Linked Data Query Wizard: A Novel Interface for Accessing SPARQL Endpoints

Patrick Hoefler
Know-Center GmbH
Inffeldgasse 13
Graz, Austria
phoefler@know-center.at

Michael Granitzer
University of Passau
Innstraße 33a
Passau, Germany
michael.granitzer@uni-passau.de

Eduardo Veas
Know-Center GmbH
Inffeldgasse 13
Graz, Austria
eveas@know-center.at

Christin Seifert
University of Passau
Innstraße 33a
Passau, Germany
christin.seifert@uni-passau.de

ABSTRACT

In an interconnected world, Linked Data is more important than ever before. However, it is still quite difficult to access this new wealth of semantic data directly without having in-depth knowledge about SPARQL and related semantic technologies. Also, most people are currently used to consuming data as 2-dimensional tables. Linked Data is by definition always a graph, and not that many people are used to handle data in graph structures. Therefore we present the Linked Data Query Wizard, a web-based tool for displaying, accessing, filtering, exploring, and navigating Linked Data stored in SPARQL endpoints. The main innovation of the interface is that it turns the graph structure of Linked Data into a tabular interface and provides easy-to-use interaction possibilities by using metaphors and techniques from current search engines and spreadsheet applications that regular web users are already familiar with.

Keywords

Linked Data, User Interface, SPARQL, RDF Data Cube

1. INTRODUCTION

The amount of Linked Data available on the web keeps growing, mainly due to an influx of new data from research and open government activities. At the time of writing, 886 Linked Open Datasets had been registered with datahub.io, 389 of those claiming to provide a SPARQL endpoint[1]. However, it is still quite difficult to access this wealth of semantically enriched data directly without having in-depth knowledge of SPARQL and related semantic technologies. In order to exploit the full value of this data, as many people as possible, with diverse backgrounds and approaches, should be able to explore and analyze the data — and not just Semantic Web experts, as it is mostly the case right now.

When it comes to working with data, many people know how to use search engines and spreadsheet applications. In comparison there are only few people who know SPARQL, the W3C standard language to query Linked Data. While SPARQL endpoints provide enormous flexibility regarding the querying of Linked Data, there are also severe challenges:

- The data contained in a SPARQL endpoint is usually only accessible for experts in semantic technologies who know how to write SPARQL queries.
- Even for people who know how to write SPARQL, it can become quite laborious at times, especially when navigating and exploring an unfamiliar SPARQL endpoint by hand.

In this paper, we present the Linked Data Query Wizard[2], a novel way to explore the data contained in a SPARQL endpoint using a tabular interface.

The Linked Data Query Wizard is a web-based data analysis tool that empowers regular web users to explore, filter, and analyze Linked Data and should dramatically simplify the process of accessing any kind of Linked Data contained in SPARQL endpoints. The prototype currently offers two entry points: Users can either initiate a keyword search over a given SPARQL endpoint, or they can select any of the already available Linked Datasets represented as RDF Data Cubes (which will be explained in more detail in Section 4.2).

In both cases, the Linked Data Query Wizard presents a table containing the results. The users can then choose which columns they are interested in, and they can set filters to narrow down the displayed data. Additionally, they can explore the data by focusing on an entity, or they can aggregate a dataset to get a quick overview of the data.

This paper is structured as follows:

In Section 2 we discuss the research context and the requirements that defined the parameters for the development of the Linked Data Query Wizard.

[1] `http://datahub.io/dataset?tags=lod`

[2] `http://code.know-center.tugraz.at/search`

LDOW2014 April 8, 2014, Seoul, Korea

In Section 3 we take a quick look at which related approaches already exist.

In Section 4 we describe the Linked Data Query Wizard and its functionality in more detail.

In Section 5 we present and discuss the results of a user study we conducted to find out if the Linked Data Query Wizard was actually usable by people who had no knowledge about Semantic Web concepts and technologies.

Finally in Section 6 we present our conclusion and describe future work that could further enhance the Linked Data Query Wizard.

2. RESEARCH CONTEXT & SYSTEM REQUIREMENTS

The Linked Data Query Wizard has been developed in the context of the EU-funded CODE project[3]. As outlined in [13] and [12], the vision of the CODE project has been to establish a tool chain for the extraction of knowledge encapsulated in scientific research papers along with its release as Linked Data, thereby facilitating the creation of new insights. One of the project goals was the development of a web-based visual analytics platform that enables regular web users to easily perform exploration and analysis tasks on Linked Data.

With these prerequisites in mind, the following system requirements had been defined:

- **R1: The system needed to be completely web-based.** It had to be usable without the need for any client software other than an up-to-date web browser, and it must not rely on any browser plug-ins or extensions. Due to the potentially complex data analysis tasks, support for mobile clients with limited screen sizes was not a requirement.

- **R2: The system needed to support data from any domain.** It was clear from the start that the system should automatically adapt to any kind of Linked Data and not be tailored to a specific domain or use case.

- **R3: The system should be based on Semantic Web standards as much as possible.** Just as it should work with any kind of Linked Data, it should also work with any SPARQL endpoint that complies with the respective current W3C standards.

- **R4: The system needed to be easy to use.** Since the Linked Data Query Wizard is intended to be used by regular web users, the interface had to be kept simple. The end users should not know that they were actually accessing the Semantic Web through SPARQL queries.

- **R5: The system should make use of what regular web users already know.** In the context of this prototype, this mainly meant how to use current search engines and spreadsheet applications.

- **R6: The system should use the semantic aspects of the data to the advantage of the users.** This means that certain things should be easier or work smarter compared to working with non-semantic data.

- **R7: The system also needed to be useful for Semantic Web experts.** While mainly supporting regular web users, it should be possible to "peek behind the curtain" and provide helpful functionality for Semantic Web researchers and developers.

The main idea for the Linked Data Query Wizard was to make use of the prior knowledge the users already possessed when it came to handling data, to make them feel as comfortable as possible, and not to reinvent the wheel. The main assumption was that the relevant target group — people who are interested in looking up and handling data on the web — already know how to use current search engines and spreadsheet applications. Therefore the Linked Data Query Wizard should make use of these concepts:

- For getting started, using a simple search box known from current search engines

- For refining search results, using a table and concepts from current spreadsheet applications

- Enhancing the interface with further functionality, made possible through the semantic aspects of Linked Data

3. RELATED WORK

The problem of easy-to-use interfaces for accessing Linked Data is still largely unsolved.

The majority of current tools are not aimed at regular web users. As an example, Sindice [14], a major Semantic Web search engine, is practically unusable for ordinary web users due to its complex search interface and results page.

Moreover only very few web-based tools used tables for representing Linked Data. One such example was Freebase Parallax [7]. Although its main feature was the ability to browse sets of related things, it also provided a table view for these result sets.

Another web-based tool that shared similarities with our prototype was the Falcons Explorer [1]. Both tools featured a search box as the main entry point — an idea that is also central to our prototype. However, in both tools the table view was not the central focus.

Another tool that shares similarities with our prototype is OpenRefine[4] (formerly known as Google Refine and Freebase Gridworks). It supports RDF, and there are also extensions such as LODRefine[5] that focus on Linked Data. OpenRefine's main focus is cleaning up tabular data, and it's also not available as a web service, even though its main interface is browser-based.

Another concept related to our approach is faceted search and navigation as described e.g. in [10] or [3], and used in OpenRefine, SIMILE Exhibit [8] or DBpedia's instance of Virtuoso's Faceted Search & Find feature[6].

Although the Linked Data Query Wizard incorporates certain similarities, most interface elements and concepts are actually much more similar to those found in current spreadsheet applications than those used in faceted search and navigation.

[3] http://code-research.eu

[4] http://openrefine.org

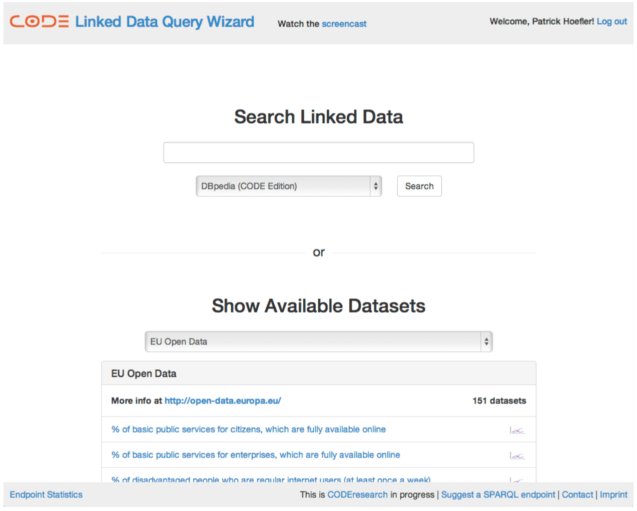
[5] http://code.zemanta.com/sparkica

[6] http://dbpedia.org/fct

4. THE LINKED DATA QUERY WIZARD

The Linked Data Query Wizard is a completely web-based tool for accessing Linked Data in SPARQL endpoints in an innovative way.

The front page of the Linked Data Query Wizard (see Figure 1) currently consists of two areas.



CODE Linked Data Query Wizard   Watch the screencast   Welcome, Patrick Hoefler! Log out

Search Linked Data

DBpedia (CODE Edition)   Search

or

Show Available Datasets

EU Open Data

EU Open Data

More info at http://open-data.europa.eu/   151 datasets

% of basic public services for citizens, which are fully available online

% of basic public services for enterprises, which are fully available online

% of disadvantaged people who are regular internet users (at least once a week)

Endpoint Statistics   This is CODEresearch in progress | Suggest a SPARQL endpoint | Contact | Imprint

Figure 1: The current front page of the Linked Data Query Wizard.

The top area is called “Search Linked Data”. It looks and works basically like current search engines: The central user interface element is the search box where the users can enter one or more search terms. These search terms are then turned into a SPARQL query and handed over to the chosen SPARQL endpoint. There, with the help of a full-text index, a search in all the rdfs:labels is performed, and the first 10 results are returned. Additionally, making use of the COUNT feature of SPARQL 1.1, the SPARQL endpoint is asked to return the total number of matching results to be displayed in the user interface.

The bottom area of the front page of the Linked Data Query Wizard is called “Show Available Datasets”. Here the users can choose from several lists of preexisting Linked Datasets that have been prepared and stored in the form of RDF Data Cubes.

In the remainder of this chapter we want to highlight different aspects of the Linked Data Query Wizard. To begin with, we will focus on the approach and hurdles of the SPARQL full-text search and explain the concept of RDF Data Cubes. Then we will showcase the tabular interface in more detail. Finally we will present the integration with other tools as well as advanced features for Semantic Web researchers and developers.

4.1 SPARQL Full-text Search

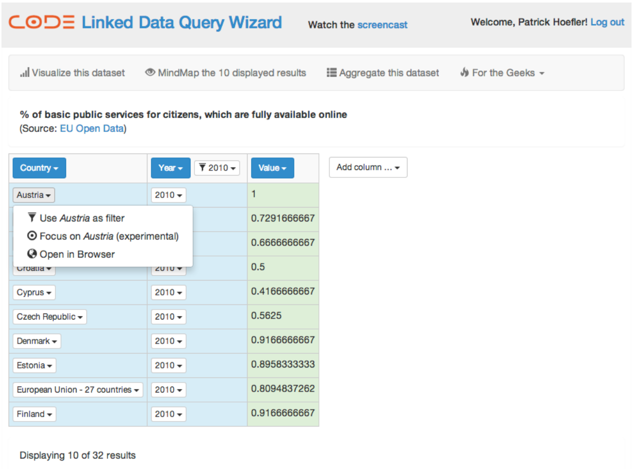
As already stated before, one of the main assumptions for the Linked Data Query Wizard was that the majority of its target group is accustomed to searching for information using one of the major search engines (Google, Bing or Yahoo). Therefore it soon became clear that the main entry point should be a simple search box that works and feels similar to what users currently expect when they search for information on the (non-semantic) web.

The technical implementation of the search feature turned out to be much more of a challenge: In the current version of the SPARQL 1.1 Query Language specification [5], the problem of performant full-text search is not addressed at all. The only officially specified way to search for something in a SPARQL endpoint is to filter the results using a regular expression. This approach, however, creates a potentially massive performance issue:

If the SPARQL query processor takes the specification literally, the only official way to “search” in a SPARQL endpoint is to first look up all matching results and then filter these results according to the regular expression. In the worst case this means going through all triples before starting to throw away the ones that do not match the filter criteria. Apart from memory considerations, a runtime performance of $\mathcal{O}(n)$ is simply not feasible in cases where the triple store contains millions or even billions of triples.

Due to this lack of specification, SPARQL endpoint vendors have come up with their own querying mechanisms for full-text search over SPARQL endpoints. Experiments in the initial phase of the development of the Linked Data Query Wizard showed that making use of these proprietary full-text search mechanism was the only way to achieve a system performance that regular web users had come to expect from current search engines. This is also the reason why the Linked Data Query Wizard currently only supports Virtuoso[7], OWLIM[8] and Bigdata[9] SPARQL endpoints in the “Search Linked Data” mode, since all of these provide integrated full-text search. The Linked Data Query Wizard has been designed to use Semantic Web standards as much as possible. Unfortunately, in the case of the full-text search feature, slightly different SPARQL queries are needed depending on the SPARQL endpoint software — sometimes even for individual SPARQL endpoints.

4.2 RDF Data Cubes



CODE Linked Data Query Wizard   Watch the screencast   Welcome, Patrick Hoefler! Log out

% of basic public services for citizens, which are fully available online
(Source: EU Open Data)

Country	Year ▼ 2010	Value
Austria	2010	1
[illegible]	[illegible]	0.7291666667
[illegible]	[illegible]	0.6666666667
Croatia	2010	0.5
Cyprus	2010	0.4166666667
Czech Republic	2010	0.5625
Denmark	2010	0.9166666667
Estonia	2010	0.8958333333
European Union - 27 countries	2010	0.8094837262
Finland	2010	0.9166666667

Displaying 10 of 32 results

Figure 2: An RDF Data Cube provided by the EU Open Data Portal is displayed in the Query Wizard.

W3C’s RDF Data Cube Vocabulary [2] provides a semantic framework for expressing datasets as Linked Data and

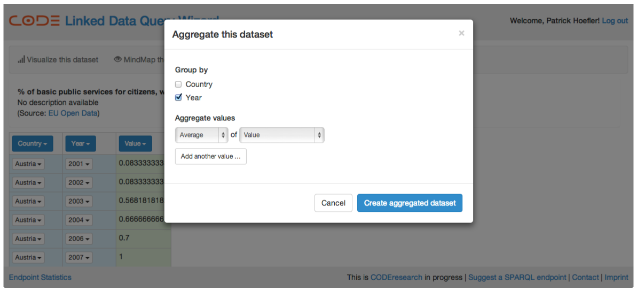
[7] http://virtuoso.openlinksw.com
[8] http://ontotext.com/owlim
[9] http://systap.com/bigdata.htm

therefore was a perfect fit for our purposes. Any datasets that comply with the RDF Data Cube standard and are publicly available through a SPARQL endpoint can easily be displayed, filtered, and explored using the Linked Data Query Wizard (see Figure 2).

The current version of the front page of the Linked Data Query Wizard features automatically generated lists of RDF Data Cubes for several publicly available SPARQL endpoints (such as EU Open Data [10] or Vienna Linked Open Data[11]).

Since the datasets are already pre-processed and mostly of reasonable size, full-text search is not necessary in this use case. This also means that the previously mentioned limitation regarding the SPARQL endpoint vendors does not apply when accessing RDF Data Cubes through the Linked Data Query Wizard.

Thanks to the underlying semantics of the Linked Data and the aggregation features of SPARQL 1.1, the Linked Data Query Wizard provides an easy interface to perform custom aggregations over any given RDF Data Cube, as long as it is saved in a publicly available SPARQL endpoint that supports SPARQL 1.1 (see Figure 3).



Welcome, Patrick Hoefler! Log out

Aggregate this dataset

Group by

□ Country

☑ Year

Aggregate values

Average of Value

% of basic public services for citizens, w
No description available
(Source: EU Open Data)

Country	Year	Value
Austria	2001	0.083333333
Austria	2002	0.083333333
Austria	2003	0.568181818
Austria	2004	0.666666666
Austria	2006	0.7
Austria	2007	1

CODE Linked Data Query Wizard
Visualize this dataset
MindMap th
×
Add another value ...
Cancel
Create aggregated dataset
Endpoint Statistics
This is CODEresearch in progress | Suggest a SPARQL endpoint | Contact | Imprint

Figure 3: Custom aggregation of an RDF Data Cube performed through the Linked Data Query Wizard.

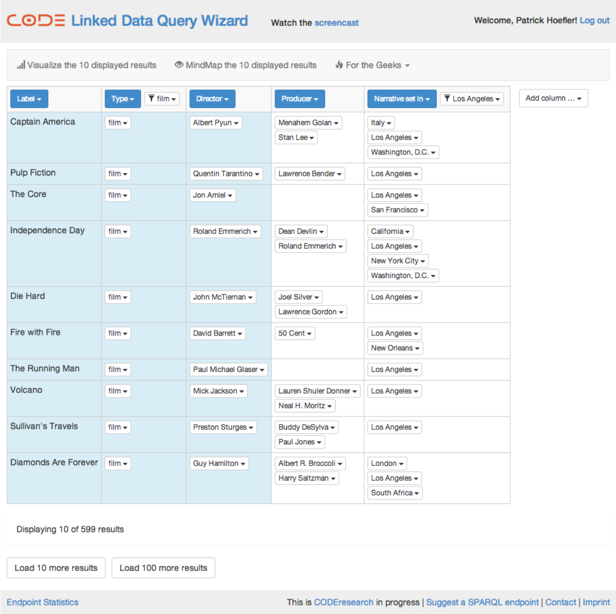
4.3 A Tabular Interface for Graph Data

After the users have either performed a full-text search over a SPARQL endpoint or selected a predefined Linked Dataset in the form of an RDF Data Cube, they are presented with the relevant results in the form of a table (see Figure 4).

The results table represents the underlying triples in the following way:

- Each row corresponds to a single subject.
- Each column represents a predicate. By default the first column displays the rdfs:label, the second column the rdf:type.
- Each cell contains the objects based on the respective row (i.e. subject) and column (i.e. predicate). Each cell can display zero, one ore more literals and/or entities represented by URIs.

Instead of the actual URIs, the Linked Data Query Wizard displays an rdfs:label, if one is available. If there is none, a short label is automatically generated based on the URI of the respective entity.



Welcome, Patrick Hoefler! Log out

Label	Type (film)	Director	Producer	Narrative set in (Los Angeles)
Captain America	film	Albert Pyun	Menahem Golan Stan Lee	Italy Los Angeles Washington, D.C.
Pulp Fiction	film	Quentin Tarantino	Lawrence Bender	Los Angeles
The Core	film	Jon Amiel		Los Angeles San Francisco
Independence Day	film	Roland Emmerich	Dean Devlin Roland Emmerich	California Los Angeles New York City Washington, D.C.
Die Hard	film	John McTiernan	Joel Silver Lawrence Gordon	Los Angeles
Fire with Fire	film	David Barrett	50 Cent	Los Angeles New Orleans
The Running Man	film	Paul Michael Glaser		Los Angeles
Volcano	film	Mick Jackson	Lauren Shuler Donner Neal H. Moritz	Los Angeles
Sullivan's Travels	film	Preston Sturges	Buddy DeSylva Paul Jones	Los Angeles
Diamonds Are Forever	film	Guy Hamilton	Albert R. Broccoli Harry Saltzman	London Los Angeles South Africa

Displaying 10 of 599 results

CODE Linked Data Query Wizard
Watch the screencast
Visualize the 10 displayed results
MindMap the 10 displayed results
For the Geeks
Add column ...
Load 10 more results
Load 100 more results
Endpoint Statistics
This is CODEresearch in progress | Suggest a SPARQL endpoint | Contact | Imprint

Figure 4: A typical results page of the Linked Data Query Wizard.

All entities (i.e. everything that has a URI) are displayed as buttons with appropriate interaction mechanisms.

In the case of the predicates displayed in the header of the table, the following functionalities are currently available:

- **Remove column.** As the name implies, this removes the current column (i.e. predicate) from the displayed table.
- **Hide empty results.** This hides all rows (i.e. subjects) that have no data for the predicate in question.
- **Add filter.** If the column contains text, numbers or dates, an appropriate filter can be set.
- **Group by.** When a RDF Data Cube is displayed, this provides a shortcut to the "Dataset Aggregation" functionality, with the current predicate already preselected in the "Group by" section of the dialog.

In the case of the objects displayed in the body of the table, the following functionalities are currently available:

- **Use as filter.** This sets the selected object as a filter for the current query.
- **Focus on.** This turns the selected object into the subject of a new query.
- **Open in Browser.** This opens the URI of the selected object in a new window. If the URI follows the Linked Data recommendations, the user should then see a human-readable version containing further information about the entity in question.

On the right side, next to the results table, users can add more predicates by clicking on the "Add column . . . " button. Users can then select the column from a list that displays

[10] http://open-data.europa.eu
[11] http://cweiss.net/lod

all available predicates that can provide additional data for one or more of the currently displayed subjects.

By default only the first 10 results are displayed for any given query. If there is more data available, users can load all results, 10 more results or 100 more results via buttons currently located below the results table.

4.4 Interfaces to External Tools

The Linked Data Query Wizard currently features 4 interfaces to other tools or services aimed at regular web users:

- CODE Visualization Wizard
- 42-data
- MindMeister
- Mendeley

4.4.1 CODE Visualization Wizard

Once the users are happy with their selected data, they can visualize it using the CODE Visualization Wizard[12] as described in [9]. The CODE Visualization Wizard is basically the sibling of the Linked Data Query Wizard. It enables visual analysis of Linked Data — in the form of RDF Data Cubes — and supports the user by automating the visualization process. This means that after analyzing the structural and semantic characteristics of the provided Linked Data, the CODE Visualization Wizard automatically suggests any of the 10 currently available visualizations — such as line charts, scatter plots, or parallel coordinates — that are suitable for the provided data. Furthermore the Vis Wizard automatically maps the data to the available visual channels of the chosen visualization. If the users wish to adjust the mapping, they can do so with a few simple clicks.

Usually more than one visualization is suitable for any given dataset. In that case, all of these visualizations can be displayed side by side. When certain parts of the data are selected in on of the visualizations, they are automatically highlighted in all of the others as well. This can provide quick insights into complicated data, taking advantage of the powerful human visual perception system.

4.4.2 42-data

42-data[13] is is the central data marketplace and integration hub of the CODE project. Users can integrate data from the Linked Data Query Wizard into answers on 42-data and thereby provide context for the data, making it even more valuable.

4.4.3 MindMeister

The MindMeister[14] mind mapping service is integrated into the Linked Data Query Wizard to turn the results table into a nicely looking mind map (see Figure 5). Each main branch of the mind map represents a subject from the results table, and the respective child branches represent the available predicates and objects. This feature is especially useful for getting a quick overview over a certain topic with only a handful of results.

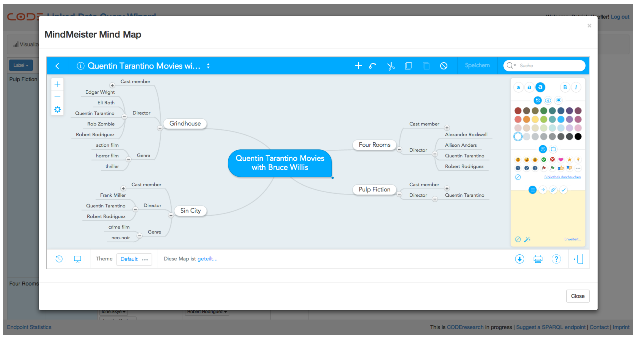


Figure 5: Data collected through the Query Wizard is displayed in a MindMeister Mind Map.

4.4.4 Mendeley

It is possible to log in to the Linked Data Query Wizard using an existing Mendeley[15] account. This feature is important for two reasons:

- The Mendeley User ID acts as a central identifying mechanism used by all CODE components. This facilitates the simple integration of all components, especially in the context of 42-data, the CODE Q&A Portal.
- Another important topic covered by the CODE project is provenance: Which user has created or modified a certain dataset, and what was the original source?

4.5 Features for Semantic Web Researchers and Developers

The main target group of the Linked Data Query Wizard are regular web users. However, it can also provide helpful resources for Semantic Web researchers and developers. The respective functionalities are currently grouped under the aptly named menu item “For the Geeks”. Currently these are:

- **Cubify the results.** For certain functionalities (e.g. the CODE Visualization Wizard), “generic” RDF needs to be turned into RDF Data Cubes first. This is done by the CODE Data Extractor[16], developed at the University of Passau. This conversion usually happens automatically in the background, without the need for any intervention by the users. However, via this menu button, the “Expert Mode” of the CODE Data Extractor can be activated, which offers more flexibility in case of problems with the data.
- **Display the SPARQL queries.** As the name implies, the Linked Data Query Wizard generates SPARQL queries according to the query and refinements made by the users. Regular web users are not interested in SPARQL queries at all — this is one of the main reasons why the Linked Data Query Wizard exists. However, for Semantic Web researchers or developers, it can be helpful to take a look or tweak the SPARQL queries generated by the Linked Data Query Wizard

[12] `http://code.know-center.tugraz.at/vis`

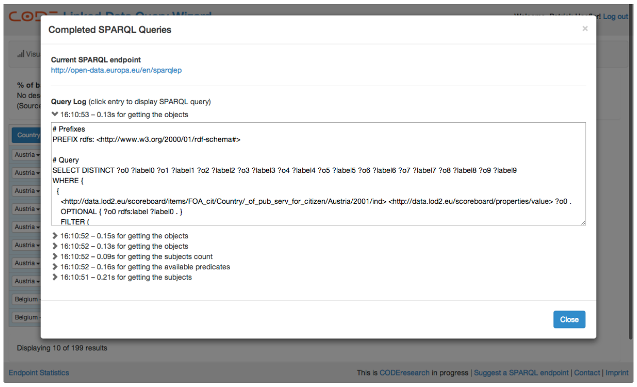
[13] `http://42-data.org`

[14] `http://mindmeister.com`

[15] `http://mendeley.com`

[16] `http://zaire.dimis.fim.uni-passau.de:8080/code-server/demo/dataextraction`

(see Figure 6). Additionally, this feature can be used for performance profiling, since not only the SPARQL queries are displayed, but also their respective runtime.



Completed SPARQL Queries

Current SPARQL endpoint
http://open-data.europa.eu/en/sparqlep

Query Log (click entry to display SPARQL query)

16:10:53 – 0.13s for getting the objects

```
# Prefixes
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

# Query
SELECT DISTINCT ?o0 ?label0 ?o1 ?label1 ?o2 ?label2 ?o3 ?label3 ?o4 ?label4 ?o5 ?label5 ?o6 ?label6 ?o7 ?label7 ?o8 ?label8 ?o9 ?label9
WHERE {
 {
  <http://data.lod2.eu/scoreboard/items/FOA_cit/Country/_of_pub_serv_for_citizen/Austria/2001/ind> <http://data.lod2.eu/scoreboard/properties/value> ?o0 .
  OPTIONAL { ?o0 rdfs:label ?label0 . }
  FILTER (
```

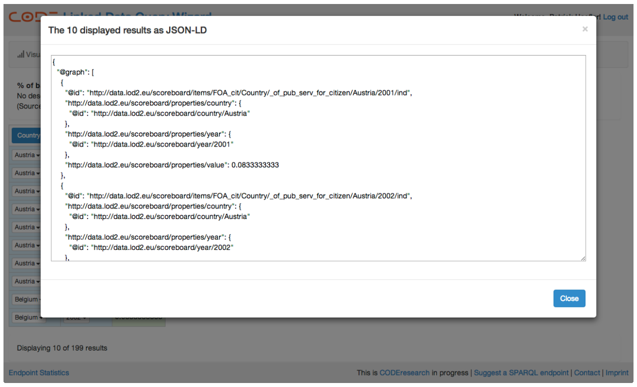
16:10:52 – 0.15s for getting the objects
16:10:52 – 0.13s for getting the objects
16:10:52 – 0.09s for getting the subjects count
16:10:52 – 0.16s for getting the available predicates
16:10:51 – 0.21s for getting the subjects

Displaying 10 of 199 results

Log out
Close
Endpoint Statistics
This is CODEresearch in progress | Suggest a SPARQL endpoint | Contact | Imprint

Figure 6: The SPARQL queries generated by the Linked Data Query Wizard and their respective runtime.

- **Display the results as JSON-LD.** Again, regular web users are probably not too interested in turning their search results into JSON-LD. For Semantic Web experts or programmers interested in getting started with JSON-LD, this can nevertheless be a helpful feature (see Figure 7).



The 10 displayed results as JSON-LD

```
{
 "@graph": [
  {
   "@id": "http://data.lod2.eu/scoreboard/items/FOA_cit/Country/_of_pub_serv_for_citizen/Austria/2001/ind",
   "http://data.lod2.eu/scoreboard/properties/country": {
    "@id": "http://data.lod2.eu/scoreboard/country/Austria"
   },
   "http://data.lod2.eu/scoreboard/properties/year": {
    "@id": "http://data.lod2.eu/scoreboard/year/2001"
   },
   "http://data.lod2.eu/scoreboard/properties/value": 0.0833333333
  },
  {
   "@id": "http://data.lod2.eu/scoreboard/items/FOA_cit/Country/_of_pub_serv_for_citizen/Austria/2002/ind",
   "http://data.lod2.eu/scoreboard/properties/country": {
    "@id": "http://data.lod2.eu/scoreboard/country/Austria"
   },
   "http://data.lod2.eu/scoreboard/properties/year": {
    "@id": "http://data.lod2.eu/scoreboard/year/2002"
   },
```

Displaying 10 of 199 results

Log out
Close
Endpoint Statistics
This is CODEresearch in progress | Suggest a SPARQL endpoint | Contact | Imprint

Figure 7: The search results provided by the Linked Data Query Wizard exported as JSON-LD.

5. EVALUATION

During the development of the Linked Data Query Wizard we followed the "release early, release often" principle. This means that as soon as a feature was complete and ready for testing, it immediately rolled out to our staging server and, if no major problems were found, a short time later (usually within hours, sometimes days) is was publicly available at our production server. This also means that the Linked Data Query Wizard has been under permanent scrutiny of fellow researchers from the CODE project as well as other interested colleagues for several months now. They regularly provided valuable feedback on stability and usability issues as well as helpful feature requests.

Additionally, the Linked Data Query Wizard was used in a workshop setting with around 20 students of the Semantic Technologies course at Graz University of Technologies. There it proved helpful in evaluating the quality of the Linked Open Data the students had previously created during the workshop.

To conclude the first development cycle of the Linked Data Query Wizard, an in-depth evaluation was performed. In the remainder of this section we will present the study design, both the quantitative as well as the qualitative study results, and a discussion of the findings.

5.1 Study Design

Our study followed the principles of the Retrospective Think Aloud protocol ([15], [4]), combined with the NASA Task Load Index ([6]).

In the following we will describe the details of the study.

In total, 14 people participated in this study, and 2 people took part in the related pre-study.

Each session started with a short explanation of the study and the signing of the declaration of consent. The session was then guided by a survey that was filled out by the participants themselves. The first page of the survey consisted of background questions about the participant:

- **How's your English?** The study was conducted in English with participants from different countries, but no English native speakers. 10 participants declared their English skills as "fluent", 3 as "okay" and 1 as "basic".
- **Have you used the Linked Data Query Wizard before?** For this study, only participants with no prior experience with the Linked Data Query Wizard were selected. Accordingly, all 14 participants answered with "no".
- **How frequently do you use spreadsheet applications?** The Linked Data Query Wizard is mainly intended for people that have prior experience with spreadsheet applications. Although this was not checked during the participant selection phase, all participants indeed had at least some experience: 2 of them used spreadsheet applications "every workday", 4 of them "several times a week", 6 of them "several times a month" and 2 of them "once a month or less often"
- **How frequently do you look up information on the Internet?** This question aimed to probe the level of the participants' web experience, which turned out to be quite high: 13 of the participants answered "every workday", one of them "several times a week".
- **How frequently do you write SPARQL queries?** This question was intended to find out if there were any Semantic Web experts among the participants. Only one of the 14 participants answered "once a month or less often", whereas 6 of them answered "never" and 7 of them "What's SPARQL?".
- **What's your age?** The final background question provided information about the age ranges of the participants: 4 of the participants were between 18 and 27 years old, 9 of them between 28 and 37, and 1 between 58 and 67.

After the initial background questions, the participants had to solve 4 tasks using the Linked Data Query Wizard. These were as follows:

- **Task 1: Service Data**

  "There is an available dataset called '% of basic public services for citizens, which are fully available online' provided by EU Open Data. We are interested only in the data from the year 2010, please filter it accordingly. After that, please visualize the results.

  You have 3 minutes to complete this task."

- **Task 2: Data Overview**

  "This task deals with the same data as before, the dataset called '% of basic public services for citizens, which are fully available online' provided by EU Open Data. However, this time we are interested in an overview of the data. Therefore, please aggregate the dataset and display the average values, grouped by year. After that, please visualize the results.

  You have 3 minutes to complete this task."

- **Task 3: Pulp Data**

  "Before you start, please select the data source called 'Wikidata (CODE Edition)' (5th from the top) in the 'Search Linked Data' section.

  There is a film called 'Pulp Fiction'.

  1. Was Bruce Willis a cast member of this film?

  2. Who was the director of this film?

  3. Are there any other films by the same director where Bruce Willis was a cast member? If so, which ones and how many?

  You have 5 minutes to complete this task."

- **Task 4: More Data**

  "Once again, before you start, please select the data source called 'Wikidata (CODE Edition)' (5th from the top) in the 'Search Linked Data' section.

  There is a music album that has the word 'Antidote' in it.

  1. Who is the performer / musical artist of this album? The one you are looking for starts with 'Mor…'.

  2. Which other albums has this artist released? Please make sure that only albums (and no singles) are displayed.

  3. Make a MindMap containing all the information that you just looked up.

  You have 5 minutes to complete this task."

After each task was finished — either by the participant successfully completing it, or by reaching the respective time limit — the participants filled out a NASA Task Load Index form and subjectively judged several aspects of the task they had just worked on. The form consisted of the following questions:

- **Mental Demand.** How mentally demanding was the task?

- **Physical Demand.** How physically demanding was the task?

- **Temporal Demand.** How hurried or rushed was the pace of the task?

- **Performance.** How successful were you in accomplishing what you were asked to do?

- **Effort.** How hard did you have to work to accomplish your level of performance?

- **Frustration.** How insecure, discouraged, irritated, stressed, and annoyed were you?

Additionally after each task the participants were asked the following question: "Any comments? What was good / bad / unexpected / difficult?" Firstly they were asked to write down what came to their minds. After that the study conductor asked about specific observations that he had made during the task. After a usually short, sometimes a little longer discussion, the participants added written remarks that came up during the discussion.

The final page of the survey consisted of four questions that gave the participants the opportunity to provide additional qualitative feedback. These questions were:

- What did you like about the Linked Data Query Wizard?

- What did you hate about the Linked Data Query Wizard?

- For which tasks would you personally use the Linked Data Query Wizard?

- If you could have solved the tasks with other tools of your choice, which ones would you have used?

After the participant had answered these final questions, the session was concluded.

The four tasks that the participants had to solve were basically divided into two groups:

- Tasks 1 and 2 concentrated on the "Show Available Datasets" mode and were intentionally of moderate complexity. Since all participants had, except for a short guided tour, no prior experience with the Linked Data Query Wizard, these tasks were intended to ease them into the system and to discover potential problems with the user experience at the same time.

- Tasks 3 and 4 concentrated on the "Search Linked Data" mode and were of significantly higher complexity. By then the participants had become more familiar with the Linked Data Query Wizard, since the learning effect should have already started to kick in.

It was clear that the lack of randomization of the tasks would lead to an uncompensated learning effect. This did not pose a problem under the circumstances, since it was not the goal of this study to compare the different tasks with each other, but rather to evaluate the general usefulness of the system and find its weak points with regard to user experience.

Apart from the quantitative feedback (NASA Task Load Index) and the qualitative feedback (Retrospective Think Aloud), the study conductor also measured the task completion rate.

Task completion time was not measured for several reasons. For example, the study was conducted on the live system and not on a lab setup, so fluctuations in (external) server response times were to be expected. Also, since the Linked Data Query Wizard offers a rather novel interface to access Linked Data, the main goal of the study was to show if users are able to use it at all and where potential problems in understanding arise.

Another idea was to go with a conventional Think Aloud study instead of using the Retrospective Think Aloud protocol. This would have had a negative impact on completion time and, due to the given time limits for completing each task, could have resulted in a lower task completion rate.

Also, competing the Linked Data Query Wizard against other tools was not really an option, since there are currently no tools that could come close enough in functionality to make a direct comparison feasible.

Another possibility would have been to compare against Google searches or SPARQL queries written by Semantic Web experts. However, in the first case, the test cases could have been constructed in a way where the Linked Data Query Wizard would have always won by a landslide, which would have defeated the purpose of a direct comparison. In the latter case, competing against manually created SPARQL queries was not ideal either, since the focus of this evaluation was on regular web users and not on Semantic Web experts.

5.2 Results and Discussion

Due to the combination of the Restrospective Think Aloud protocol with the NASA Task Load Index, it was possible to generated four different kinds of results from the study:

- Quantitative results (NASA Task Load Index)
- Quantitative results (Task Completion Rate)
- Comparison between participants with and without a background in computer science
- Qualitative results (Retrospective Think Aloud)

5.2.1 Quantitative Results (NASA Task Load Index)

The quantitative results of the NASA Task Load Index can be seen in the box plots of Figure 8. The six plots represent the results for the six different aspects of the NASA Task Load Index. Those were mental demand, physical demand, temporal demand, performance, effort, and frustration.

- The mental demand was rather low for the first two tasks and increased only slightly for the more complex last two tasks. The variance between the participants was quite high.
- The physical demand was, as expected, very low throughout the study.
- The temporal demand — with respect to the time limits of the tasks — basically corresponded with the results from the mental demand, showing a generally low demand with a high degree of variance between the participants.
- The performance scores were very high with a median of 10 out of 10 for all four tasks. Out of the 56 tasks performed in total by the 14 participants, 49 were successfully completed, 6 were not completed entirely in time, and only 1 was not completed at all.
- The subjective effort of the participants showed a high variance between the participants, however it also showed the learning effect very nicely: The effort necessary by the participants decreased after the first task, since the second task was similar to the first one. The third task was completely different, which raised the level of necessary effort again. The fourth task was similar to the third task, which again resulted in lower effort.
- The frustration level was rather low throughout the study, but again with a very high variance between the participants.

5.2.2 Quantitative Results (Task Completion Rate)

In addition to the subjective quantitative results measured via the NASA Task Load Index, the task completion rate was also measured objectively by the study conductor. There was, however, no significant difference between the subjective performance as judged by the participants themselves and the objectively measured task completion rate. In detail, this means:

- 13 out of 14 participants were able to complete task 1 completely. 1 participant only received 2 out of 10 points.
- All 14 participants were able to solve task 2 completely in time.
- Task 3 turned out to be the most difficult one: 10 out of the 14 participants were able to solve it completely, the other 4 participants only received 5 out of 10 points.
- Task 4 was completely solved by 12 out of the 14 participants. 1 participant only received 5 out of 10 points, and 1 participant received 0 points.

5.2.3 Background in Computer Science

An interesting research question came up in the preparation of the study: Would there be a significant difference in the results of participants with and participants without a background in computer science? For this reason, 7 of the study participants had a background in computer science, whereas the other 7 did not.

To determine if there was indeed a difference between these two groups, independent two-sample t-tests with equal sample size were performed, comparing all 24 results of the NASA Task Load Index (6 aspects * 4 tasks) as well as the objectively measured task completion rates. The result was that all calculated p-values were larger than 0.1. This means that for our study, no significant difference regarding the results of the subjective NASA Task Load Index or the objective task completion rate between participants with and without a background in computer science could be measured.

5.2.4 Qualitative Results

The qualitative results of the evaluation were based on the statements of the participants collected during the Retrospective Think Aloud phase of the study.

Regarding task 1, the main problem that the participants encountered concerned the filtering: To set a URI filter, the participants had to click on the respective entity (in task 1, it was the year 2010) and select "Add as filter" in its context menu. However, 10 of the 14 participants had problems with

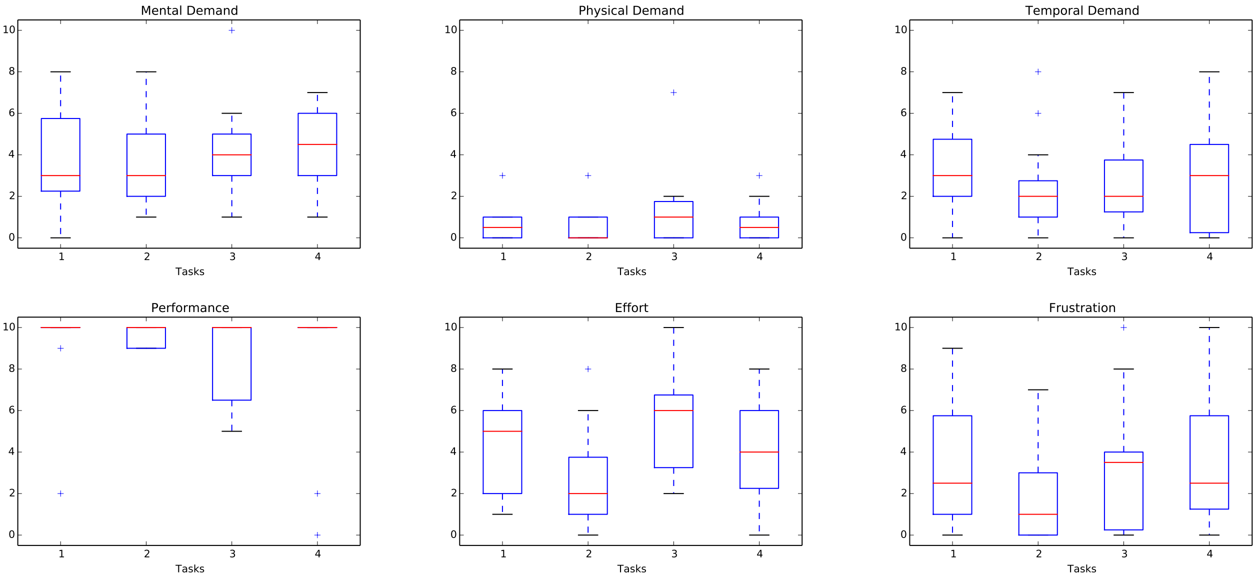


Figure 8: Quantitative results based on the NASA Task Load Index

setting the filter because they expected it to be set through a context menu item in the table header, as it is the case for other filters (text, number and date) in the Linked Data Query Wizard and in most current spreadsheet applications.

Regarding task 2, the feedback was much more positive, the use of the "Aggregate dataset" feature did not cause any major problems.

Regarding task 3, the combination of multiple filters turned out to be quite a challenge for the participants. Also, the fact that all cast members of the matching movies were displayed even after "Bruce Willis" had been set as a filter confused some users. Because all of the cast members were displayed for each movie, this also meant that the rows became quite high, which several participants found irritating.

The majority of task 4 did not pose a problem for the participants after having completed the similar third task. The fact that the relevant search result did not appear on the first, but on the second result page, caused huge confusion for almost all of the participants, even though the number of total results and the "Load all results" button were visible to all participants all of the time.

When asked about what they liked about the Linked Data Query Wizard, they general opinion was that once they had worked out how the filtering worked, the interface was easy enough to use. Additionally they liked how they could create a useful list of results from a huge database with only a few simple steps.

When asked about what they didn't like about the Linked Data Query Wizard, there was no general theme. Four of the participants mentioned that it would have been hard for them to choose a data source if they had not been told which one to use.

When asked about what they would personally use the Linked Data Query Wizard for, no clear trend could be recognized. The answers ranged from "don't know yet" and "statistical data" to "newspaper entries" and "exploration of data sources".

Finally, when asked about which other tools they would have used to solve similar tasks if the Linked Data Query Wizard had not been available, it became very clear what the direct competitors for the Linked Data Query Wizard were: Almost every participant immediately mentioned that they would use Google to search for data or information. The majority of participants mentioned that they would also use specialized portals to look for certain information, e.g. IMDB for data about movies. When it came to working with data, analyzing and visualizing it, almost all participants mentioned that they would use Microsoft Excel and that they would probably manually collect and copy the data into the spreadsheet.

6. CONCLUSION & FUTURE WORK

In this paper we introduced the Linked Data Query Wizard, a novel interface for accessing Linked Data in SPARQL endpoints, either through a keyword search or by selecting available Linked Datasets represented as RDF Data Cubes.

The results of the conducted user study showed that the tool had a few weak spots that could be improved, but was in general very usable, both for people with and without a background in computer science.

In the future we plan to address the main challenges that came up during the user study, mainly the possibility to add all types of filters through the table header.

Also the total number of results could be displayed more prominently, and the implementation of an "infinite scroll" mechanism that automatically displays more data as soon as the users scroll to the bottom of the screen could circumvent the problem that users need to load more data manually in order to find what they are looking for.

Another important point for improvement is the current limitation that users can only search one SPARQL endpoint at a time, which they need to select beforehand. The integration of services like Balloon Fusion [11] could help in this regard, providing SPARQL rewriting based on collected co-reference information combined with automatic endpoint discovery, resulting in an intelligent query federation.

7. ACKNOWLEDGEMENTS

The Linked Data Query Wizard has been developed within the CODE project at the Know-Center, Graz, Austria.

The CODE project is funded by the EU Seventh Framework Programme, grant agreement number 296150.

The Know-Center is funded within the Austrian COMET Program — Competence Centers for Excellent Technologies — under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth, and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency (FFG).

8. REFERENCES

[1] G. Cheng, H. Wu, S. Gong, W. Ge, and Y. Qu. Falcons Explorer: Tabular and Relational End-user Programming for the Web of Data. In *Semantic Web Challenge*, 2010.

[2] R. Cyganiak and D. Reynolds. The RDF Data Cube Vocabulary, 2013.

[3] O. Erling. Faceted Views over Large-Scale Linked Data. *Linked Data on the Web (LDOW)*, 2009.

[4] Z. Guan, S. Lee, E. Cuddihy, and J. Ramey. The validity of the stimulated retrospective think-aloud method as measured by eye tracking. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, page 1253, 2006.

[5] S. Harris and A. Seaborne. SPARQL 1.1 Query Language, 2013.

[6] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 904–908. Sage Publications, 2006.

[7] D. Huynh and D. Karger. Parallax and companion: Set-based browsing for the data web. *WWW Conference*, 2009.

[8] D. F. Huynh, D. R. Karger, and R. C. Miller. Exhibit: lightweight structured data publishing. *Proceedings of the 16th international conference on World Wide Web*, Banff, Alb:737–746, 2007.

[9] B. Mutlu, P. Hoefler, G. Tschinkel, E. Veas, V. Sabol, F. Stegmaier, and M. Granitzer. Suggesting Visualisations for Published Data. In *Proceedings of IVAPP 2014*, Lisbon, Portugal, 2014.

[10] E. Oren, R. Delbru, and S. Decker. Extending faceted navigation for RDF data. In *The Semantic Web ISWC 2006 5th International Semantic Web Conference ISWC 2006 Athens GA USA November 59 2006 Proceedings*, volume 4273, pages 559–572, 2006.

[11] K. Schlegel, F. Stegmaier, S. Bayerl, M. Granitzer, and H. Kosch. Balloon Fusion: SPARQL Rewriting Based on Unified Co-Reference Information. In *5th International Workshop on Data Engineering Meets the Semantic Web, co-located with the 30th IEEE International Conference on Data Engineering*, 2014.

[12] C. Seifert, M. Granitzer, P. Höfler, B. Mutlu, V. Sabol, K. Schlegel, S. Bayerl, F. Stegmaier, S. Zwicklbauer, and R. Kern. Crowdsourcing Fact Extraction from Scientific Literature. In *Workshop on Human-Computer Interaction and Knowledge Discovery (SouthCHI)*, volume 7947 of *LNCS*, Maribor, Slovenia, 2013. Springer.

[13] F. Stegmaier, C. Seifert, R. Kern, H. Patrick, S. Bayerl, M. Granitzer, H. Kosch, S. Lindstaedt, B. Mutlu, V. Sabol, K. Schlegel, and S. Zwicklbauer. Unleashing Semantics of Research Data. In *The Second Workshop on Big Data Benchmarking WBDB2012in*, 2012.

[14] G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the open linked data. *Lecture Notes in Computer Science*, 4825:552–565, 2007.

[15] M. Van Den Haak, M. De Jong, and P. Jan Schellens. Retrospective vs. concurrent think-aloud protocols: testing the usability of an online library catalogue. *Behaviour & Information Technology*, 22(5):339–351, 2003.