commons-fileupload	http://www.apache.org/licenses/LICENSE-2.0.html
commons-io	http://www.apache.org/licenses/LICENSE-2.0.html
commons-logging	http://www.apache.org/licenses/LICENSE-2.0.html
dsiutils	http://www.gnu.org/licenses/lgpl-2.1.html
fastutil	http://www.apache.org/licenses/LICENSE-2.0.html
flot	http://www.opensource.org/licenses/mit-license.php

Fig 55 – Starting Jetty

After that, we have to go back to our prompt and hit *Ctrl-C* to stop Jetty because we haven't included the OpenRDF Workbench WAR files yet, which is the next step.

- Download the OpenRDF Workbench latest release from http://sourceforge.net/projects/sesame/files/Sesame%202/ that will be a ZIP file. At the time of writing, the latest version is 2.7.1. Then unzip this file and copy the archives *openrdf-sesame.war* and *openrdf-workbench.war* in order to paste them into the *webapps* folder of our Jetty installation.
- Then, we have to restart Jetty, writing *java –jar start.jar* in the console, always within our Jetty directory.
- Go to <u>http://localhost:8080/openrdf-workbench/repositories/NONE/repositories</u> to access the OpenRDF Workbench Web application. As you can see in Fig 56, there is a repository by default.

Firefox T OpenRDF V	Workbench - List of Repositori	+			- 8 ×
Calhost:8080/openrdf	-workbench/repositories/NONE/	repositories		☆ ▼ C Scogle	₽ 🖡 🏠 🖸
Workben	ch				OpenRDF
Sesame server	Current Selections:				
Repositories New repository Delete repository	Repository: - none - User (optional): - none -	change			
Explore	List of Re	positori	es		
Summary Namespacies Contexts	🔍 🍞 ld 🛛 Descri	ption	Location http://localhost:8080/openrdf-sesame/repositories/SYSTEM		
Types Explore Query Saved Queries					
Export Modify SPARQL Update					
Add Remove Clear					
System Information					
	Copyright © Aduna 1997 Aduna - Semantic Power	-2011			

Fig 56 – OpenRDF Workbench overall interface

- Then, we can create a repository. Go to Repositories-New repository to create one, give the required information and press *Next* button.

Firefox T OpenRDF Wo	orkbench - New Repository +		- 0 ×
🗲 🕙 localhost:8080/openrdf-v	vorkbench/repositories/NONE/create	☆ ▼ C Roogle	P + ↑ □•
📕 Workbenc	h		OpenRDF ^
Sesame server Repositories New repository Delete repository	Current Selections: Seame server: [chance] Repository: - none - [chance] User (optional): - none - [chance]		
Delete repository Explore Summary Namespaces Contexts Types Explore Query Saved Queries Export Modify SNAKQL Update Add Remove Clear System Information	New Repository Type: Native Java Store Dot Data Title: Linked Data Repository Next Copyright © Aduma 1997-2011 Aduma - Semantic Power	▼ cel	
<			>

Fig 57 – Creating a new repository with OpenRDF Workbench (1)

Then, given the default items, press the Create button.

🗲 🕲 localhost:8080/ope	rdf-workbench/repositories/NONE/create?type=native&id=LData&title=Linked+Data+Repository&next=Next 🔄 🛡 🕑 🔀 👻 Google	₽ 🖡 🏠 🖸
Workbe	nch	OpenRDF
Sesame server	Current Selections:	
Repositories New repository Delete repository	Sesame server: [change Repository: -none: [change User (optional): -none: [change	
Explore Summary	New Repository	
Namespaces Contexts Types	Type: Native Java Store V ID: LData	
Explore	Title: Linked Data Repository	
Query Saved Queries Export	Triple indexes: spoc.posc	
Modify SPARQL Update Add Remove Clear	Create	
System Information		
	Copyright © Aduna 1997-2011 Aduna - Semantic Power	

Fig 58 – Creating a new repository with OpenRDF Workbench (2)

Fig 59 shows the new repository that we have already created within OpenRDF Workbench.

Firefox T OpenRDF W	orkbench - List of Repo	sitori +			- 0 ×
(Iocalhost:8080/openrdf-	workbench/repositories	s/LData/repositories		☆ ▼ C 🛛 🚼 - Google	₽ 🖡 🏠 🖸 ד
Workben d	h				OpenRDF ^
Sesame server Repositories New repository Delete repository	Current Selectio Sesame server: Repository: User (optional):		(change) a) (change) (change)		
Explore	List of	Repositor	ries		
Summary Namespaces	Q 72 Id	Description	Location		
Contexts	y y system	System configuration reposite	nv http://localhost:8080/openrdf-sesame/repositories/SYSTEM		
Types Explore	🖌 🖌 LData	Linked Data Repository	http://localhost:8080/openrdf-sesame/repositories/LData		
Query					
Saved Queries Export					
Modify					
SPARQL Update					
Add Remove					
Clear					
System					
Information					
	Copyright © Adu	1997 2011			
	Aduna - Semant				

Fig 59 - Creating a new repository with OpenRDF Workbench (3)

- We may add RDF data into our repository. Go to Modify-Add, select the RDF file that we want to upload and click *Upload*.

Firefox 🔹 🔲 OpenF	DF Workbench - Add RDF +		- 0 ×
🗲 🛞 localhost:8080/ope	nrdf-workbench/repositories/LData/add	☆ ▼ C [*] S × Google	P 🖡 🏠 🖸
📕 Workbe			OpenRDF
Sesame server	Current Selections: Sesame server: [change]		
Repositories New repository Delete repository	Repository: Linked Data Repository (LData) [change] User (optional): - none - [change]		
Explore Summary	Add RDF		
Namespaces Contexts Types	Base URI: file://textual_example_l ✓ use base URI as context identifier		
Explore Query	Context: <file: td="" textual_example_<=""><td></td><td></td></file:>		
Saved Queries Export	Data format: (autodetect) v Cocation of the RDF data you wish to upload		
Modify SPARQL Update	RDF Data URL:		
Add	 Select the file containing the RDF data you wish to 	upload	
Remove Clear	RDF Data File: E:\Repositorio\ark_knc Examinar		
Information	RDF Content:	, bu	
	Copyright © Aduna 1997-2011 Aduna - Semantic Power		

Fig 60 – Adding RDF data with OpenRDF Workbench

Then, we can go to Explore-Contexts and click on the RDF file that we have already uploaded in order to see its information.

O localhost:8080/oper	nrdf-workbench/repositories/LData	/explore?resource=<	file%3A%2F%2Ftextual_example_LD.rdf> ☆ マ C 8 Google	▶ ♦ ⋒
Workbe	nch			OpenRDF
esame server	Current Selections:			
epositories lew repository)elete repository	User (optional): - none		(change)	
cplore	Explore	(<file: <="" td=""><td>/textual_example_LD.rdf>)(1-99 of 99)</td><td></td></file:>	/textual_example_LD.rdf>)(1-99 of 99)	
ummary lamespaces contexts	Resource: <file: td="" text<=""><td></td><td></td><td></td></file:>			
ypes xplore	Results per page:	100 🗸		
uery	Results offset:	Previ	Dus 100 Next 100	
aved Queries xport	Show data types & lang	uage tags: 🗸		
odify	Subject	Predicate	Object	Context
ARQL Update dd emove	<http: archk.tk="" dd3=""></http:>	att:author	<u>"Rafael"</u>	<file: exa<="" td="" textual=""></file:>
ear	<http: archk.tk="" dd3=""></http:>	att:category	"Structural"	<file: td="" textual_exa<=""></file:>
stem formation	< <u>http://archk.tk/DD3></u>	att:decision	"Common Business Logic is Packaged as a Laver"	
	<http: archk.tk="" dd3=""></http:>	att:risk	۵	<file: td="" textual_exa<=""></file:>
	<http: archk.tk="" dd3=""></http:>	att:scope	"Universal"	<file: td="" textual_exa<=""></file:>
	<http: archk.tk="" dd3=""></http:>	att:state	"Approved"	
	<http: archk.tk="" dd3=""></http:>	att:rationale	The business logic of the application must be encapsulated in a set of components for both the mobile application and the web server, the common set of components is defined as a separate layer.	<file: td="" textual_exa<=""></file:>
	<http: archk.tk="" dd3=""></http:>	rat:Laver	"Business"	<file: td="" textual_exa<=""></file:>
	<http: archk.tk="" dd3=""></http:>	att:timeStamp	2012-05-01T13:55:33.280368+01:00	<file: exa<="" td="" textual=""></file:>
	<http: archk.tk="" dd3=""></http:>	kind:classification	"Existence"	<file: td="" textual_exa<=""></file:>
	<http: archk.tk="" dd3=""></http:>	conn:constrains	<http: archik.tk="" dd4=""></http:>	<file: td="" textual_exa<=""></file:>
	<http: archk.tk="" dd3=""></http:>	conn:constrains	<http: archik.tk="" dd5=""></http:>	<file: td="" textual_exa<=""></file:>
	<http: archk.tk="" dd3=""></http:>	conn:constrains	http://archk.tk/DD72	<file: td="" textual_exa<=""></file:>
	<http: archk.tk="" dd3=""></http:>	conn:constrains	<pre>shttp://archk.tk/DD8></pre>	<file: td="" textual_example:<=""></file:>
	<http: archk.tk="" dd3=""></http:>	conn:excludes	<http: archk.tk="" dd6=""></http:>	<file: examples<="" td="" textual=""></file:>

Fig 61 – Exploring RDF data with OpenRDF Workbench

- Finally, among other things, we can execute SPARQL queries going to Explore-Query and pressing the *Execute* button.

Firefox OpenRDF Worl	kbench - Query Reposi	ory +		- 0 ×
🗲 🕙 localhost:8080/openrdf-wo	orkbench/repositories/	Data/query	☆ マ C Soogle	₽ 🖡 🏠 🖸 י
📕 Workbencl	h			OpenRDF [^]
Sesame server Repositories New repository Delete repository	User (optional): - I	[change] ked Data Repository (LData) [change] one - [change]		
Explore Summary	Query	Repository		
Namespaces Contexts Types	Query Language:	SPAROL V		
Explore		SELECT * WHERE		
Query Saved Queries Export		{ ?8 ?p ?o }		
Modify SPARQL Update Add	Query:			
Remove Clear				
System Information				
			al	
	Results per page:	100 🗸		
	Action Options:	✓ Include inferred statements Save privately (do not share)		
	Actions:	Clear Execute Save query		
	Copyright © Adun Aduna - Semantic			

These are the query results for the previous SPARQL query.

🗧 🛞 localhost:8080/ope	enrdf-workbench/repositories/LData/	query?action=exec&qu	eryLn=SPARQL&query=SELECT * %0AWHERE %0A {%0A %3Fs %3 🏠 🛡 C 🔀 🛪 Google	₽ ♣ ♠
Workbe	ench			OpenRDF
same server	Current Selections:			
epositories lew repository	Sesame server: Repository: Linked E User (optional): - none -		(shange) (shange (shange	
elete repository	Query Re	sult (1-	99 of 99)	
ummary amespaces ontexts	Download format: BIN		Download	
ypes xplore	Results per page:	100 🗸		
Juery aved Queries	Results offset:	Previous	100 Next 100	
Export	Show data types & langu			
Aodify	Show data types a langu	ago tags.		
SPARQL Update	s	Ρ	0	
Add Remove	<http: archk.tk="" dd3=""></http:>	att:author	"Rafael"	
lear	<http: archk.tk="" dd3=""></http:>	att:category	"Structural"	
	<http: archk.tk="" dd3=""></http:>	att:decision	"Common Business Logic is Packaged as a Laver"	
System Information	<http: archk.tk="" dd3=""></http:>	att:risk	<u>0</u>	
mormadon	<http: archk.tk="" dd3=""></http:>	att:scope	"Universal"	
	<http: archk.tk="" dd3=""></http:>	att:state	"Approved"	
	<http: archk.tk="" dd3=""></http:>	att:rationale	The business logic of the application must be encapsulated in a set of components for both the mubile application and the web serv the common set of components is defined as a separate layer.	/er/
	<http: archk.tk="" dd3=""></http:>	rat:Laver	"Business"	
	<http: archk.tk="" dd3=""></http:>	att:timeStamp	2012-05-01T13:55:33.280368+01:00	
	<http: archk.tk="" dd3=""></http:>	kind:classification	"Existence"	
	<http: archk.tk="" dd3=""></http:>	conn:constrains	<http: archk.tk="" dd4=""></http:>	
	<http: archk.tk="" dd3=""></http:>	conn:constrains	<http: archk.tk="" dd5=""></http:>	
	<http: archk.tk="" dd3=""></http:>	conn:constrains	<http: archk.tk="" dd7=""></http:>	
	<http: archk.tk="" dd3=""></http:>	conn:constrains	<http: archk.tk="" dd8=""></http:>	_
	<http: archk.tk="" dd3=""></http:>	conn:excludes	<http: archk.tk="" dd6=""></http:>	_
	<http: archk.tk="" dd3=""></http:>	prop:type	"DesignDecision"	
	<http: archk.tk="" dd4=""></http:>	att:author	"Cris"	_
	<http: archk.tk="" dd4=""></http:>	att:category	"Structural"	_
	<http: archk.tk="" dd4=""></http:>	att:decision	"Defining classes of the mobile application for managing the on-site ticketing process"	_
	<http: archk.tk="" dd4=""></http:>	att:risk	Q	
	<http: archk.tk="" dd4=""></http:>	att:scope	"Universal"	
	<http: archk.tk="" dd4=""></http:>	att:state	"Approved"	
		man to the total of total of the total of		
	<http: archk.tk="" dd4=""></http:>	att:rationale	"Rationales"	
	<http: archk.tk="" dd4=""> <http: archk.tk="" dd4=""></http:></http:>	att:rationale rat:Laver	"Rationale4" "Business"	_

Fig 63 – SPARQL query results with OpenRDF Workbench

4.6 Mulgara

Mulgara [16][17] is a scalable open source RDF database written entirely in Java, licensed under the Open Software License 3.0. This tool is like a relational database due to you can store information and retrieve it via a query language. Unlike a relational database, Mulgara is optimized for the storage and retrieval of RDF statements (subject-predicate-object).

These are the general features of this semantic store:

- Native RDF support
- Multiple databases (models) per server
- Simple SQL-like query language (SPARQL or TQL²)
- Small footprint
- Full text search functionality
- Large storage capacity
- Optimized for metadata storage and retrieval
- Multi-processor support
- Low memory requirements
- Streamed query results

Apart from that, Mulgara provides mechanisms for assuring *reliability* (full transaction support, clustering and store level fail-over, permanent integrity), *connectivity* (Jena, SOAP,

² Text Query Language

Software Developers Kit, etc.), *manageability* (near zero administration, web based configuration and monitoring tools) and *scalability* (XA Triplestore engine) of our system.

Thus XA Triplestore, which is the storage engine of Mulgara, provides scalability due to the following features:

- 64-bit data structures, that allow Mulgara to store very large amounts of data, up to the limits imposed by the host operating system
- Multiple sessions with no lock contention, so a single writing session in addition to multiple reading sessions can access the triplestore concurrently without the reading sessions being required to acquire a global lock while processing a query
- On-line backups, so you can modify and query concurrently with a backup operation
- Permanent integrity, thanks to on-disk data structures of the triplestore
- Use of Java NIO (new I/O), that provides transactions, permanent integrity and good performance while still remaining a pure Java implementation

Mulgara is capable of querying any type of data source by using *Resolvers* which accept and process queries against data contained in a file, a database or other data source. In most cases, the data being queried against is not in a meaningful format, i.e. it is not RDF, and so the resolver must convert the data first.

The resolver database class is highly configurable, allowing you to optimally set up Mulgara for the appropriate usage requirements. By inserting different classes into the constructor of the database class you can set it up as a:

- *Heavyweight store* that uses disk input and output as its primary storage, making it *persistent* across executions of the server
- Lightweight, memory based store that is faster but subject to memory limitations and no persistence

There are five configurable parts for the database with respect to its operation:

- 1. Persistent Node Pool
- 2. Persistent String Pool
- 3. Temporary Node Pool
- 4. Temporary String Pool
- 5. System Resolver Factory

The *Persistent String* and *Node Pools* maintain the mappings of node *id* to string representations for all current models in both the System and External models. The *Temporary String* and *Node Pools* are used for storing temporary nodes during a query.

4.6.1 Architectural Knowledge as Linked Data: step by step with Mulgara

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *Mulgara*.

1. First of all, go to <u>http://www.mulgara.org/download.html</u> and download the latest version of Mulgara. At the time of writing, the latest release is 2.1.13. Notice that we are going to download the ZIP file.

2. Then, unzip this file in order to start a Mulgara server. Go to a console terminal and type *java -jar mulgara-2.1.13.jar* to start a Mulgara server with default settings.

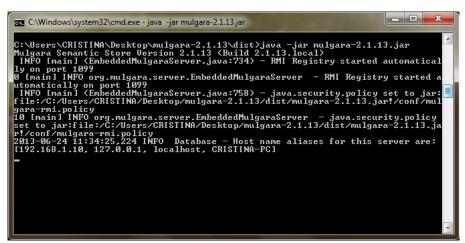
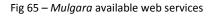


Fig 64 – Starting a Mulgara server

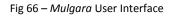
3. A default configuration for a standalone Mulgara server runs a set of web services, including the Web User Interface. So, we can go to http://localhost:8080/ and show the list of services available, as you see in Fig 65.

Hirefox Mulgara Web Services A O localhost 8080	+	•••	rar C 8 - Google	A 1 1
Available Services				
Web Services User Interface TQL HTTP Service				
User Tutorial SPARQL HTTP Service				



4. If we click on the *User Interface* link (<u>http://localhost:8080/webui/</u>), a page should appear as below:

firefox MulgarsViewer > ③ localhost8080/webul/	+		☆ マ C Socopie	P (1)
	SEMANTIC ST	DRE		
Graph URI:				
Query Text:				
File:	Submit Query Clear Query	Examinar. Upload		



This Web UI page is used for general access to a Mulgara database.

5. Then, we can upload a RDF file in order to manage it later. First, we have to specify a graph URI, i.e. any valid absolute URI, for example: *urn:mydata*. Then, press the *Examinar...* button, choose the RDF file and click *Upload*.

Firefox * Maldgara Viewer	+	•		ring ⊂ C S + Google	
	Ulgara SEM	ANTIC STORE			
Graph URI:	um:mydata				
Query Text:					
	Submit Query Clear Query				
File:	G:\Universidad\Doctorado\R	apositorio\Líneas de trabajo\op	en (Examinar. Upload		

Fig 67 – Uploading a RDF file into *Mulgara* server (1)

It should appear a success message like this:

Firefox *	i initigara viewer	+				
+ >	localhost:8080/webui/ExecuteQu	ery			☆ マ C 🚼 - Goo	gle 🤌 🖓 🖻
	m 🍲	JIgara" SEMA	NTIC STORE			
	Graph URI:	um:mydata				
	Query Text:					
		Submit Query Clear Query				
	File:			Examinar_ Upload		
	Results:	(1 query, 0,720 seconds)				
	Query Execu		irtual/textual_example_LD.rdf> int			
	Result Messa	ge: Successfully loaded 99 stat	ements from http://mulgara.org/v	rtual/textual_example_LD.rdf into	o urn:mydata	

Fig 68 - Uploading a RDF file into *Mulgara* server (2)

6. Once we have uploaded our data, we can query it using, in this case, the SPARQL language. For example, we can write *SELECT* * *WHERE* {?s ?p ?o} in the *Query Text* area, which contains a single command or query in either TQL or SPARQL languages, and click on *Submit Query* button. Fig 69 shows the results for this SPARQL query.

		SEMANTIC STORE	
Graph URI:	um:my	data	
	SELEC WHERE { ?		
Query Text:			
	Subm	it Query Clear Query	
File:			Examinar Upload
Results:		(1 query, 0,000 seconds) SELECT * WHERE (?s ?p ?o)	Examinar) Upload
Results: Query Exe	cuted:	(1 query, 0,000 seconds) SELECT * WHERE [?s ?p ?o] P	•
Results: Query Exe s http://arch	cuted:	(1 query, 0,000 seconds) SELECT * WHERE (3: 10: 10: 10 http://archik.tk/dd/attributes#author	2draet
Results: Query Exe http://arch http://arch	cuted:	(1 query, 0,000 seconds) SELECT * WHERE (1s to to) P http://archk.tk/dd/attributes#outhor http://archk.tk/dd/attributes#category	۰ Tafaer 'کtructurat'
Results: Query Exe http://arch http://arch http://arch	cuted: :: 1k.tk/DD3 1k.tk/DD3 1k.tk/DD3	(1 query, 0,000 seconds) SELECT * WHERE [15 10 to] http://archk.tk/dd/attributes#outhor http://archk.tk/dd/attributes#outhor http://archk.tk/dd/attributes#outhor	• "Earlace" • "Structural"
Results: Query Exec s http://arch http://arch http://arch	cuted: : ik.tk/D03 ik.tk/D03 ik.tk/D03 ik.tk/D03	(1 query, 0,000 seconds) SELECT * WHERE [5 Tp To] P P http://archk.tk/dd/attributes#author http://archk.tk/dd/attributes#decision http://archk.tk/dd/attributes#fisk	Cafact Structural Someon Business Logic is Packaged as a Layer Someon Business Logic is Packaged as a Layer
Results: Query Exer http://arch http://arch http://arch http://arch	cuted: 1	(1 query, 0,000 seconds) SELECT * WHERE [15 bp ?o] P http://archa.tk/dd/attributes#author http://archa.tk/dd/attributes#cetegory http://archa.tk/dd/attributes#risk http://archa.tk/dd/attributes#risk http://archa.tk/dd/attributes#risk	© <u>"Rafaef"</u> <u>"Structural"</u> <u>"Common Business Logic is Packaged as a Layer"</u>
Results: Query Exe http://arch http://arch http://arch http://arch http://arch	cuted: 3 ik. tk/DD3 ik. tk/DD3 ik. tk/DD3 ik. tk/DD3 ik. tk/DD3 ik. tk/DD3	(1 query, 0,000 seconds) SELECT * WHERE [5 Tp To] P P http://archk.tk/dd/attributes#author http://archk.tk/dd/attributes#decision http://archk.tk/dd/attributes#fisk	Caraction Common Business Logic is Packaged as a Layer* Common Business Logic is Packaged as a Layer* T***http://www.wi.org/2001/XMLSchemafiinteger. Turkersaf Zéparoned
Results: Query Exer http://arch http://arch http://arch http://arch	cuted: 3 ik. tk/DD3 ik. tk/DD3 ik. tk/DD3 ik. tk/DD3 ik. tk/DD3 ik. tk/DD3	(1 query, 0,000 seconds) SELECT * WHERE [15 bp ?o] P http://archa.tk/dd/attributes#author http://archa.tk/dd/attributes#cetegory http://archa.tk/dd/attributes#risk http://archa.tk/dd/attributes#risk http://archa.tk/dd/attributes#risk	O Tafaet Siructurat Common Busines Logic is Packaged as a Layer 'Common Busines Logic is Packaged as a Layer 'Lonkersat Toriestat
Results: Query Exe http://arch http://arch http://arch http://arch http://arch	cuted: : ik. tk/D03 ik. tk/D03 ik. tk/D03 ik. tk/D03 ik. tk/D03 ik. tk/D03 ik. tk/D03	(1 query, 0,000 seconds) SELECT * WHERE [Is Tp To] P http://archk.tk/dd/attributes#author http://archk.tk/dd/attributes#category http://archk.tk/dd/attributes#csion http://archk.tk/dd/attributes#cope http://archk.tk/dd/attributes#state	Common Business Logic is Packaged as a Layer" Common Business Logic is Packaged as a Layer" Vorwerat Transport Common Business Logic of the application and the web server; The business logic of the application must be encapsulated in a set of components for both the mobile application and the web server; The business logic of the application must be encapsulated in a set of components for both the mobile application and the web server;
Results: Query Exe http://arch http://arch http://arch http://arch http://arch http://arch	cuted: 3 ak.tk/DD3	(1 query, 0,000 seconds) SELECT * WHERE [Is Ip To] P http://archk.tk/dd/attributes#author http://archk.tk/dd/attributes#decision http://archk.tk/dd/attributes#decision http://archk.tk/dd/attributes#stel	Common Business Logic is Packaged as a Layer' Common Business Logic is Packaged as a Layer' Vorw.w3.org/2001/3MLSchema#integer: Universat Zeproved The business logic of the application must be encapsulated in a set of components for both the mobile application and the web server;

Fig 69 – Querying RDF data with Mulgara

4.7 RedStore

RedStore [18] is a lightweight RDF triplestore written in C language, using the Redland library that is a set of free software libraries that provide support for RDF. It supports, in addition to its native persistent or in-memory storage, a variety of storage backend adapters, including MySQL, Postgres, and Virtuoso. In native mode, RedStore uses *hash tables* for persisting RDF data.

This tool has some main features, cited as below:

- SPARQL over HTTP support
- Built-in HTTP server

- Support for a wide range of RDF formats
- Only one runtime dependency: Redland
- Unit and integration test suite

But it has some limitations too:

- Single process/single threaded
- No request timeouts

There are some available *storage modules*³ for RedStore that you can explicit while starting it, using the *-t options* directive:

- Hashes that provides in-memory or persistent storage via Sleepycat/Berkeley DB (BDB) -> -t hash-type='bdb'
- Memory that provides a simple and fast in-memory store with no persistence
- File that provides an in-memory model initialised from the RDF/XML content in a file
- *MySQL* that is compiled in when MySQL 3 or 4 is available and provides persistent storage using the MySQL open source database
- *PostgreSQL* that is based on MySQL store and is compiled in when PostgreSQL is available. This store provides persistent storage using the PostgreSQL open source database
- *SQLite* that provides persistent storage via the SQLite relational database when available and supports SQLite V2 and V3
- Tstore that provides persistent storage via the AKT Triplestore when available
- URI that provides an in-memory model initialised from the RDF/XML content in a URI
- Virtuoso that is capable of efficiently handle large amounts of data with persistence

So, as an example, if you want to start RedStore on port 8080 using a persistent hashes store, you have to type this command: *redstore -p 8080 -s hashes -t "hash-type='bdb"*. Notice that you have to install a Berkeley DB in order to use the hash-type *bdb* (go to http://linux.softpedia.com/progDownload/Berkeley-DB-Download-75.html).

4.7.1 Architectural Knowledge as Linked Data: step by step with RedStore

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *RedStore*.

 First of all, notice that RedStore is not available for Windows environments, but for Mac OS X and Linux. In this case, I'm going to use a Linux environment like Ubuntu in order to try out RedStore. If we don't have a machine with Ubuntu as the Operating System, we can download *VirtualBox* from <u>https://www.virtualbox.org/wiki/Downloads</u> for emulating a Linux Operating System.

Notice that we can use, for example, *Cygwin* for simulating a Linux bash terminal, instead of a virtual machine environment, but it is not recommended because, in this case, Cygwin doesn't work properly during the RedStore installation.

³ <u>http://librdf.org/docs/api/redland-storage-modules.html</u>

- Then, go to <u>http://releases.ubuntu.com/</u> in order to download an Ubuntu disk image. At the time of writing, the latest release is Ubuntu 13.04. This image has to be set within VirtualBox, so we can follow this tutorial <u>http://www.psychocats.net/ubuntu/virtualbox</u> for install it.
- Once we have installed Ubuntu inside VirtualBox, start this virtual machine and open a terminal. Then, we are going to follow this tutorial in order to install RedStore <u>http://wiki.filteredpush.org/wiki/RedStore_Installation_on_firuta</u>.
- 4. Within the Linux terminal, write the following lines in order to download RedStore and its dependencies into a *tmp* folder:

```
cd Desktop
mkdir tmp
cd tmp
wget http://download.librdf.org/source/raptor2-2.0.6.tar.gz
wget http://download.librdf.org/source/rasqal-0.9.28.tar.gz
wget http://download.librdf.org/source/redland-1.0.15.tar.gz
wget http://www.aelius.com/njh/redstore/redstore-0.5.4.tar.gz
Then, unpack these archives:
tar -zxvf raptor2-2.0.6.tar.gz
tar -zxvf rasqal-0.9.28.tar.gz
tar -zxvf redland-1.0.15.tar.gz
tar -zxvf redstore-0.5.4.tar.gz
```

5. Next, we have to install the dependencies for the Autogen script:

```
sudo su
apt-get install automake
apt-get install libtool
apt-get install gtk-doc-tools
```

6. Now, within the Raptor directory, run the Autogen script, install Libxml, which is the XML parser to be used by Raptor, compile and install it:

```
cd raptor2-2.0.6
./autogen.sh
apt-get install libxml2-dev
./configure
make
make install
```

7. Next, we must compile Rasqal, but first we need the latest version of Flex and Bison:

```
apt-get install flex apt-get install bison
```

8. Now we can run the Autogen script followed by configure, make and install for Rasqal:

```
cd ..
cd rasqal-0.9.28
./autogen.sh
./configure
make
make install
```

9. Now for Redland:

```
cd ..
cd redland-1.0.15
./autogen.sh
./configure
make
make install
```

10. Finally, we can install RedStore given that its dependencies are taken care of:

```
cd redstore-0.5.4
./configure
make
make install
ldconfig
```

11. Once we have installed RedStore, we can start it on port 8086:

redstore -p 8086 -b localhost

If we go to <u>http://localhost:8086/</u>, we can see the main page of RedStore as in Fig 70.

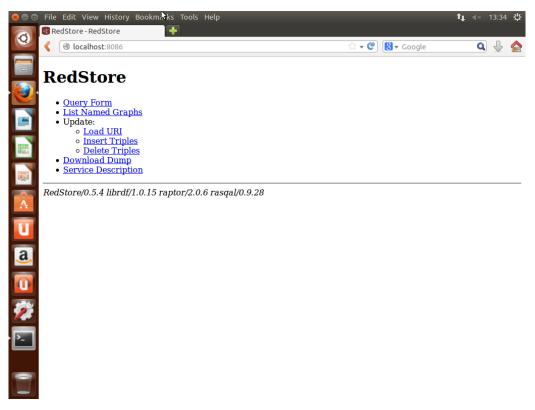


Fig 70 – Main page of RedStore

12. Then, if we go to *Insert Triples*, we can update our dataspace with our RDF data, selecting RDF/XML as the triple syntax, indicating a base URI and pressing the *Submit Query* button, as you see in Fig 71 and Fig 72.

8 🖨 🗇	File Edit View History Bookmarks Tools Help			ी∎ ≪ 14:06 ∰
Q	Kostore-insert inpres Sociality (insert)	ೆ • 🥙	<mark>8</mark> ▼ Google	۹ 🖟
	Insert Triples			
	<pre><?Xml.version="1.0" encoding="utf-8"?> <!DOCTYPE rdf:BDCI <entity 'http:="" 02="" 1999="" 22-rdf-syntax-ns#'="" rdf="" www.w3.org=""> <!--ENTITY rdf 'http://www.w3.org/2000/01/rdf-schema#'--> <!--ENTITY rdf 'http://www.w3.org/2001/XMLSchema#'--> <!--ENTITY rsf 'http://archk.tk/rdf/kinds#'--> <!--ENTITY rdf 'http://archk.tk/dd/kinds#'--> <!--ENTITY kind 'http://archk.tk/dd/attributes#'--> <!--ENTITY rat 'http://archk.tk/dd/attributes/rationale#'--> </entity> <</pre>			
1 1 1 1	<pre>-crdf:BDE xml:base="http://archk.tk/DD3" xmlps:cdfs="http://www.w3.grg/2000/01 //cdf:schema#" xmlps:xsd="http://www.w3.grg/2001/XHLSchema#" xmlus:gcom="http://archk.tk/dd/relation#" xmlps:arb="http://archk.tk/ /property#" xmlps:kind="http://archk.tk/dd/kinds#" xmlps:att="http://archk.tk/ dd/attributes#" xmlps:cdf="http://archk".tk/dd/kinds#" xmlps:att="http://archk.tk/ xmlps:crat="http://archk.tk/dd/attributes/rationale#">- <rdf:description gdf:about="http://archk.tk/dd/attributes/rationale#">- <rdf:description gdf:about="http://archk.tk/dd/attributes/rationale#">-</rdf:description></rdf:description></pre>			
<u>a</u> ,	Triples syntax: RDF/XML ‡ Graph: (optional)			
	Base URI: http://archk.tk/DD3 (optional)	al)		
	Reset Submit Query			
<u>□</u> 22	RedStore/0.5.4 librdf/1.0.15 raptor/2.0.6 rasqal/0.9.28			
	Fig 71 — Inserting triples with Red File Edit View History Bookmarks Tools Help RedStore-OK	Store (I)		1 ∎ ≪ 14:08 -∰
	File Edit View History Bookmarks Tools Help		🞗 🕶 Google	tµ ∢× 14:08 & Q ⊕ 🏠
	File Edit View History Bookmarks Tools Help		<mark>8</mark> ▼ Google	
	File Edit View History Bookmarks Tools Help Image: Constraint of the second s		8 ▼ Google	
	File Edit View History Bookmarks Tools Help RedStore-OK Cocalhost:8086/insert CK		<mark>≳</mark> ▼ Google	
	File Edit View History Bookmarks Tools Help RedStore - OK Cocalhost: 8086/insert OK Successfully added triples to: the default graph.		<mark>8</mark> ▼ Google	
	File Edit View History Bookmarks Tools Help RedStore - OK Cocalhost: 8086/insert OK Successfully added triples to: the default graph.		<mark>8</mark> ▼ Google	
	File Edit View History Bookmarks Tools Help RedStore - OK Cocalhost: 8086/insert OK Successfully added triples to: the default graph.		<mark>≷</mark> ▼ Google	
	File Edit View History Bookmarks Tools Help RedStore - OK Cocalhost: 8086/insert OK Successfully added triples to: the default graph.		<mark>≷</mark> ▼ Google	
	File Edit View History Bookmarks Tools Help RedStore - OK Cocalhost: 8086/insert OK Successfully added triples to: the default graph.		▼ Google	
	File Edit View History Bookmarks Tools Help RedStore - OK Cocalhost: 8086/insert OK Successfully added triples to: the default graph.		<mark>≷</mark> ▼ Google	

Fig 72 – Inserting triples with RedStore (II)

13. Finally, if we go back to the main menu, we can choose the *Query Form* option and make a SPARQL query, as you see in Fig 73 and Fig 74.

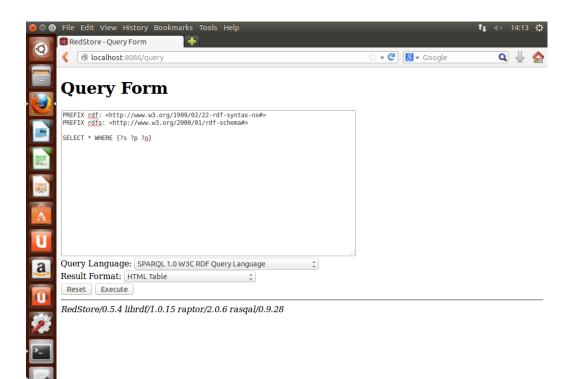


Fig 73 – Making a SPARQL query within RedStore (I)

2 🗆 🖻	File Edit View Histor	y Bookmarks Tools Help	tų ∢× 1	14:14	ψ
0	SPARQL Query Results	+			
	localhost:8086/	query?query=PREFIX+rdf%3A+ <ht< th=""><th>ttp%3A%2F%2Fwww.w3.org%2 ☆ ♥ 🕲 ♥ Google 🔍</th><th>\mathbb{Q}</th><th></th></ht<>	ttp%3A%2F%2Fwww.w3.org%2 ☆ ♥ 🕲 ♥ Google 🔍	\mathbb{Q}	
	?s	?p	?0] f
	http://archk.tk/DD1	<u>http://archk.tk</u> /dd/attributes#category	Structural		
	http://archk.tk/DD2	<u>http://archk.tk</u> /dd/attributes#timeStamp	2012-05-01T12:55:33.280368Z^^ <http: www.w3.org<br="">/2001/XMLSchema#dateTime></http:>		
	http://archk.tk/DD4	<u>http://archk.tk</u> /property#type	DesignDecision		
	http://archk.tk/DD6	<u>http://archk.tk</u> /dd/attributes#category	Structural		
F	http://archk.tk/DD5	<u>http://archk.tk</u> /dd/kinds#classification	Existence		
	http://archk.tk/DD5	<u>http://archk.tk</u> /dd/attributes#rationale	Rationale5		
	http://archk.tk/DD3	<u>http://archk.tk</u> /dd/relation#excludes	http://archk.tk/DD6		
	http://archk.tk/DD2	<u>http://archk.tk</u> /dd/kinds#classification	Existence		
<u>a</u> ,	http://archk.tk/DD7	<u>http://archk.tk</u> /dd/attributes#rationale	Rationale7		
1	http://archk.tk/DD4	<u>http://archk.tk</u> /dd/attributes#category	Structural		
2	http://archk.tk/DD8	<u>http://archk.tk</u> /dd/attributes#state	Approved		
▶_	http://archk.tk/DD8	<u>http://archk.tk</u> /dd/attributes#scope	Universal		
	http://archk.tk/DD6	<u>http://archk.tk</u> /dd/attributes#rationale	Rationale6		
	http://archk.tk/DD7	<u>http://archk.tk</u> /dd/attributes /rationale#Layer	Business		
		A			71 V

Fig 74 - Making a SPARQL query within RedStore (II)

4.8 Callimachus

Callimachus [19] is a framework for data-driven applications based on Linked Data. It allows web developers to quickly and easily create web applications based on Linked Data given that they only need a web browser to create a data-driven application.

In addition, Callimachus builds on either Sesame (section 4.5) or Mulgara (section 4.6) for RDF storage, AliBaba (a RESTful object-RDF library) and uses a revolutionary template-byexample technique for viewing and editing resources. One of the interesting aspects of Callimachus is that templates are parsed to build SPARQL from RDFa markup and then filled with query results.

4.8.1 Architectural Knowledge as Linked Data: step by step with Callimachus

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *Callimachus*.

- First of all, we have to make sure that it is installed a JDK and JRE in our machine, and that the version is correct. In this case, we need JDK 1.6 or 1.7. We can type java –version in the command prompt in order to see what version is locally installed. Go to <u>http://www.oracle.com/technetwork/es/java/javase/downloads/index.html</u> for downloading them.
- 2. Once we have installed JDK and JRE in our computer, we have to set the environment variables for JDK_HOME (pointing to our JDK directory), JAVA_HOME (pointing to JDK our directory) and JRE_HOME (pointing to our JRE directory).
- 3. Next step is to download Callimachus. Go to http://callimachusproject.org/get-started.xhtml?view and download the latest version which is nowadays 1.1.2. It is a ZIP file that you have to unzip into an empty target directory.
- 4. Then, copy the file *etc/callimachus-defaults.conf* to a new file called *etc/callimachus.conf*, and open it in order to make some necessary changes, like remove the '#' before the PORT and ORIGIN variables. We can modify the values of these variables for make them more appropriate to our system.
- 5. Create a file called *etc/mail.properties* to allow for mail functionality within Callimachus.
- 6. Then, from a console, initialize Callimachus typing *callimachus-setup.bat* within the *bin* folder. After a few seconds, a new window is opened in our browser, as you can see in Fig 75. Fill in the information required and press "Yes, sign me up!" button.

Firefox Y Jigest account Sign Up +		- 0 ×
🕘 🕙 localhost:8080/auth/digest+account?register&token=7f33fc0753a03a51047219cde9d58bb3&nonce=862e8305&email=Cris%40	Cris-PC&return_to=⊦ 🏫 ⊽ C 🛛 🗧 Google	۹ 🖡 د
Callimachus		Sign in
localhost:8080 » auth » Digest account		
Digest account Sign Up		
localhost		
Sign in with your email address and a site password		
To accept this invitation please confirm the information belove and provide a password.		ne is used to indentify n this website
Full name		address is used to
cris	manage you shown on th	ir password and is not ie website
Email address		me is used to uniquely
tinarodasanchez@gmail.com	identify your website	account on this
Username		ble check your email
cris	address is c	orrect before signing up
New password		
•••••		
Good		
Yes, sign me up! I'll come back later		
This resource was last modified at 11:22		Callimachus

Fig 75 – Assigning a password to my Callimachus account

7. Then, a new sign-in window is showed. We have to write our credentials and press the "Sign in" button in order to access the Callimachus client application. Fig 77 shows the main interface, once we have signed in.

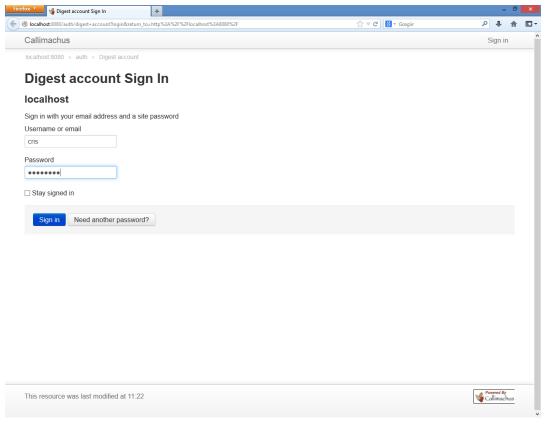


Fig 76 – Signing in Callimachus

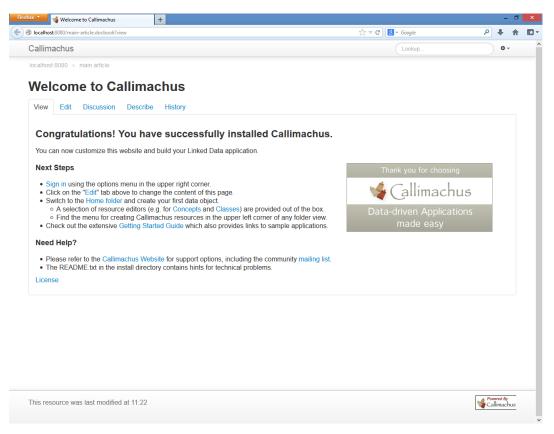


Fig 77 – Main interface in Callimachus

8. Now, we can import linked data into Callimachus. Navigate to the Home Folder link and create a new folder for our data, pressing the "Create" button.

localhost:8080/?view						C 🔄 - Go	ogie	P 🖡 👘
	Callimachus				Lookup		••	
	View Edit Discussion Describe History							
	Create O	localhost:8080						
	File	Updated	Readers	Subscribers	Contributors	Editors	Administrators	
	well known	11:22	public	power			admin	
	auth 🖿	11:22	public	everyone			admin	
	Calimachus	11.22	public				super	
	4 describe	11:22	power				admin	
	# favicon	11:22	public	users		power staff	admin	
	🗑 main article	11:22	public	users		power staff	admin	
	E robots	11:22	public	users		power staff	admin	
	<1 sparql	11:22	power				admin	
	Showing results 8 of 8							
							Collinachus	

Fig 78 – Managing linked data into Callimachus (I)

Choose the *Folder* option.

					合文	C 🛃 - Ga	ogle	P 🖡 🏦
Callimachus					Lookup_		•-	
View Edit Discussion	n Describe History							
Create 🔘		localhost:8080						
Article	File	Updated	Readers	Subscribers	Contributors	Editors	Administrators	
Concept		11:22	public	power			admin	
PURL		11:22	public	everyone			admin	
WORL		11:22	public				super	
More options +		11:22	power				admin	
W favic on		11:22	public	users		power staff	admin	
🛞 main article		11.22	public	users		power staff	admin	
i robots		11:22	public	users		power staff	admin	
-4 sparql		11.22	power				admin	
Showing results 8 of 8								

Fig 79 - Managing linked data into Callimachus (II)

Give a folder name and press the "Create" button.

Firefox 🛀 🍕 New Folder 🔶		- 0 ×
🗲 🛞 localhost.8080/?create=http%3A%2F%2F1ocalhost%3A8080%2Fcallimachus%2F1.0%2Ftypes%2FFolder	☆ マ C 00000	P 🖡 🕆 🖸•
Callimachus	Lookup O -	^
New Folder		
Name LData	Choose a recognisable label to be	
LData	used in the URL	
Create Cancel		
This resource was last modified at 11:22	Powerd By Calinachus	

Fig 80 - Managing linked data into Callimachus (III)

Then, click on the Upload option in order to upload the linked data file.

O localhost 8080/LData/Niew					台王	C 🛛 - Go	ogle	ه م
	Callimachus				Lookup		0+	
	tocathost 2000 + LData View Edit Discussion Describe History Crease 0	locathost:8000 / LD	ata					
	Upload file File Showing results 0 of 0	Updated		Subscribers	Contributors	Editors	Administrators	

This resource was last modified at 13.20	Collimachus
localhost:8080/LData/?create=http://localhost:8080/callimachus/10/types/File	

Fig 81 - Managing linked data into Callimachus (IV)

Then, select the corresponding RDF file and click "Upload".

History 🔧 🧃 Upload File	+				-	- 0	×
🔄 🕲 localhost 8080/LData/7create=http%3A%2F%2Flocalh	ost%3A8080%2Fcellimachus%2F1.0%2Ptypes%2FFile		🟫 🔻 C 🔣 * Google		₽.	Ĥ	1
Callimachu	IS		Lookop.	0-			î
localhost 8080	> LData						
Upload	- File						
	a File						
File	textual_example_LD rdf						
Examinar	textual_example_LU.rdr						
Upload	í l						
This resource	was last modified at 13:20		-	Callimachus			
				•			

Fig 82 - Managing linked data into Callimachus (V)

Now, our RDF data is on <u>http://localhost:8080/LData/textual_example_ld.rdf</u>, as you can see in Fig 83.

Firefox * 🙀 testual example id 🛛 × 🙀 http://localhost.00ual_example_id.nff × 🕂		- 8 ×
C C C C C C C C C C C C C C C C C C C	🚖 🗢 🥲 🔣 - Google	P 🖡 🏦 🖬 -
Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.		
- <rdf:rdf xml:base="http://archk.tk/DD3"></rdf:rdf>		
- <rdf:description rdf:about="http://archk.tk/DD3"></rdf:description>		
<att:author>Rafael</att:author>		
<att:category>Structural</att:category>		
<att:decision>Common Business Logic is Packaged as a Layer</att:decision>		
<att:risk rdf:datatype="http://www.w3.org/2001/XMI.Schema@integer">0</att:risk>		
<att:scope>Universal</att:scope>		
<att:state>Approved</att:state>		
- <att:rationale></att:rationale>		
The business logic of the application must be encapsulated in a set of components for both the mobile application and the web server; the common set of c	components is defined as a separate layer.	
<rat:layer>Business</rat:layer>		
<att:timestamp rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2012-05-01T13:55:33.280368+01:00</att:timestamp>		
<kind:classification>Existence</kind:classification>		
<conn:constrains rdf:resource="http://archk.tk/DD4"></conn:constrains>		
<conn:constrains rdf:resource="http://archk.tk/DD5"></conn:constrains>		
<conn:constrains rdf:resource="http://archk.tk/DD7"></conn:constrains>		
<conn:constrains archk.tk="" dd6"="" http:="" rdf:resource="http://archk.tk/DD\$*/></td><td></td><td></td></tr><tr><td><conn:excludes rdf:resource="></conn:constrains>		
<pre>sprop:type>DesignDecision</pre>		
= <rdf:description rdf:about="http://archk.tk/DD1"></rdf:description>		
<att:author>Cris</att:author>		
<att:category>Structural</att:category>		
<att:decision>Using a three-layered architecture</att:decision>		
<att:risk 2001="" http:="" rdf:datatype="http://www.w3.org/2001/XMLSchema@integer>0</att:risk></td><td></td><td></td></tr><tr><td><atts:cope>Universal</atts:cope>
<atts:tate>Approved</atts:tate></td><td></td><td></td></tr><tr><td><attstate>Approved<attstate> <attstate></td><td></td><td></td></tr><tr><td> attrationale-randomic/attrationale/ attrationale/attrationale/attrationale/ </td><td></td><td></td></tr><tr><td><pre><rail.asymr>business<rail.asymr>
<atiinesitamp rdf.datatype=" www.w3.org="" xmlschema@datetime"="">2012-05-01T13:55:33.280368+01:00</att:risk>		
<artimestamp att:tumestamp="" runcatatype="mp./www.org.2007.581.5028mawaate1me=22012-05-07115-55.55.280508-01.004"> <kind:classificatios exitences="" kind:classificatios=""></kind:classificatios></artimestamp>		
<amultitassintations archi.tk="" dd3="" http:="" refi:resource="http://archk.tk/DD2'/></td><td></td><td></td></tr><tr><td><comm constraints rdfresource="></amultitassintations> <comm.constraints archk.tk="" dd4="" http:="" rdfresource="http://archi.tk/DD3/></td><td></td><td></td></tr><tr><td><cons:constrains rdf:resources"></comm.constraints> >		
<com archk.tk="" constrains="" dd2"="" http:="" rdf:resource="http://archk.tk/DD5/></td><td></td><td></td></tr><tr><td><pre><pre>contentstation</pre>type></pre></td><td></td><td></td></tr><tr><td></rd></td><td></td><td></td></tr><tr><td>-<rdf:Description rdf:about="></com>		
<att author="">Cris</att>		
attractions.com		
<att:decision>Using a User-Interface laver</att:decision>		
<att:riskrdf:datatype="http: 2001="" www.w3.org="" xmlschema#integer"="">0</att:riskrdf:datatype="http:> 0000000000000000000http://www.m3.org/2001/XMLSchema#integerhttp://www.m3.org/2001/XMLSchema#integerhttp://www.m3.org/2001/XMLSchema#integerhttp://www.m3.org/2001/XMLSchema#integerhttp://www.m3.org/2001/XMLSchema#integerhttp://www.m3.org/2001/XMLSchema#integerhttp://wwww.m3.org/2001/XMLSchema#integer <td></td> <td></td>		

Fig 83 - Managing linked data into Callimachus (VI)

9. Finally, we can create a named query, i.e. a SPARQL query that you can manage within Callimachus (edit, delete, show it results...). Go to our LData folder and click on the "Query" option.

Tirefox *	+									-	8
localhost/8080/LData/?view								C 🚼 - Go	agik 🐪	Ρ 🗍 1	R 8
	Callimachus						Lookup		0-		
		ussion Describe His									
	Create O	Book		localhost.8080 /							
	Concept	Class		Updated	Readers	Subscribers	Contributors	Editors	Administrators		
	E Folder	Hypertext File			public system	everyone		power staff	admin		
	@ PURL	Page		13:25	public system	everyone		power staff	aomin		
	More options	Pipeline Script									
	more options	E Style	Domain								
		Text File	& Group								
		 Transform XQuery 	# Query								
			📓 Graph Document								
		More options >	& Invited User								
	This resource was last i	modified at 13:38							Callinachus		
sst:8080/LData/7create=http://local	host:8080/callimachus/1.0/types/Na	imedQuerv									

Fig 84 – Creating a SPARQL query with Callimachus (I)

Then, we can write our SPARQL query and save it within Callimachus, pressing the "Create" button at the bottom of the screen.

🔹 🐪 New Named Query +			- 0 🗾	
host 8080/LDstar/tcreate=http%34%2F%2Fiocalhost%3A8080%2Fcallmachus%2F1.0%2Ftypes%2FNamedQuery	☆ ▼ C 🔡 - Go	fîr ∀ C S - Google		
Callimachus	Leokup	0.		
<pre>Cumurations Textures texture textures textu</pre>				

Fig 85 - Creating a SPARQL query with Callimachus (II)

Then, give a name to the concrete query and click on the "Save" button.

Firefox +				- 0 ×
🗲 🖲 localhost 8080/LData/?create=http%3A%2F%2Flocalhost%3A8080%2Fcallimach	🗧 🕲 localhost 000/LData/Tcreate=http%3A%2F%2Fiocalhost%3A000%2Fcallimachus%2F1.0%2Ftypes%2FNamedQuery			P ∔ ☆ ID+
Callimachus		Lookup	•••	^
localhost:8080 » LData	Save As	*		
1 + 2 + @Cache-Control: 3 + @infer true	Name:	query_2		
4 # @view 5 # 6 PREFIX xad: <btp: <="" td="" www.w=""><td>Create O localhost:8080 / LDa</td><td>ata</td><td></td><td></td></btp:>	Create O localhost:8080 / LDa	ata		
6 PREFIX xadi <http: www.wi<br="">7 PREFIX xdi: <http: www.wi<br="">8 PREFIX rdi: <http: td="" www.wi<=""><td>.org 3.or</td><td>Updated</td><td></td><td></td></http:></http:></http:>	.org 3.or	Updated		
9 10 SELECT * WHERE (78 70 70 11) 🖈 query 1	13:39		
**	atextual example Id	13:25		
	Showing results 2 of 2			
Create		Save Cancel	>	

Fig 86 - Creating a SPARQL query with Callimachus (III)

Finally, we may see the query results.

ocalhost 8000/LData/qu	uery_21view		🖄 🔻 C 🛛 🔂 - Google		P 🖡 🏦
	Callimachus		Laokup -	0-	
	Ideathost 8080 + LData + query 2 QUERY 2 View Edit Discussion Describe	History			
	8	р	0		
	http://localhost.8080/callimachus/changes/2013/07 /22/t14005afd01bx0	http://www.w3.org /ns/prov#wasGeneratedBy	http://localhost.8080/callimachus/changes/2013/07/22/t14005af401bx0Wprovenance		
		http://www.w3.org /ns/proi#wasGeneratedBy	http://localhost.8080/callimachus/changes/2013/07/22/t14005afd01bx0#provenance		
		http://www.w3.org /ns/prov#wasGeneratedBy	http://localhost.8080/callimachus/changes/2013/07/22/14005afd01bx0Wprovenance		
		http://www.w3.org /ns/prov#wasGeneratedBy	http://localhost.8080/callimachus/changes/2013/07/22/t14005afd01bx0#provenance		
	http://localhost.8080/callimachus/changes/2013/07 /22/t14005afd01bx1	http://www.w3.org /ns/prox#wasGeneratedBy	http://localhost.8080/callimachus/changes/2013/07/22/t14005afd01bx1Wprovenance		
	http://localhost.8080/callimachus/changes/2013/	http://www.w3.org /ns/prov#wasGeneratedBy	http://localhost.8080/callimachus/changes/2013/07/22/t14005afd01bx1#provenance		
		http://www.w3.org /ns/prov#wasGeneratedBy	http://localhest:8080/callimachus/changes/2013/07/22/t14005afd01bx1Mprovenance		
	http://localhost:8080/callimachus/changes/2013/07 /22/t14005afd01bx2	http://www.w3.org /ns/proi#wasGeneratedBy	http://localhost:8080/callimachus/changes/2013/07/22/t14005afd01bx2#provenance		
		http://www.w3.org /ns/pro#wasGeneratedBy	http://localhost:8080/callimachus/changes/2013/07/22/t14005afd01bx2#provenance		
	http://localhost.8080/callimachus/changes/2013/07 /22/t14005afd01bx3	http://www.w3.org /ns/prov#wasGeneratedBy	http://localhost.8080/callimachus/changes/2013/07/22/t14005afd01bx3#provenance		
		http://www.w3.org /ns/pro#wasGeneratedBy	http://localhost:8080/callimachus/changes/2013/07/22/t14005afd01bx3#provenance		
	http://localhost.8080/callimachus/changes/2013/07 /22/t14005afd01bx4	http://www.w3.org /ns/prov#wasGeneratedBy	http://localhost:8080/callimachus/changes/2013/07/22/t14005atid01bx4#provenance		
		http://www.w3.org /ns/proi#wasGeneratedBy	http://localhost:8080/callimachus/changes/2013/07/22/t14005afd01bx4#provenance		
	http://localhost.8080/auth/groups/users	http://www.w3.org /ns/prov#wasGeneratedBy	http://localhost.8080/callimachus/changes/2013/07/22/t14005af401bx5#provenance		
	http://iocalhost.8080/auth/groups/power	http://www.w3.org /ns/proi#wasGeneratedBy	http://localhost:8080/callimachus/changes/2013/07/22/t14005af401bx5#provenance		
	http://localhost.8080/auth/groups/staff	http://www.w3.org	http://localhost.8080/callimachus/changes/2013/07/22/114005afd01bx5#provenance		

Fig 87 – SPARQL query results

4.9 Other tools

This section provides some information about several Linked Data tools that we couldn't test properly due to the fact of problems with installation, start-up or data management. Therefore, these applications are not going to take part in the feature analysis presented in section 5.

4.9.1 Longwell

Longwell [20] is a web-based RDF-powered highly-configurable faceted browser. This application is written as an open source Java web application designed to let you visualize, browse and search an RDF complex dataset, especially large ones.

Longwell mixes the flexibility of the RDF data model with the effectiveness of the faceted browsing UI paradigm, allowing you to build a user-friendly web site out of your data within minutes and without requiring any code at all.

This tool has been developed by *Massachusetts Institute of Technology* (MIT) through the Simile project.

4.9.1.1 Architectural Knowledge as Linked Data: step by step with Longwell

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *Longwell*.

- 1. First of all, we need some specific requirements in order to run Longwell:
 - A Java 1.5 or later compatible virtual machine for our operating system
 - An installation of Apache Maven 2.0 or later (<u>http://maven.apache.org/download.cgi</u>)
 - (Optional) A servlet container (such as Jetty or Apache Tomcat) or any J2EE compatible application server (such as Apache Geronimo, JBoss, WebSphere, Orion or Weblogic)
- 2. Go to <u>http://simile.mit.edu/dist/longwell/</u> and download the latest release of Longwell. At the time of writing, the latest version is 2.5.5. In this case, we are going to download the

ZIP file instead of the TAR.GZ one. Then, unzip this file in order to start to use Longwell in our machine.

3. Then, open the terminal DOS Prompt on Windows. Go to the base directory of the Longwell distribution that we have already downloaded and type *mvn package* to download the dependencies and build the code, as you see in Fig 88.



Fig 88 – Starting with Longwell

4. Type ./longwell -r path/to/RDF/data/ on Unix or MacOSX, or longwell /r path\to\rdf\data\ on Windows, where "path to RDF data" is the a location on your disk where the RDF data you want to load is currently stored.

NOTE: Longwell is capable of loading files with the extension RDF, RDFS, OWL, N3 and RSS. If the path is a directory, Longwell will recursively scan it for all files that look like RDF. If the file is GZIP compressed (e.g., ends in .rdf.gz), Longwell will automatically decompress it before reading it.

5. Once Longwell has stopped initializing, loading and processing the data (which might take from a few seconds to hours, depending on the size of your dataset), we can point our browser at http://127.0.0.1:8080/ and enjoy it.

Notice that we don't include this tool in our feature analysis given that it doesn't work properly, as you can see in Fig 89 when we execute step 3, so navigating to http://127.0.0.1:8080/ doesn't show anything. In addition, Longwell is a retired project, as we may see in http://simile.mit.edu/mail.html, so it doesn't be supported anymore.

C:\Windows\system32\cmd.exe - 🗆 🗙
C:\Users\Cris\Desktop>cd longwell-2.5.5
C:\Users\Cris\Desktop\longwell-2.5.5>mun package [INFO] Scanning for projects [WARNING]
[WARNING] Some problems were encountered while building the effective model for edu.mit.simile:longwell:jar:2.5.5 [WARNING] 'build.plugins.plugin.version' for org.mortbay.jetty:maven-jetty-plugi
n is missing. @ line 64, column 15 [WARNING] 'build.plugins.plugin.version' for org.codehaus.mojo:minijar-maven-plu vic dia giogram. @ line.021 or line.15
gin is missing. @ line 87, column 15 [WARNING] 'build_plugins.plugin.version' for org.apache.maven.plugins:maven-comp iler-plugin is missing. @ line 36, column 12
[WARNING] 'build plugins plugin devsion' for organache maden plugins:maden-war-
UMARNING) 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-ecli [WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-ecli pse-plugin is missing. C line 57, column 15 [WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-sour
LWARNINGJ 'bulld.plugins.plugin.version' for org.apache.maven.plugins:maven-sour ce-plugin is missing. @ line 44, column 15 [WARNING] 'reporting.plugins.plugin.version' for org.apache.maven.plugins:maven-
Javadoc-plugin is missing. C line 100, column 15 [WARNING] 'reporting.plugins.plugin.version' for org.apache.maven.plugins:maven-
assembly-plugin is missing. @ line 120, column 15 [WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten t he stability of your build.
[WARNING] [WARNING] For this reason, future Maven versions might no longer support buildin g such malformed projects.
ĒVARNING] LINFO]
[INFO] [INFO] Building SIMILE Longwell 2.5.5 [INFO]
[WARNING] The POM for edu.mit.simile:fresnel:jar:1.4-alpha-SNAPSHOT is missing, no dependency information available
[WARNING] The POM for edu.mit.simile:banach:jar:1.0-SNAPSHOT is missing, no depe ndency information available
INFO] INFO] BUILD FAILURE INFO]
[INFO] Total time: 0.890s [INFO] Finished at: Fri Jun 21 14:01:33 CEST 2013
[INFO] Final Memory: 6M/106M [INFO]
[ERROR] Failed to execute goal on project longwell: Could not resolve dependenci es for project edu.mit.simile:longwell:jar:2.5.5: The following artifacts could not be resolved: edu.mit.simile:fresnel:jar:1.4-alpha-SNAPSHOT, edu.mit.simile:b
anach:jar:1.0-SNAPSHOT: Failure to find edu.mit.simile:fresnel:jar:1.4-alpha-SNA PSHOT in http://repositor.aduna-software.org/maven2-snapshots was cached in the
local repository, resolution will not be reattempted until the update interval of aduna-snapshot-repo has elapsed or updates are forced \rightarrow [Help 1]
[ERROB] [ERROR] To see the full stack trace of the errors, re-run Maven with the -e swit ch.
cn. [EERROR] Re-run Maven using the -X switch to enable full debug logging. [EERROR]
[ERROR] For more information about the errors and possible solutions, please rea d the following articles: [ERROR] [Help 1] http://cviki.apache.org/confluence/display/MAUEN/DependencyReso
lutionException
C:\Users\Cris\Desktop\longwell-2.5.5>

Fig 89 – Longwell error

4.9.2 Pubby

Pubby [21] can be used to add Linked Data interfaces to SPARQL endpoints. Therefore, Pubby makes it easy to turn a SPARQL endpoint into a Linked Data server. It is implemented as a Java web application.

Many triple stores and other SPARQL endpoints can be accessed only by SPARQL client applications that use the SPARQL protocol. It cannot be accessed by the growing variety of Linked Data clients. Pubby is designed to provide a Linked Data interface to those RDF data sources. Fig 90 shows its architecture.

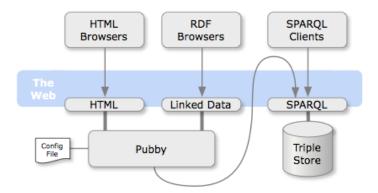


Fig 90 – Pubby architecture

When setting up a Pubby server for a SPARQL endpoint, you will configure a mapping that translates those URIs to dereferenceable URIs handled by Pubby. It will handle requests to the mapped URIs by connecting to the SPARQL endpoint, asking it for information about the original URI, and passing back the results to the client. It also handles various details of the HTTP interaction, such as the content negotiation between HTML, RDF/XML and Turtle descriptions of the same resource.

So, to sum up, its main features are:

- Linked Data interface to local or remote SPARQL protocol servers
- Dereferenceable URIs by rewriting URIs found in the SPARQL-exposed dataset into the Pubby server's namespace
- A simple HTML interface showing the data available about each resource
- Compatible with Tomcat and Jetty servlet containers
- A metadata extension to add metadata to the provided data

Pubby has some limitations too, such as:

- Only works for SPARQL endpoint that can answer DESCRIBE queries
- Multiple dataset support may not work as expected: If a requested URI is matched by the *conf:datasetURIPattern* of more than one dataset (or one doesn't have a *conf:datasetURIPattern*), then only one of the possible endpoints will be queried at a time. Pubby will never try to query multiple endpoints in order to create a single response. In most cases, it is recommended to simply set up a separate Pubby instance for each dataset.
- Hash URIs on the web side are not supported

4.9.2.1 Architectural Knowledge as Linked Data: step by step with Pubby

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *Pubby*.

- 1. First of all, go to <u>http://wifo5-03.informatik.uni-mannheim.de/pubby/download/</u> and download the latest version of Pubby which is 0.3.3 at the time of writing.
- 2. If you haven't already, download and install a servlet container, like Tomcat or Jetty (see 4.5.1 to use 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*.
- 4. Before starting Pubby, we have to modify the configuration file to suit our needs. This file is located within Pubby's *webapp* directory, at */WEB-INF/config.ttl*. In our case, this is the final look of the configuration:

```
# Example configuration which loads a static RDF file
     # and re-publishes it (textual example LD.rdf).
     # This configuration allows using Pubby as an RDF server for publishing
static RDF files.
     # Assumes Pubby is running at http://localhost:8080/
     @prefix conf: <http://richard.cyganiak.de/2007/pubby/config.rdf#> .
     <> a conf:Configuration;
         conf:projectName "Example";
         conf:webBase <http://localhost:8080/>;
         conf:projectHomepage <http://archk.tk/>;
         # When the homepage of the server is accessed, this resource will
         # be shown.
         conf:indexResource <http://archk.tk/DD1>;
         conf:dataset [
          conf:datasetBase <http://archk.tk/>;
          conf:addSameAsStatements "true";
          conf:loadRDF <textual_example_LD.rdf>;
         ];
```

Notice that this configuration loads a static RDF file and re-publishes it, so it allows using Pubby as an RDF server for publishing RDF files. These files have to be at the same directory as our configuration file. In addition to this, if we change the name of the configuration file, we have to indicate it on the *web.xml* file, which is located at the same folder.

5. Now we may start Pubby, going to our Jetty directory and writing the following line in the console: *java –jar start.jar*. Then, if we go to <u>http://localhost:8080</u> through our browser, we can see the initial page:

Firefox T DD1 Example	+		- 0 ×
♦ ♦ Ø localhost:8080/page/DD1		☆ ▽ 연 🛛 🗧 Google	₽ 🖡 🏠 🖸 ד
DD1 at Example http://localhost:8080/DD1			\$
Property	Value		
?:Layer	Business		
?:author	Cris		
?:category	 Structural 		
?:classification	Existence		
?:constrains	 <http: dd2="" localhost.8080=""></http:> <http: dd3="" localhost.8080=""></http:> <http: dd4="" localhost.8080=""></http:> <http: dd5="" localhost.8080=""></http:> 		
?:decision	 Using a three-layered architecture 		
?:rationale	Rationale1		
?:risk	 0 () 		
?:scope	 Universal 		
?:state	Approved		
?:timeStamp	2012-05-01T13:55:33.280368+01:00 ()		
?:type	 DesignDecision 		

Fig 91 – Initial page in Pubby

Through this page, we can navigate between the different Design Decisions of our RDF data, given that they are written as links, for example by clicking on <u>http://localhost:8080/DD3</u>, information about design decision DD3 is displayed:

	+
🛞 localhost:8080/page/DD3	$\begin{array}{c} \begin{array}{c} \begin{array}{c} \\ \end{array} \end{array} = \mathcal{C} \end{array}$
DD3 at Example	
ttp://localhost:8080/DD3	
mp.//localitost.0000/DD3	
Property	Value
?:Layer	 Business
author?:author?	Rafael
?:category	Structural
?:classification	Existence
s ?:constrains of	<http: dd1="" localhost:8080=""></http:>
?:constrains	 <htp: 8080="" d04-<="" li="" localhost=""> <htp: 8080="" d05-="" localhost=""></htp:> <htp: 8080="" d07-="" localhost=""></htp:> <htp: 8080="" d07-="" localhost=""></htp:> </htp:>
?:decision	Common Business Logic is Packaged as a Layer
?:excludes	 - command based on the charged as a carry in - white//localhost.8080/DD6>
?:rationale	The business logic of the application must be encapsulated in a set of components for both the mobile application and the web server; the common set of components is defined as a separate layer.
risk	• 0 ()
?:scope	Universal
?:state	Approved
?:timeStamp	2012-05-01T13:55:33.280368+01:00 ()
?:type	DesignDecision
s Turtle As RDF/XML Browse	in Disco Browse in Tabulator Browse in OpenLink Browser

Fig 92 – Navigating through Pubby

Notice that we don't include this tool in our feature analysis due to the fact it doesn't give access and query to data. We can use query agents as <u>SemWeb Client Library</u> or <u>SWIC</u> to carry out SPARQL queries and access our data, thus Pubby itself doesn't satisfy our initial requirements.

4.9.3 AK-RDF

AK-RDF is a query system developed for managing and querying Architectural Knowledge networks. In this sense, this application is designed for documenting all design decisions of a particular Software Architecture, simulating the network created among them. Thus, we can access and execute SPARQL queries for recover information from this network in an easy, fast and efficient way. Notice that the Architectural Knowledge will be represented using RDF.

In addition, AK-RDF provides functionality to register user queries in order to access them at any time. As well it supports five possible types of servers in order to perform store and query tasks, but the server installation process is responsibility of the user, namely:

- Virtuoso (more information in section 4.1)
- AllegroGraph (not supported on Windows systems)
- Apache Jena (more information in section 4.3)
- 4Store (not supported on Windows systems)
- Sesame (more information in section 4.5)

So, the user has to install one (or more) of these servers, either locally or remotely (i.e. accessed only through its URI), in order to use the application. AK-RDF can be used in Windows and Linux environments.

This application provides the following functionality:

- Architectural Knowledge management
 - Create a new element: Requirement, Component or Design Decision
 - o Edit an element
 - o Delete an element
- Database query service
 - o Execute customized query
 - o Execute predetermined query
- Dataset management
 - o Register a new dataset
 - Open a dataset
 - o Delete a dataset
- Query management
 - o Register a new query
 - $\circ \quad \text{Edit a query} \\$
 - o Delete a query
- System settings

4.9.3.1 Architectural Knowledge as Linked Data: step by step with AK-RDF

In this section, we are going to describe in detail the steps you have to follow in order to manage Architectural Knowledge as Linked Data, using *AK-RDF*. Notice that we are working with Windows.

- AK-RDF has been developed with MonoDevelop/Xamarin Studio, therefore we have to download *Mono for Windows* in order to execute it properly. Go to <u>http://www.gomono.com/mono-downloads/download.html</u> and select the latest stable version available. At the time of writing, the latest release is 2.10.9.
- 2. Then, run the setup file in order to install Mono for Windows on your computer. Once you have done it, execute the Mono Command Prompt, locate the path on your AK-RDF-Windows directory and write *mono RDFinterface.exe*, as you see in Fig 93.

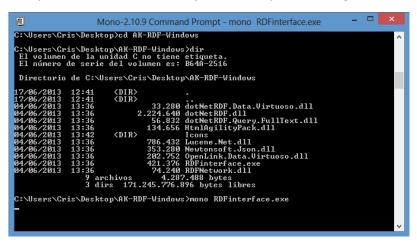


Fig 93 – Mono Command Prompt

After that, AK-RDF main interface is running, as you see in Fig 94.

😵 Sistema de	consulta para una red de conocimiento ar	rquitectónico – 🗆 🗙
Archivo Editar DataSet Ayuda		
🗅 🖻 😹 🗊 🖸 🥱 🍖 🔳 🔺 🔴	% ∃	
Consultas Predeterminadas	Consulta Personalizada SELECT ?s ?p ?o	
Ejecutar	WHERE({?s ?p ?o} UNION {GRAPH ?g {?s ?p ?o}} }	
	Registrar	Consultar

Fig 94 – AK-RDF main interface

3. Next, we have to register a new dataset in order to use a specific server, in this case we are going to choose Fuseki server. Go to Archivo-Nuevo DataSet, as you see in Fig 95.

Sister	na de (consulta para una red de conocimiento arquitectónico	- 🗆 🗙
Archivo Editar DataSet Ayuda			
🗋 <u>N</u> uevo DataSet Ctrl+N 🖕		Sin diama and a sin diama	
Abrir DataSet Ctrl+O			
🗑 <u>E</u> liminar DataSet			
% Prefere <u>n</u> cias			
🐔 Salir Ctrl+Q			
Consultas Predeterminadas		Consulta Personalizada	
NONE	~	SELECT ?s ?p ?o WHERE{	
Ejecutar		{?s ?p ?o} UNION {GRAPH ?g {?s ?p ?o}}	
		}	
		Registrar	Consultar

Fig 95 – Registering a new server with AK-RDF (1)

Then, a new window will appear in order to register the new server. We select Fuseki option.

6	AK-RDF: Nuevo DataSet	×
NONE		•
NONE		
Fuseki		
AllegroGraph		
FourStore		
Sesame		
Virtuoso		
	<u>C</u> ancelar <u>A</u> cepta	r

Fig 96 - Registering a new server with AK-RDF (2)

And fill in the required information about, in this case, Fuseki server.

AK-RDF: Nuevo DataSet	×
Fuseki	~
URI Base	
http://localhost:3030/	
Store ID	
apache.snapshots	
URI	
http://localhost:3030/	
<u>C</u> ancelar <u>A</u> ce	ptar

Fig 97 - Registering a new server with AK-RDF (3)

4. Then, we have to select this dataset in order to work with it. Go to Archivo-Abrir DataSet.

8	Sister	na d	le d	ons	sulta para una red de conocimiento arc	quitectónico -	×
Archivo Editar DataSet Ayu	da						
<u>N</u> uevo DataSet Ctrl+N	4			S	4		
🗎 <u>A</u> brir DataSet Ctrl+O	-						
🗑 <u>E</u> liminar DataSet							
% Preferencias							
🐔 <u>S</u> alir Ctrl+Q							
Consultas Predeterminadas		~			ulta Personalizada LECT ?s ?p ?o		
		¥			HERE{		
Ejecutar				}	{?s ?p ?o} UNION {GRAPH ?g {?s ?p ?o}}		
					Registrar	Consultar	

Fig 98 – Opening server with AK-RDF (1)

Then, select the Fuseki server which we have already created and click Aceptar button.

®	AK-RDF: Abrir Dataset	
Туре	ID	
Fuseki	i apache.snapshots	
	<u>C</u> ancelar <u>A</u> ceptar	
		1

Fig 99 - Opening server with AK-RDF (2)

 The next step is to create our own design decision network with AK-RDF in order to manage it later. Go to DataSet-Nuevo Elemento...-Decisión de Diseño, as you can see in Fig 100, to create a new design decision.

🕫 Sistema de c	onsulta para una red de conocimiento ar	rquitectónico – 🗆 🗙
Archivo Editar DataSet Ayuda DataSet Ayuda Nuevo Elemento Editar Elemento Registrar consulta Editar consulta Editar consulta	 Requisito Componente Decisión de Diseño 	
Consultas Predeterminadas C generica V Ejecutar	SeLECT ?s ?p ?o SELECT ?s ?p ?o WHERE{ {?s ?p ?o} UNION {GRAPH ?g {?s ?p ?o}} }	
	Registrar	Consultar

Fig 100 – Creating a design decision network with AK-RDF (1)

Thus, Fig 101 illustrates an example of the creation of design decision DD1, including all its properties indicated at the beginning of section 4 in RDF/XML format, namely:

```
<rdf:Description rdf:about="http://archk.tk/DD1">
         <att:author>Cris</att:author>
         <att:category>Structural</att:category>
         <att:decision>Using a three-layered architecture</att:decision>
         <att:risk rdf:datatype="&xsd;integer">0</att:risk>
         <att:scope>Universal</att:scope>
         <att:state>Approved</att:state>
         <att:rationale>Rationale1</att:rationale>
         <rat:Layer>Business</rat:Layer>
         <att:timeStamp rdf:datatype="&xsd;dateTime">2012-05-01T13:55:33.280368+01:00</att:timeStamp>
         <kind:classification>Existence</kind:classification>
         <conn:constrains rdf:resource="http://archk.tk/DD2" />
         <conn:constrains rdf:resource="http://archk.tk/DD3" />
         <conn:constrains rdf:resource="http://archk.tk/DD4" />
         <conn:constrains rdf:resource="http://archk.tk/DD5" />
         <prop:type>DesignDecision</prop:type>
</rdf:Description>
```

8	AK-RDF: Nue	va D	ecisión de D	iseño				×
Identificador			Clase Existence					V
			Existence					•
Decisión (Headline) Using a three-layered architectu	Ire							
Base Lógica (Rationale)								
Rationale1								
Propiedad[es] especifica[s] (de la F Formato: PROPIEDAD - VALOR, y e								
Layer - Business								
Alcance	Estado	_	Categoría			Riesgo		
Universal	Approved	v	Structural			0		¥
Autor				Dia	Mes	Ai		
Cris				1	× 5	¥	2012 🗸	
Decision[es] constreñida[s] (Const	rains), <u>separados por comas</u>							
DD2, DD3, DD4, DD5								
Decision[es] excluida[s] (Excludes)	, <u>separados por comas</u>							
Decision[es] inhibida[s] (Inhibits),	separados por comas							
Componente[s] implementado[s]	(Implements), <u>separados por</u>	coma	<u>s</u>					
Requisito[s] original[es] (Trace Fro	m), <u>separados por comas</u>							
Componente[s] resultante[s] (Trac	e To), <u>separados por comas</u>							
Los campos en negrita son OBLIG	ATORIOS							
						Can	celar	Aceptar

Fig 101 - Creating a design decision network with AK-RDF (2)

Finally, we click on the Aceptar button and our design decision DD1 will be created.

This final step returns an error, so DD1 cannot be created. Apart from that, we don't include this tool in our feature analysis because it is based on Apache Jena and Fuseki (section 4.3) so that we are really using and analysing the same technology.

5 Feature analysis

In this section, we carry out an analysis between the Linked Data tools presented in section 4 in order to compare their features and determine what is/are the best one/s. Table 1, Table 2 and Table 3 show their own features. These tables detail some features that we have considered highly relevant in order to manage AK data, namely:

- *Type of tool* specifies the type of the analysed Linked Data tool.
- Interaction UI (User Interface) indicates if the Linked Data tool has a friendly user interface or not.
- *Data persistence* indicates if the Linked Data tool provides persistence for its stored data.
- Data storage specifies where the data is stored.
- *Query languages* points out the different query languages that we can use to manipulate our linked data.
- Supported schemas/vocabularies within the Linked Data tool, like XML or RDF.

- *Federated queries* indicates if the Linked Data tool supports data searching across multiple databases.
- *Input data formats* supported by the Linked Data tool in order to store and manage it.
- Query output formats provided by the Linked Data tool when you execute a query.
- *License* informs of the type of license that the Linked Data tool has.
- *Security* indicates how the Linked Data tool guarantees and ensures that the information is always safely stored.
- *RDF serialization formats* supported by the Linked Data tool, like RDF/XML, Turtle, etc.
- SDK^4 support indicates if the Linked Data tool allows you to create applications for a certain software framework.
- *Complexity* of the Linked Data tool, i.e. how easy is it to install, to manage data on it or to use. Namely, complexity of *Installation, Start-up* and *Data management*.

⁴ Software Development Kit

		Virtuoso	LMF	Apache Jena + Fuseki	
Тур	e of tool	Server	Server application	Java framework + SPARQL server	
Inte	raction UI	OpenLink Data Spaces (ODS) - Briefcase	LMF	Fuseki	
Data j	persistence	Yes	Yes	Yes (<i>loc=DIR</i> option when starting Fuseki)	
Dat	a storage	Virtuoso server	LMF server	Fuseki server	
Query	/ languages	SQL, SPARQL, XQuery, XPath 1.0, XSLT 1.0	SPARQL	SPARQL, RDQL	
	pported s/vocabularies	RDF, XML, FOAF, OWL	RDF, DC, SKOS, <i>RDF Path Program</i> (to adapt the search to our specific domain)	RDF, OWL, RDFS, DC, RDFa, OWL 2, RSS, VCARD, DB (for Database properties)	
Federa	ated queries	Yes (SPARQL 1.1)	Yes (SPARQL 1.1)	Yes (SPARQL 1.1)	
Input c	Input data formats XML, SQL, RDF, free text		CSV, Excel, XML, RDF	RDF, XML	
Query output formats		HTML, XML, JSON, Javascript, NTriples, RDF/XML, spreadsheet	SPARQL-XML, JSON, HTML, chart (Sgvizler), XML	JSON, XML, Text, CSV, TSV, SSE ⁵ (only in Apache Jena ARQ, <i>ResultSetFormatter</i> class)	
L	icense	GPL ⁶ v2 (Open source: OpenLink Virtuoso)	Apache license (Free software)	Apache license (Free software)	
Security		User permissions on folders and login to ODS	Authentication and authorization mechanism (pre-defined profiles)	Authentication and control of the number of concurrent requests can be added using an Apache server and either blocking the Fuseki port to outside traffic or by listening only the <i>localhost</i> network interface. Data can be updated without access control if the Fuseki server is started with the <i>update</i> argument. If started without that argument, data is read-only.	
RDF serialization formats		N-Triples, Turtle, N3, RDF/XML, RDFa	RDF/XML, N3, Turtle, N-Triples, JSON-LD (JSON for Linked Data)	Turtle, RDF/XML, N-triples, RDF/JSON, TriG, N-Quads, RDFa, N3	
SDł	< support	No	No	Yes	
	Installation	Low	Low	High (Jena Java libraries + Eclipse + Fuseki)	
Complexity	Start-up	Low	Low	Medium	
	Data management	Low	Medium	Medium	

Table 1 - Features of Linked Data tools (I)

⁵ SPARQL Syntax Expressions - <u>http://jena.apache.org/documentation/notes/sse.html</u> ⁶ GNU *General Public License*

		TopBraid Suite	Sesame	Mulgara	
Туре	Type of tool Suite of Semantic products		Java framework	RDF database	
Intera	action UI	TopBraid Composer/Live	OpenRDF Workbench	Mulgara Viewer	
Data p	ersistence	Yes (TBL)	Yes (In Memory Store repository - Persist parameter)	Yes (Resolver Database Class configuration)	
Data	storage	TBL Personal/Enterprise server	Sesame server	Mulgara server	
Query	languages	SPARQL	SPARQL, SeRQL	TQL, SPARQL	
	ported vocabularies	RDF, RDFS, OWL, RDFa, XML	RDF, RDFS, OWL, DC, FOAF, XML, DOAP, EARL, XPath, Sesame, Sesame QName, SKOS	RDF, OWL, DAML, OIL	
Federat	ted queries	Yes (SPARQL 1.1)	Yes (SPARQL 1.1)	No	
Input data formats		RSS/Atom, RDBMS, CMS, XML, JSON, Excel, HTML/Calais text, CSV, e-mail, UML, RDF	RDF	RDF, XML, RDBMS, RSS	
Query ou	tput formats	XML, HTML, Text/CSV, Text/TSV	BINARY, SPARQL/XML, SPARQL/JSON, SPARQL/TSV, SPARQL/CSV	HTML	
License		Commercial, Pay Licensed Closed Source ⁷	BSD licenses ⁸ (Free software)	Open Software License 3.0	
Security		Version control and governance ⁹	User and password (server)	The commercial code that Mulgara was originally based on, contained both transport layer security (SSL/TLS) and store level authentication/authorization. The open source releases of Mulgara contain no security infrastructure.	
RDF serializ	zation formats	RDF/XML, N3, N-Triples, Turtle	TriG, BinaryRDF, TriX, N-Triples, N-Quads, N3, RDF/XML, RDF/JSON, Turtle	RDF/XML, N3, N-Triples	
SDK support		No	Yes	No	
	Installation	Low	Medium	Low	
Complexity	Start-up	Low	Medium	Low	
	Data management	Medium	Low	Low	

Table 2 - Features of Linked Data tools (II)

⁷ Pay Licensed Closed Source is a category of commercial software licenses distinguished by the fact that you must pay to use a software and cannot view the source code.

⁸ A *BSD-style license* is a particular free software license that grants wide permissions on usage and redistribution of a program.

⁹ TBC can be used with Subversion (SVN). *Data governance* is a system that describe who can take what actions with what information, and when, under what circumstances, using what methods.

		RedStore	Callimachus	
Туре	of tool	RDF triplestore	Linked Data management system	
Intera	action UI	RedStore	Callimachus	
Data pe	ersistence	Yes (-t <options=hashes, mysql,="" or="" postgresql,="" sqlite,="" tstore="" virtuoso="">)</options=hashes,>	Yes	
Data	storage	HTTP server	Callimachus server	
Query l	languages	SPARQL, LAQRS, RDQL	SPARQL	
	ported vocabularies	RDF	RDF, RDFS, OWL, RDFa, SKOS, DC, FOAF, Freebase, GoodRelations, GeoNames, Open Graph Protocol, SIOC, VCARD	
Federate	ed queries	Yes (SPARQL 1.1)	Yes (SPARQL 1.1)	
Input data formats		RDF	RDF	
Query out	tput formats	XML, JSON, Table, CSV, TSV, HTML, Turtle, RDF/XML, N-Triples, RDF/XML (XMP Profile), RDF/XML (Abbreviated), RSS 1.0, Atom 1.0, GraphViz DOT format, RDF/JSON Triples, RDF/JSON Resource-Centric, N- Quads	HTML	
Lic	cense	GPL (Free software)	Apache License 2.0 (Free software)	
Sec	curity	User permissions for the database server (MySQL, PostgreSQL, Tstore and Virtuoso)	By default, it allows public read-only access and requires authenticated users for editing and deleting resources. Authorization is setup using user accounts and groups.	
RDF serializ	ation formats	RDF/XML, N-Triples, Turtle, N-Triples-plus, N3, TriG, RSS Tag Soup, guess, RDFa, N-Quads	RDF/XML, Turtle, RDFa	
SDK s	support	No	No	
	Installation	High	Medium	
Complexity	Start-up	Medium	Low	
	Data management	Low	Medium	

Table 3 - Features of Linked Data tools (III)

Notice that almost all Linked Data tools are able to provide federated queries as they support SPARQL 1.1. SPARQL [22] can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. Namely, *SPARQL 1.1 Federated Query* extension has been created for executing queries distributed over different SPARQL endpoints.

Now, in order to get to know which the best/worst Linked Data tools are regarding to their features, we are going to establish our preferences for every feature and how we are going to score them.

ANALIZED	FEATURES	PARAMETER	PREFERENCES	HOW TO SCORE
Туре	of tool	-	(Without preferences)	(Not scoring, only informative)
Intera	ction UI	IUI	A User Interface that allows users to interact with the Linked Data tool.	1 If the tool has a UI 0 Otherwise
Data persistence		DP	A platform that provides data persistence over the time.	1 If the tool provides data persistence 0 Otherwise
Data	storage	-	(Without preferences)	(Not scoring, only informative)
Query la	anguages*	QL1 QL2	A tool that supports, at least SPARQL. The more query languages supported the more preferable.	1 If the tool supports SPARQL 0 Otherwise (Number of query languages supported)/5
	oorted ocabularies*	SS1 SS2	A tool that supports, at least RDF and OWL as schemas. The more schemas supported the more preferable.	1 If the tool supports RDF and OWL as schemas 0 Otherwise (Number of supported schemas)/13
Federate	ed queries	FQ	A tool that has the option of carry out federated queries.	 1 If the tool provides federated queries 0 Otherwise
Input dat	a formats*	IDF1 IDF2	A tool that supports, at least RDF as an input data format. The more input data formats supported the more preferable.	1 If the tool supports RDF as an input data format 0 Otherwise (Number of input data formats)/11
Query output formats		QOF	The more query output formats supported the more preferable.	(Number of query output formats)/17
Lic	ense	Li	A tool with an Open Software license.	1 If the tool has an Open Software license 0 Otherwise
Sec	curity	Sec	A tool with security services.	1 If the tool has security services 0 Otherwise
RDF serialization formats*		RSF1 RSF2	A tool that supports, at least RDF/XML as a RDF file format. The more RDF file formats supported the more preferable.	1 If the tool supports RDF/XML as a RDF file format 0 Otherwise (Number of RDF file formats)/10
SDK s	support	SDKS	A tool that provides SDK support.	1 If the tool has SDK support 0 Otherwise
	Installation	CI		0.2 If CI is <i>Low</i> 0.1 If CI is <i>Medium</i> 0 If CI is <i>High</i>
Complexity	exity Start-up CSU	CSU	It is desirable a <i>Low</i> complexity in all cases.	0.3 If CSU is Low 0.15 If CSU is Medium 0 If CSU is High
	Data management	CDM		0.5 If CDM is Low 0.25 If CDM is Medium 0 If CDM is High
			Table 4 – Feature preferences	

Table 4 – Feature preferences

(*) At least one specific value, such as SPARQL or RDF. These features are compulsory.

Therefore, the formula used to mark each tool will be the following one:

 $Total \ score = QL1 \cdot SS1 \cdot IDF1 \cdot RSF1 \cdot (IUI + DP + QL2 + SS2 + FQ + IDF2 + QOF + Li + Sec + RSF2 + SDKS + CI + CSU + CDM)$

These parameters will be calculated using the rules presented in Table 4, namely in *How to score* column. Parameters QL1, SS1, IDF1 and RSF1 represent the compulsory features, therefore their values will be 1 or 0, depending on their support.

In this manner, the tool that obtains the highest score will be the best Linked Data tool analysed and thus, the lowest score will belong to the worst Linked Data tool. Notice that every feature is considered as important as the others, therefore they are equally scored, namely *1 point* per each feature at most –so the score varies between 0 and 1 and the maximum score is 12.

Table 5 shows the marks for each Linked Data tool. Notice that we have omitted the two features that are not scoring, i.e. *Type of tool* and *Data storage*.

			Virtuoso	LMF	Apache Jena & Fuseki	TopBraid Suite	Sesame	Mulgara	RedStore	Callimachus
Interaction UI		IUI	1	1	1	1	1	1	1	1
Data persistence		DP	1	1	1	1	1	1	1	1
Query la	nguages	QL1 QL2	1 5/5=1	1 1/5=0.2	1 2/5=0.4	1 1/5=0.2	1 2/5=0.4	1 2/5=0.4	1 3/5=0.6	1 1/5=0.2
Supported schemas/vocabularies		SS1 SS2	1 4/13=0.31	0 4/13=0.31	1 9/13=0.69	1 5/13=0.39	1 12/13=0.92	1 4/13=0.31	0 1/13=0.08	1 13/13=1
Federated queries		FQ	1	1	1	1	1	0	1	1
Input data formats		IDF1 IDF2	1 4/11=0.36	1 4/11=0.36	1 2/11=0.18	1 11/11=1	1 1/11=0.09	1 4/11=0.36	1 1/11=0.09	1 1/11=0.09
Query output formats		QOF	7/17=0.41	5/17=0.29	6/17=0.35	4/17=0.24	5/17=0.29	1/17=0.06	17/17=1	1/17=0.06
License		Li	1	1	1	0	1	1	1	1
Security		Sec	1	1	1	1	1	0	1	1
RDF serialization formats		RSF1 RSF2	1 5/10=0.5	1 5/10=0.5	1 8/10=0.8	1 4/10=0.4	1 9/10=0.9	1 3/10=0.3	1 10/10=1	1 3/10=0.3
SDK support		SDKS	0	0	1	0	1	0	0	0
	Installation	CI	0.2	0.2	0	0.2	0.1	0.2	0	0.1
Complexity	Start-up	CSU	0.3	0.3	0.15	0.3	0.15	0.3	0.15	0.3
	Data management	CDM	0.5	0.25	0.25	0.25	0.5	0.5	0.5	0.25
	TOTAL SCORE		8.58	0	8.82	6.98	9.35	5.43	0	7.3

Table 5 - Linked Data tools scores

As we can see in Table 6, the first position is for *Sesame* (9.35), second *Apache Jena and Fuseki* (8.82), then *Virtuoso* (8.58), *Callimachus* (7.3), *TopBraid Suite* (6.98), *Mulgara* (5.43), and the last ones, *RedStore* (0) and *Linked Media Framework* (0) that have no score because they do not support OWL as an schema (parameter SS1).

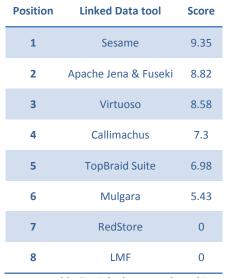


Table 6 - Linked Data tools ranking

6 References

- [1] Open Knowledge Foundation, "Open Data An Introduction," 2012. .
- [2] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data The Story So Far," *Int. J. Semant. Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, Jan. 2009.
- [3] T. Berners-Lee, "Linked Data Design Issues," 2009. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html. [Accessed: 18-Apr-2013].
- [4] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, vol. 1, no. 1. Morgan & Claypool, 2011, pp. 1–136.
- [5] OpenLink Software, "Virtuoso Universal Server," 2013. [Online]. Available: http://virtuoso.openlinksw.com/. [Accessed: 05-Jun-2013].
- [6] OpenLink Software, "OpenLink Data Spaces Framework & Application Suite," 2009.
 [Online]. Available: http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/OdsApplicationSuite.
 [Accessed: 05-Jun-2013].
- [7] OpenLink Software, "OpenLink Data Spaces (ODS)," 2013. [Online]. Available: http://my.openlinksw.com/ods/. [Accessed: 25-Sep-2013].
- [8] OpenLink Software, "OpenLink Virtuoso SPARQL Query Editor," 2013. [Online]. Available: http://my.openlinksw.com/sparql/. [Accessed: 25-Sep-2013].

- [9] OpenLink Software, "OpenLink iSPARQL," 2009. [Online]. Available: http://my.openlinksw.com/isparql/. [Accessed: 25-Sep-2013].
- [10] Semantic Web, "Linked Media Framework," 2012. [Online]. Available: http://semanticweb.org/wiki/Linked_Media_Framework.
- [11] The Apache Software Foundation, "Apache Jena," 2013. [Online]. Available: http://jena.apache.org/. [Accessed: 10-Jun-2013].
- [12] TopQuadrant, "TopBraid Suite," 2013. [Online]. Available: http://www.topquadrant.com/products/TB_Suite.html. [Accessed: 17-Jun-2013].
- [13] TopQuadrant, "TopBraid Composer," 2013. [Online]. Available: http://www.topquadrant.com/products/TB_Composer.html. [Accessed: 11-Jun-2013].
- [14] TopQuadrant, "TopBraid Live," 2013. [Online]. Available: http://www.topquadrant.com/products/TB_Live.html. [Accessed: 26-Jun-2013].
- [15] OpenRDF, "Sesame 2.7," 2013. [Online]. Available: http://openrdf.callimachus.net/sesame/2.7/docs/users.docbook?view. [Accessed: 11-Jun-2013].
- [16] Mulgara Project, "Mulgara," 2012. [Online]. Available: http://www.mulgara.org/. [Accessed: 24-Jun-2013].
- [17] T. Adams, "Mulgara Semantic Store," 2007. [Online]. Available: http://docs.mulgara.org/overview/faq.html. [Accessed: 24-Jun-2013].
- [18] N. J. Humfrey, "RedStore," 2011. [Online]. Available: http://www.aelius.com/njh/redstore/. [Accessed: 25-Jun-2013].
- [19] "Callimachus," 2013. [Online]. Available: http://callimachusproject.org/. [Accessed: 19-Jul-2013].
- [20] Simile, "Longwell," 2008. [Online]. Available: http://simile.mit.edu/wiki/Longwell. [Accessed: 20-Jun-2013].
- [21] C. Cyganiak, Richard; Bizer, "Pubby A Linked Data Frontend for SPARQL Endpoints," 2011. [Online]. Available: http://wifo5-03.informatik.uni-mannheim.de/pubby/. [Accessed: 25-Jun-2013].
- [22] W3C, "SPARQL 1.1 Federated Query," 2013. [Online]. Available: http://www.w3.org/TR/sparql11-federated-query/. [Accessed: 05-Jul-2013].
- [23] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker, "An empirical survey of Linked Data conformance," Web Semant. Sci. Serv. Agents World Wide Web, vol. 14, pp. 14–44, Jul. 2012.
- [24] C. Bizer, R. Cyganiak, and T. Heath, "How to publish Linked Data on the Web," 2008.
 [Online]. Available: http://wifo5-03.informatik.unimannheim.de/bizer/pub/LinkedDataTutorial/. [Accessed: 26-Jul-2013].

[25] A. Tang, "A Rationale-based Model for Architecture Design Reasoning," Swinburne University of Technology, 2007.