Exploring the Web of Spatial Data with Facete

Claus Stadler[*]
University of Leipzig
Augustusplatz 10
04109 Leipzig, Germany
cstadler@informatik.uni-leipzig.de

Michael Martin
University of Leipzig
Augustusplatz 10
04109 Leipzig, Germany
martin@informatik.uni-leipzig.de

Sören Auer
Universität Bonn & Fraunhofer IAIS
Römerstraße 164
53117 Bonn, Germany
auer@cs.uni-bonn.de

ABSTRACT

The majority of data (including data published on the Web as Linked Open Data) has a spatial dimension. However, the efficient, user friendly exploration of spatial data remains a major challenge. We present *Facete*, a web-based exploration and visualization application enabling the spatial-faceted browsing of data with a spatial dimension. Facete implements a novel spatial data exploration paradigm based on the following three key components: First, a domain independent faceted filtering module, which operates directly on SPARQL and supports nested facets. Second, an algorithm that efficiently detects spatial information related to those resources that satisfy the facet selection. The detected relations are used for automatically presenting data on a map. And third, a workflow for making the map display interact with data sources that contain large amounts of geometric information. We demonstrate Facete in large-scale, real world application scenarios.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Selection process*

Keywords

SPARQL; RDF; faceted search; geospatial data; property path retrieval; map visualization

1. INTRODUCTION

The growth of the Linked Open Data (LOD) cloud[1] goes hand in hand with an increasing amount of geospatial information being made available for online querying via SPARQL. For instance, the community projects DBpedia and LinkedGeoData provide access to more than 1.3 million and 2 billion point geometries based on Wikipedia and OpenStreetMap data, respectively. Government agencies have started maintaining public SPARQL endpoints, such as *Ordnance Survey*[2] and the *European Environment Agency*[3]. The latter, for instance, hosts 123 RDF graphs that contain a sizable number of resources described using the WGS84 vocabulary[4].

Yet, the user friendly exploration of spatial data accessible via public SPARQL remains a major challenge, especially due to performance, volume, scalability and UI complexity issues. In this demo, we present the spatial data exploration and visualization application *Facete*, depicted in Figure 1. Facete is a client-side JavaScript application, which interacts with a server-side SPARQL endpoint. The major problem when exploring spatial data is that the spatial dimension is in most cases not directly attached to all the data items. For example, a dataset about multi-partner research projects would contain entities for the projects itself, the project partners, their legal entities, the branch offices of these legal entities and finally the cities they are located in. If we want to visualize such data intuitively, we need to determine the path from data items of interest to the indirectly related spatial entities. Facete solves this problem with a automatic discovery of the most likely property path. Compared to other faceted browsers, Facete not only allows to constrain over facets of a particular set of entities, but over facets across full property paths. This allows to explore data more intuitively and in ways previously impossible.

In the remainder, we explain Facete's user interface in section 2, followed by a description of a use case scenario in section 3. In section 4 we briefly outline the conceptual background of Facete's main components. Subsequently, section 5 discusses related work. Finally, section 6 concludes and gives pointers to future work. Comprehensive accompanying material including the open-source code, screencasts and documentation can be found at `http://facete.aksw.org`. Under the umbrella of the *JAvascript Suite for Sparql Access (Jassa)*[5] project, we are working on making Facete's core components available as reusable library components, which developers may reuse and customize as needed.

[*]Corresponding Author
[1]`http://lod-cloud.net/`

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577022.

[2]`http://data.ordnancesurvey.co.uk`
[3]`http://semantic.eea.europa.eu/sparql`
[4]`http://www.w3.org/2003/01/geo/wgs84_pos`
[5]`https://github.com/GeoKnow/Jassa`

2. THE FACETE SPARQL EXPLORER

Facete is a Single Page Application (SPA) whose user interface comprises several UI components, which are depicted in Figure 1 and explained in the following. In the top area, there are elements that enable the user to select a SPARQL endpoint and chose from one or more of its contained named graphs. The main panel is divided into three columns containing a set of widgets with the following functionality: *1. Selection.* The first widget, labeled *Facet*, shows a facet tree that corresponds to the properties of all resources that match the set constraint. If there are no constraints, all resources that appear as a subject in the selected graphs are matched. Three actions can be performed for node in the facet tree: A click on the facet's name lists the facet's values in the *Facet Value* widget, where these values can be used for defining constraints. Clicking the caret symbol toggles the display of corresponding child facets. These are the properties of the the selected facet's values. Lastly, a facet can be pinned as a column to the *Table View*. Note, that the root of the facet tree is formed by a facet labelled *Item*. This facet's values correspond to the set of resources in subject positions of the selected RDF graphs. The *Facet Values* widget enables a user to paginate through a selected facet's values and optionally filter these values by a search term. Furthermore, clicking the checkbox next to a value creates a constraint. The *Filters* widget lists all active constraints. Individual constraints can be removed by clicking their entry, whereas the *Clear Filters* button purges them all.

2. Data. The middle area contains the *Table View*, which lists a table whose content is based on resources that match the active constraints and the facets that were pinned as columns. Columns can be sorted by clicking the caret icons. Multiple orders are supported by holding the shift key down while clicking.

3. Geographical. The *Geo-Link* drop down menu enables the user to choose from property paths connecting the resources that match the constraints with those that can be shown on the map. By default, the option *automatic* is enabled, which always picks the shortest path among the found ones. The *Map* widget displays markers corresponding to the selected resources and the geo-link. Blue boxes indicate areas that contain too many markers to be shown at once. These boxes disappear when sufficiently zoomed in. Clicking a marker shows its details in the *Detail View*. The *Detail View* shows the excerpt of the *Table View* that corresponds to the selected marker(s).

3. A SOPHISTICATED USE CASE

We demonstrate Facete in a use case scenario, which to the best of our knowledge, none of the existing SPARQL browsers can serve easily.

The dataset about *ICT research projects under EU-FP7* [6], for which an RDF version exists[7], contains comprehensive information about funded projects, partner organizations and particularly also how much money was granted to partners in FP7-ICT projects. The most important classes and their relations are as follows: Every *Project* is related to a set of *Fundings* (grants), whereas each instance represents the amount of money granted to a single project partner (beneficiary). Partners carry address information, which includes resources for the country and city. The cities' geo-coordinates were obtained based on interlinking them with places in *LinkedGeoData*[8] (LGD). This model is depicted in Figure 2.

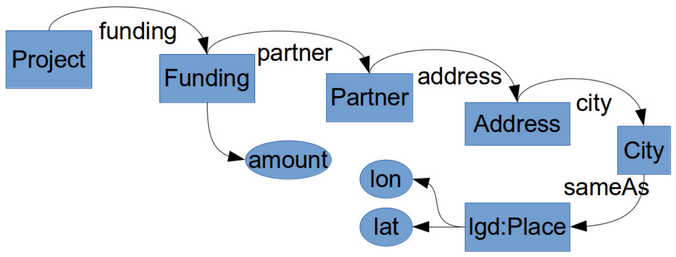


Figure 2: Excerpt of the CORDIS dataset model showing the path how projects are related to geo-coordinates.

There are several questions one may ask, such as: What are the projects related to a certain country? Where are the partners of a particular project located? How much money was granted to partners in a specific project?

With Facete, the following procedure can be performed to answer these questions:

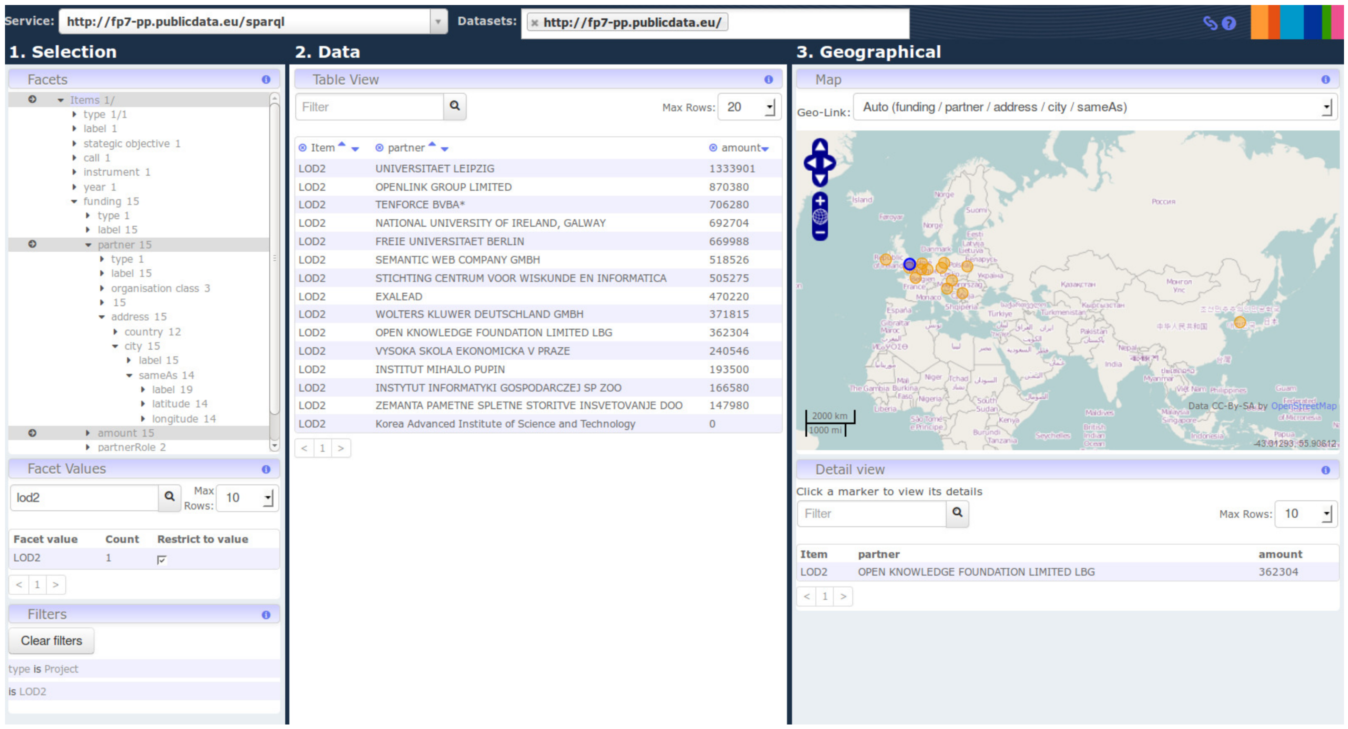
- First, select the *type* facet, and constrain it to the value *Project*. The main table will now only list project names, and the map component will automatically determine connections that link projects to coordinates.
- A project is related to a country if there is at least one project partner located in that country. Expanding the *funding* facet will list several nested facets, including *partner* and *amount*. We can then dive deeper into the facet tree by expanding first *partner* and then *address*. This reveals a *country* facet. Clicking it's label lists its corresponding facet values, which can then be constrained to a country of interest, such as “Germany”. The main table, as well as the map, will update to only show projects related to this country.
- A project can then be singled out by clicking the label of the *Item* facet and constraining it to the resource of a specific project, such as *LOD2*. The map view now only shows those partners that are located in the previously selected country.
- So far, the main table only listed names of projects. Pinning the *partner* and *amount* facets to the main table shows this project's related information. Drop the constraint on countries for the map to show partners world wide.

Note, that this use case stands exemplary for a large number of similar use cases involving spatial data.

4. CONCEPTS

In this section we briefly outline the key concepts used in Facete, which are related to faceted search, detection of property paths that connect concepts and dealing with large amounts of spatial data.

[6]`http://open-data.europa.eu/en/data/dataset/ict-research-projects-under-eu-fp7`
[7]`http://fp7-pp.publicdata.eu`
[8]`http://linkedgeodata.org`


Service: http://fp7-pp.publicdata.eu/sparql Datasets: http://fp7-pp.publicdata.eu/

1. Selection

Facets

```
Items 1/
  type 1/1
  label 1
  stategic objective 1
  call 1
  instrument 1
  year 1
  funding 15
    type 1
    label 15
    partner 15
      type 1
      label 15
      organisation class 3
      15
      address 15
        country 12
        city 15
          label 15
          sameAs 14
            label 19
            latitude 14
            longitude 14
    amount 15
    partnerRole 2
```

Facet Values

lod2 — Max Rows: 10

Facet value	Count	Restrict to value
LOD2	1	✓

Filters

Clear filters

type is Project

is LOD2

2. Data

Table View

Filter — Max Rows: 20

Item	partner	amount
LOD2	UNIVERSITAET LEIPZIG	1333901
LOD2	OPENLINK GROUP LIMITED	870380
LOD2	TENFORCE BVBA*	706280
LOD2	NATIONAL UNIVERSITY OF IRELAND, GALWAY	692704
LOD2	FREIE UNIVERSITAET BERLIN	669988
LOD2	SEMANTIC WEB COMPANY GMBH	518526
LOD2	STICHTING CENTRUM VOOR WISKUNDE EN INFORMATICA	505275
LOD2	EXALEAD	470220
LOD2	WOLTERS KLUWER DEUTSCHLAND GMBH	371815
LOD2	OPEN KNOWLEDGE FOUNDATION LIMITED LBG	362304
LOD2	VYSOKA SKOLA EKONOMICKA V PRAZE	240546
LOD2	INSTITUT MIHAJLO PUPIN	193500
LOD2	INSTYTUT INFORMATYKI GOSPODARCZEJ SP ZOO	166580
LOD2	ZEMANTA PAMETNE SPLETNE STORITVE INSVETOVANJE DOO	147980
LOD2	Korea Advanced Institute of Science and Technology	0

3. Geographical

Map

Geo-Link: Auto (funding / partner / address / city / sameAs)

Data CC-By-SA by OpenStreetMap

Detail view

Click a marker to view its details

Filter — Max Rows: 10

Item	partner	amount
LOD2	OPEN KNOWLEDGE FOUNDATION LIMITED LBG	362304

Figure 1: Screenshot of Facete showing information from the CORDIS dataset about FP7 research funding.

4.1 Facted Search

There are several systems that support faceted filtering of a set of RDF resources by immediate (possibly predefined) properties. However, faceted search over indirectly related properties is much more challenging. Facete's approach is based on the following key concepts:

- A *SPARQL concept* is a pair comprising a SPARQL graph pattern and a variable thereof. As such, it intentionally describes a set of resources. For instance, the pair *({?s a Project}, ?s)* describes the set of project resources. SPARQL concepts are the key enabler for indirect faceted search. Internally, Facete uses them to represent *any* set of resources, i.e. the set of facets, the set of child facets, the set of facet values and the set of resources with geometric information.
- *Property Steps* are used to navigate from a set of resources to a related set of resources by following a specific property. A direction attribute determines whether to follow that property in forward or inverse direction. Hence, a destination SPARQL concept can be obtained from a given origin SPARQL concept and a property step.
- A *Property Path* is a sequence of property steps.
- *Constraint Specifications* express constraints via references to property paths. Constraint specifications are internally translated to corresponding SPARQL graph patterns.

4.2 Finding Connections between Concepts

Due to data modeling, the spatial dimension is in most cases not directly attached to instances of a certain type. In order to visualize the spatial dimension of such objects efficiently and intuitively we need an approach to find connecting property paths between two arbitrary SPARQL concepts efficiently. As our use case demonstrates, these paths can become relatively long, and naive path discovery approaches are not feasible. Our approach is outlined as follows: Because we are only interested in the detection of property paths, we pre-compute a *property join summary.* The basic SPARQL query for this purpose is:

```
CONSTRUCT { ?p1 :joinsWith ?p2 } {
  { SELECT DISTINCT ?p1 ?p2 {
      ?a ?p1 [ ?p2 ?b ]
} }
```

Conceptually, we could search for arbitrary complex paths, such as ones that include cyclic (same property followed multiple times in the same direction) and zig-zag (forward and backward on the same property traversals. However, for our use cases the restriction to directed acyclic paths leading from a *source concept* to a *target concept* was sufficient: We query the source concept for all of its properties *?p*, and conceptually add triples *(:source :joinsWith ?p)* to the join summary. Thereby *:source* is a distinguished resource representing the source concept. From a target concept's graph pattern, such as *(?s geo:long ?x ; geo:lat ?y, ?s)*, we can infer that we need to search for properties that according to the join summary are connected to both geo:long and geo:lat. As a consequence, we can query the join summary for a set of candidate target properties using:

```
SELECT DISTINCT ?p {
  ?p :joinsWith geo:long , geo:lat
}
```

If the extensions of the source and target concepts have resources in common, this query's result set includes *:source* as a candidate.

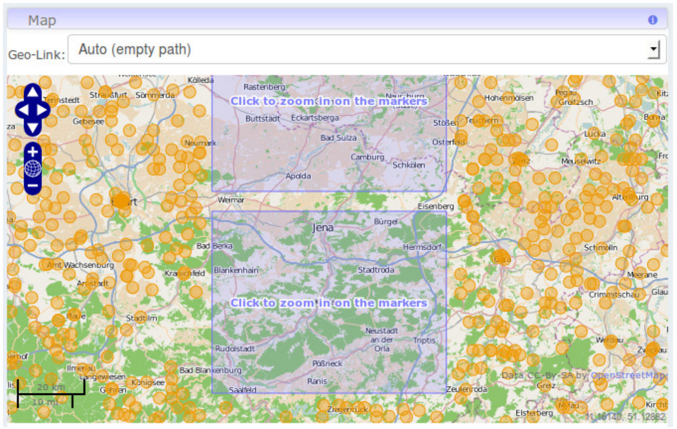

Map
Geo-Link:
Auto (empty path)
Click to zoom in on the markers
Click to zoom in on the markers
Jena

Figure 3: Display of Freebase instances in Germany.

We can now search for candidate paths on the join summary that connect *:source* with each of the candidate properties. For each candidate path we then fire an ASK query to check whether the given dataset contains an instance of it. Those paths that actually exist, are then listed in Facete's Geo-Link drop down box.

Note, that this approach is independent of any concrete vocabulary.

4.3 Display of large Amounts of Geometries

Some spatial RDF datasets, such as DBpedia or Freebase, contain significantly more spatial information than what can performance and bandwidth wise be reasonably retrieved and displayed on a map in a web application. Facete handles such cases using a quad tree data structure:

- Based on the users constraints on the facets and the geo-link, a corresponding SPARQL concept, named *geo-concept*, is created. The geo-concept specifies the set of resources to be shown on the map.
- A count of the number of instances matching the geo-concept is requested. If the count is below a configured threshold, all instances are retrieved at once and placed into the root node of the quad tree.
- If this count exceeds the threshold, the extent of the whole map is split recursively into four tiles of equal size. The recursion stops if either a maximum depth is reached, or if the tiles have reached a certain relative size when compared to the map viewport (e.g. when about 4x4 tiles are visible). For each tile, the geo-concept is then modified to only refer to resources within that tiles' bounding box. A tile's resources are only retrieved, if the new count is again below a configured threshold.
- Tiles that still contain too many geometries are rendered as boxes on the map.

An example of such display is shown in Figure 3, which shows a subset of the approx. 20.000 resources with geo-positions in Germany. For each set of constraints, Facete creates a new quad tree that acts as a cache for the user's current configuration.

5. RELATED WORK

The *RelFinder* system [2] is capable of finding property paths connecting a pair of *resources*, whereas Facete searches for paths between SPARQL *concepts*. Over the past decade, faceted search has become omnipresent, such as in web shop and content management systems. *Apache Solr*[9] is a popular system that offers extensive faceted search features, however, it does not offer native SPARQL support and thus requires pre-processing of RDF data. *Rhizomer* [1] and the *Virtuoso faceted browser*[10] support changing the focus from one set of resources to a related one (known as pivoting). However, with these systems, the user actually navigates between related list views of resources, whereas in Facete the user pins facets as new columns to the table view.

6. CONCLUSIONS AND OUTLOOK

In this demo we present *Facete*, an advanced faceted browser for SPARQL accessible data with geo-spatial capabilities. We demonstrate Facete's potential based on a sophisticated use case which requires retrieval of information from a set of indirectly related resources. Users are empowered to create customized tabular data views showing attributes they are interested in. Based on related geometric information, these data are also shown on a map. Facete's three key concepts are: (1) The generic faceted filtering engine, that supports showing and filtering by nested facets, (2) the property path detection, which efficiently detects how resources can be shown on the map based on indirectly related spatial information, and (3) how the system offers an interactive map display even for large amounts of geometric information.

There are a number of directions for future work: One is adding support for aggregation functions. Another logical step is to define more advanced views than the current table and map. Users should then be empowered to connect such views to their created data table. For instance, once a user has defined a table which lists for each city or country the total amount of money given to partners located in it, then based on this information a heat map view could be rendered. Our vision is, for the creation of visualizations from Linked Data to become as simple as in spreadsheet applications, where advanced chart and map visualizations can be created with a few clicks.

7. ACKNOWLEDGMENT

This work was supported by grants from the EU's 7th Framework Programme provided for the projects LOD2 (GA no. 257943) and GeoKnow (GA no. 611845).

8. REFERENCES

[1] J. M. Brunetti, R. Gil, and R. Garcia. Facets and pivoting for flexible and usable linked data exploration. In *Interacting with Linked Data Workshop (ILD)*, 2012.

[2] P. Heim, S. Hellmann, J. Lehmann, S. Lohmann, and T. Stegemann. RelFinder: Revealing relationships in RDF knowledge bases. In *3rd Int. Conf. on Semantic and Media Technologies (SAMT)*, volume 5887 of *LNCS*, pages 182–187. Springer, 2009.

[9] `http://lucene.apache.org/solr/`

[10] `http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtuosoFacetsWebService`