

# Mobile Semantic Geospatial Visualization and Exploration

Uroš Milošević\*, Claus Stadler\*\*

\* Institute Mihajlo Pupin, Belgrade, Serbia

\*\* Agile Knowledge Engineering Group, University of Leipzig, Leipzig Germany

\* uros.milosevic@pupin.rs, \*\* cstadler@informatik.uni-leipzig.de

**Abstract** — The work presented in this papers describes **GEM, the first cross-platform, mobile, semantic faceted geospatial browser that fully exploits the potential of the Linked Open Data paradigm. The tool is built on top of Jassa (Javascript Suite for Sparql Access) and offers a rich Web of Data experience by rising above the common mobile geospatial visualization limitations by relying on open, crowd-sourced and semantically linked information found in publicly available sources, such as the LOD Cloud. This information is loaded and filtered according to user’s needs, on demand, in order to prevent maps from overpopulating. Finally, special attention is paid to client-side optimization to deliver both acceptable performance and comfortable user/visual experience.**

## I. INTRODUCTION

The explosion of location aware technology has made the move of geographical information to their, perhaps, more natural setting, i.e. mobile devices, inevitable for any geospatial software striving to survive the demands of the ever-growing market. However, the functionality of the majority of available navigation systems is developed upon closed and proprietary solutions for both maps and software applications. Furthermore, such applications are unable to offer information specifically tailored to user’s needs, and cannot be extended by third parties. Although recent attempts propose ways of overcoming some of these barriers [1, 2, 3], none leverage the full power of the Linked Data paradigm.

The design and usability choices of desktop applications make them often hard or impossible to interact with on mobile devices due to both hardware (smaller screens, lower screen resolutions, lack of buttons, less processing power etc.) and software constraints. Therefore, most existing spatial-semantic visualization and exploration applications might prove impractical for a user on the go (e.g. in a car, on a bike, on foot etc.). Also, a mobile semantic browser designed with flexibility and extensibility in mind could pave the way for semantic authoring of information on the move, which could, in turn, enable crowdsourcing of geospatial information on the Web of Data.

Our mobile spatial-semantic visualization and exploration tool aims to complement the desktop experience through a rich mobile alternative that will exploit all strengths of Linked Data and further rise above the common mobile geospatial visualization limitations by

relying on open, crowdsourced and semantically linked information found in publicly available sources, such as the LOD Cloud. This information must be loaded and filtered according to user’s needs, on demand, in order to prevent maps from overpopulating. Moreover, in order to reach a larger target population and spark community engagement and contribution, such a browser needs to be able to run on at least one major mobile platform (Android, iOS, Windows Phone).

In Section 2, we will first look at Jassa, the foundation library we build on top of, and the basics of semantic faceted classification and navigation. Next, in Section 3, we explore the existing and tailor-made technologies that are needed to take the semantic faceted browsing over RDF datasets to handheld devices. Section 4 showcases our take at a mobile geospatial Linked Data exploration experience, and, finally, Section 5 discusses future work and concludes this paper.

## II. FOUNDATION

To provide the means to easily explore and visualize spatial content retrieved from the Web of Data, new information retrieval paradigms need to be explored. Apart from this, new techniques need to be developed to combine semantic information with geospatial data and display them with state-of-the-art map rendering tools. Moreover, to support the reuse of certain application units and separate different concerns, like browsing and presentation, a software implementing these paradigms and techniques should consist of distinguishable components.

With RDF, a data model was introduced, which enables global identification and integration of resources as well as cross-dataset interlinking. On top of RDF, SPARQL became a standard language for accessing Web databases. Yet, the development of Web applications for the exploration and visualization of SPARQL-accessible data still poses several challenges related to performance and design. The Open Source *JavaScript Suite for Sparql Access (Jassa)* [4] is motivated by the goal of creating the generic widgets for faceted browsing of RDF data. The library is designed to tackle many of the challenges encountered during the development process.

### A. Jassa

Jassa is an umbrella term for a set of three related projects. These projects are summarized as follows:

The research presented in this paper is partly financed by the European Union (FP7 GeoKnow, Pr.No: 318159), and partly by the Ministry of Science and Technological Development of the Republic of Serbia (SOFIA project, Pr. No: TR-32010).

- *Jassa Core*<sup>2</sup> is a project that provides a layered set of modules for: the representation of RDF and SPARQL, the execution of queries, SPARQL to JSON mapping, and most prominently, faceted search.
- *Jassa UI*<sup>3</sup> is a project for user interface components based on Jassa Core and the *AngularJS*<sup>4</sup> framework.
- *Complementary Services for Jassa*<sup>5</sup> is a Java project that offers server side APIs that enhance Jassa, such as a SPARQL cache proxy. Another service is capable of finding property paths connecting two sets of resources.

The Jassa code base can be used both on the client and the server side. The only difference between these settings lies in the dependencies that need to be included. Furthermore, Jassa provides several RDF and SPARQL foundation classes which are nearly identical to those of the excellent API of the Java-based *Apache Jena*<sup>6</sup> project. The rationale followed by Jassa is to exploit existing, well-known API designs, rather than to invent new ones. At present, Jassa comprises the following modules:

- *util*: A utility module. Contains collections, such as *HashMap* and *HashSet*.
- *rdf*: The module that holds core classes related to RDF.
- *vocab*: A module for vocabularies, expressed in terms of classes of the *rdf* module.
- *sparql*: A module for core classes related to SPARQL. Builds upon the prior modules.
- *service*: Abstraction layer for SPARQL endpoints.
- *facete*: A faceted search module.
- *sponate*: A SPARQL-to-JSON mapper. Particularly powerful in combination with the generation of web frameworks that offer a clean separation between DOM and application logic, such as AngularJS and Ember.js.

The Jassa core classes aim to serve as a solid foundation for JavaScript-based Semantic Web applications. The *rdf* module provides the *Node* class (through *rdf.NodeFactory*) for encapsulating RDF terms.

In contrast to approaches that are based on plain JSON, a Jassa node object provides methods such as *isLiteral()* and *getLiteralDatatypeUri()*. Furthermore, the *toString()* method is implemented to yield meaningful string representations. The *sparql* module contains several classes for the syntactic representation of SPARQL queries. An example showing a simple query for obtaining all DBpedia instances of type *Airport* is given in Listing 1. Note, that all mentioned namespaces reside in the global *jassa* object.

### B. Faceted geospatial browsing

The Oxford Dictionary defines a *facet* as ‘one side of something many-sided, especially of a cut gem’<sup>7</sup>. *Faceted classification* is an analytic-synthetic classification scheme which relies on multiple taxonomies to classify objects [5]. *Faceted browsing* builds upon faceted classification to enable navigation through information by applying multiple filters (i.e. *facets*). As far as Linked Data is concerned, tools such as *Sparklis*<sup>8</sup> and *Pelorus*<sup>9</sup> offer semantic faceted navigation over RDF datasets.

Jassa comes with a powerful SPARQL-based faceted search module in the *facete* namespace, which supports the definition of custom constraint types as well as constraining sets of resources by indirectly (possibly inversely) related properties. Unlike Sparklis and Pelorus, Jassa features support for nested and inverse properties and offers reusable components. Some recent development efforts which provide similar features are [6] and [7].

Faceted browsing widgets are implemented as AngularJS directives (Listing 2) and can thus be embedded into AngularJS applications using the corresponding HTML snippets. Note that the widgets are synchronized by AngularJS on the state of the *facetTreeConfig* object; any change will automatically trigger an update of the widgets.

*Facete*<sup>10</sup> uses the above described features of Jassa and offers out of the box faceted browsing over SPARQL end-points to ease the navigation of RDF data using

```

1  var s = rdf.NodeFactory.createVar( 's' );
2  var o = rdf.NodeFactory.createUri( 'http://dbpedia.org/ontology/Airport' );
3  var t = new rdf.Triple( s, vocab.rdf.type, o );
4
5  var query = new sparql.Query();
6  query.setResultStar( true );
7  query.setQueryPattern( new sparql.ElementTriplesBlock([ t ] ) );
8  query.setLimit( 10 );
9  console.log( 'As string: ' + query );
10 // Output : Select * { ?s a <http://dbpedia.org/ontology/Airport > } Limit 10

```

Listing 1. Forming a query with Jassa

<sup>2</sup> <https://github.com/GeoKnow/Jassa-Core>

<sup>3</sup> <https://github.com/GeoKnow/Jassa-UI-Angular>

<sup>4</sup> <https://angularjs.org/>

<sup>5</sup> <https://github.com/AKSW/jena-sparql-api>

<sup>6</sup> <http://jena.apache.org>

<sup>7</sup> <http://www.oxforddictionaries.com/definition/english/facet>

<sup>8</sup> <http://www.irisa.fr/LIS/ferre/sparklis/>

<sup>9</sup> <http://clarkparsia.com/pelorus>

<sup>10</sup> <https://github.com/GeoKnow/Facete2>

```

1 $scope.fctTreeConfig = new facete.FacetTreeConfig();
2 $scope.selectedPath = null; // Start with no selection
3 $scope.selectFacet = function( path ) { $scope.selectedPath = path; }

```

Listing 2. Showing values for a selected facet

advanced faceted search techniques. Spatial data is automatically detected and visualized on a map, even if the geometric information is only indirectly related to the resources specified by the faceted search.

*Mappify*<sup>11</sup> allows easy creation of embeddable map view snippets. It builds on reusable components of Facete2 and thus enables a facet based definition of points of interests based on a SPARQL accessible dataset. Users are enabled to quickly style the map display by choosing marker icons and defining templates for the content to show when clicking the markers.

### III. SPATIAL-SEMANTIC VISUALIZATION AND EXPLORATION ON MOBILE DEVICES

Our mobile spatial-semantic visualization and exploration tool is envisioned to give the user the ability to load the application with a custom dataset through one of the available SPARQL endpoints and retrieve the data they might be interested in on request, instead of (over)loading the mobile map with irrelevant information from the start.

Our goal is to deliver an easy to use, yet powerful mobile visualization and exploration tool that will provide a highly customizable and information-rich *slippy* map to the geospatial data consumers on the move. More specifically, the user interface should consist of:

- An interactive map component, to be used for quick and easy exploration of geographical areas, showing the user's own position (GPS coordinates) and the surrounding area;
- A semantic facet filtering component and a result view;
- A data source management component, to be used for quick loading and removing of visible information.

#### A. *Slippy map*

*Leaflet.js*<sup>12</sup> is an open-source JavaScript library for mobile-friendly interactive maps. It is extremely lightweight, yet has all the features we need for the task at hand. The library is designed with simplicity, performance and usability in mind. It works efficiently across all major desktop and mobile platforms, out of the box, taking advantage of HTML5 and CSS3 on modern browsers while still being accessible on older ones. It can also be extended with numerous plugins, and is an adequate lightweight replacement for the *OpenLayers*<sup>13</sup> library being used with Jassa for Facete2 and Mappify.

#### B. Platform

The identified requirements have narrowed down the choice of mobile development frameworks to *Apache Cordova*<sup>14</sup> / *Adobe Phonegap*<sup>15</sup> (formerly known as *Apache Callback*), which is an open-source set of device APIs that allow accessing native device function such as the camera or accelerometer from JavaScript. Combined with a UI framework such as *jQuery Mobile*<sup>16</sup> or *Dojo Mobile*<sup>17</sup> or *Sencha Touch*<sup>18</sup>, this allows a smartphone app to be developed with just HTML, CSS, and JavaScript, which in turn, allows us to build upon some of the above mentioned existing work on desktop geospatial browsers, while focusing on the mobile user experience.

Since these JavaScript APIs are consistent across multiple device platforms and built on web standards, our solution is expected to be portable to other device platforms with minimal to no changes (Cordova is available for: iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada, and Symbian). This mobile development framework provides a set of uniform JavaScript libraries that can be invoked, with device-specific native backing code for those libraries. Moreover, the app would still be packaged using the platform SDKs and could be made available for installation from each device's application store (e.g. Google Play, Apple App Store, Windows Phone Store etc.).

### IV. GEM

Our approach to the problem of putting the power of the above components in the hands of the mobile data consumers, titled GEM (Geospatial-semantic Exploration on the Move), relies on the previously mentioned, tailor-made and open-source Web technologies that are optimized for handheld devices.

The design philosophy behind GEM aims at maximizing the usable application/screen area through one stationary design component, and four on-demand widgets. The slippy map component represents the base layer on top of which the control widgets appear on request. The four components are given in Figures 1 and 2, and their functionalities described below:

- The *Facets* (left-hand) side drawer holds the loaded resource facet tree;
- The *Source Manager* (right-hand) side drawer is used to add/edit/remove available SPARQL endpoints / Linked Open Data sources;

<sup>14</sup> <http://cordova.apache.org/>

<sup>15</sup> <http://phonegap.com/>

<sup>16</sup> <http://jquerymobile.com/>

<sup>17</sup> <http://dojotoolkit.org/features/mobile>

<sup>18</sup> <http://www.sencha.com/products/touch/>

<sup>11</sup> <https://github.com/GeoKnow/Mappify>

<sup>12</sup> <http://leafletjs.com/>

<sup>13</sup> <http://openlayers.org/two/>

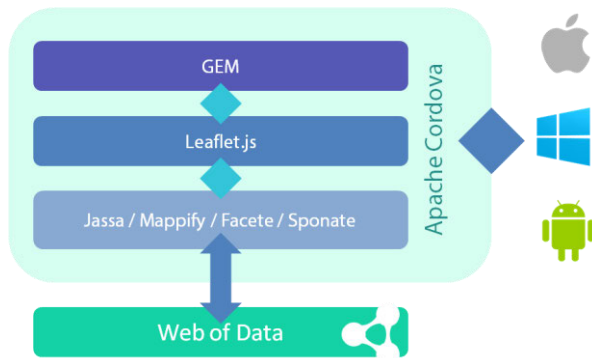


Figure 1. GEM technology stack

- The *Filter* text box (at the top), as the name suggests, is used to filter the resources on screen. The widget also holds the orientation indicator (i.e. compass);
- The *Resource details* bottom drawer pops-up to display the relevant information (e.g. label, URI and related triples) for the selected feature.

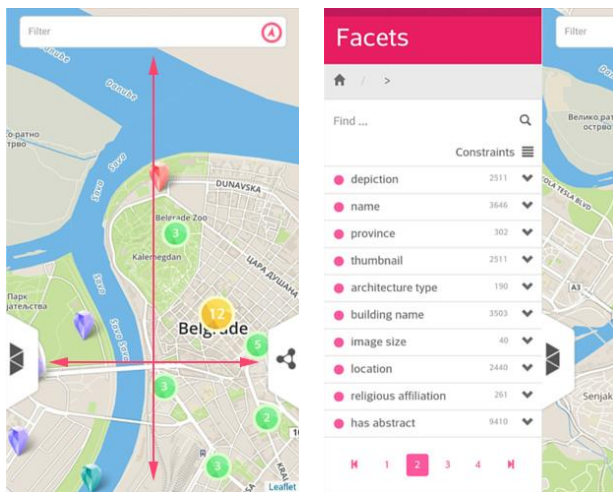


Figure 1. a) User interface; b) Facet tree side drawer

Different icon colors are used to indicate different resources / information sources. Moreover, to further prevent the map from overpopulating, and avoid overloading the mobile device resources in situations where multiple features are in each other's vicinity (relative to the map zoom level), we resort to marker clustering, i.e. grouping. Marker cluster groups are indicated by circles with resource counts. The color of the circle depends on the number of resources being grouped (ranging from green to red; green indicating small groups; Figure 1.a).

#### A. Data source management

The information shown on the screen is easily controlled using the Source Manager widget, which lets the user specify the name of the source (for convenience), the corresponding SPARQL endpoint and graph, as well as the desired resource type to be retrieved / displayed on the map. As Cordova relies on the host device's built-in Internet browser facilities, we exploit the browser's

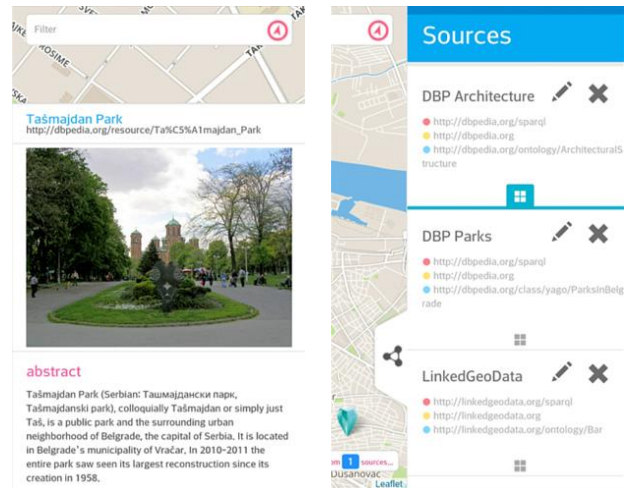


Figure 2. a) Resource details; b) Source management side drawer

HTML5 Local Storage to maintain the user preferences across sessions (i.e. automatically load the last used resource list the next time the application starts). Before HTML5, application data had to be stored in cookies, included in every server request. Local storage is more secure, and large amounts of various data can be stored locally, without affecting website/application performance.

#### B. Usage

We hereby outline the essential functionalities of the mobile geospatial-semantic browser provided by the above described facilities, and the basic steps to using them:

- The first time the application starts, GEM automatically populates the list of available LOD sources with DBpedia and LinkedGeoData SPARQL endpoints, using architectural sites, parks and bars as Base concepts (this is currently for demonstration purposes (e.g. a tourism-focused scenario) and might change in the future). The user can easily change this, by either editing the existing source list, or by creating a new one (i.e. by adding new endpoints / data types and/or removing the default ones; Figure 2.b).
- On startup, if the host OS location services are available (e.g. GPS), the map will zoom in on the user's own location (indicated by a gold pin), and the orientation indicator / compass in the filter widget will show (and update, based on) the direction the user's device is pointing to.
- Browsing the map will load the corresponding resources for the visible map area.
- To further make best use of the available screen space, upon detecting map interaction / movement, GEM will hide away the GUI elements the user cannot interact with while browsing through the map.
- Additionally, the application can also work in landscape mode.
- Should the user decide to narrow down their search, they can do so either by using the Filter box, or by using the Facet tree side drawer (Figure 1.b), which functions much in the way the Facete2 facet tree does.

The user first picks the facets (i.e. properties), and then the values to restrict the results to only those that fit the given description.

- Finally, to retrieve the details related to a single resource, the user only has to select the corresponding marker on the map. The bottom drawer reveals the feature's name (i.e. label) and URI. Clicking on the resource label expands the bottom drawer to display other relevant attributes (i.e. corresponding triples; Figure 2.a).

## V. CONCLUSION

The GEM prototype described above represents a one-of-a-kind mobile faceted geospatial-semantic browser for Linked Open Data. It builds on top of the efforts already invested in state of the art open source desktop solutions and technologies, ranging from Jassa, Facete2 and Mappify to well-known third-party, community-maintained frameworks and libraries, such as AngularJS and Leaflet.js. It is deployed using Apache Cordova / Adobe Phonegap, allowing single-branch development for multiple major mobile platforms.

The current state of affairs opens up opportunities for future efforts, starting with backend improvements that will enable importing data based on other coordinate systems, allowing for richer and more complex data integrations. Moreover, an authoring component would give an entirely new dimension to the two-way interaction between the user and the application, and hopefully make a push towards wide community acceptance through crowdsourcing. Finally, further user experience improvements will be considered as well,

such as automatic (re)source recommendations and tighter integration with Mappify, enabling easy sharing of information between the desktop and mobile clients, and making GEM an all-in-one solution for geospatial exploration on the move.

## REFERENCES

- [1] C. Becker and C. Bizer, *Exploring the Geospatial Semantic Web with DBpedia Mobile*. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, vol. 7, pp. 278–286, 2009.
- [2] C. J. Van Aart, B. J. Wielinga and W. R. van Hage, *Mobile Cultural Heritage Guide: Location-Aware Semantic Search*. In 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010), vol. 6385 of Lecture Notes in Computer Science, pp. 257–271, 2010.
- [3] M. Ruta, F. Scioscia, S. Ieva, G. Loseto, E. Di Sciascio, *Semantic Annotation of OpenStreetMap Points of Interest for Mobile Discovery and Navigation*. Proceedings of the 2012 IEEE First International Conference on Mobile Services, IEEE Computer Society, 2012.
- [4] C. Stadler, P. Westphal and J. Lehmann, *Jassa - A JavaScript Suite for SPARQL-based Faceted Search*. Proceedings of the ISWC Developers Workshop 2014, co-located with the 13th International Semantic Web Conference, pp. 31–36, October 2014.
- [5] W. Gödert, *Facet classification in online retrieval*. International Classification, Vol. 18 No. 2, pp. 98–109, 1991.
- [6] M. Arenas, B. Cuenca Grau, E. Kharlamov, S. Marciuska, D. Zheleznyakov, and E. Jimenez-Ruiz, *Semfacet: Semantic faceted search over Yago*. In 23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, Companion Volume, pp. 123–126, 2014.
- [7] H. Bast, F. Bärle, B. Buchhold, and E. Haußmann, *Easy access to the freebase dataset*. In 23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, Companion Volume, pp. 95–98, 2014.