RDF2Map Library

Luis Daniel Fernandes Rotger, Ana Cristina Trillos Ujueta, Jaime Manuel Trillos Ujueta

University of Bonn
s6lufern@uni-bonn.de, trillos@informatik.uni-bonn.de,
trillosj@informatik.uni-bonn.de

Abstract. The Web provides a large number of geospatial information. Recently this information can be represented using Resource Description Framework (RDF), which allow to interlink data. However these geospatial data must be processed before being displayed on a map. In this paper, we will present RDF2Map library, a developing tool that will help developers to automatically display geospatial data in a map.

1 Introduction

One of the main problems developers encounter when using linked Data technologies, lies in the fact that the tools provided to encourage and help development processes are still restricted. Nevertheless, linked Data and Semantic Web provide powerful properties that could be greatly valuable for a wide range of use cases; e.g., it is quite helpful when dealing with distributed data that are linked to each other, it maintains the veracity of each concept since they are unique in the Web, and it is universally accessible [3]. These are few examples that could improve almost any application or software product development, specially for geo-spatial software. Rather than having a big dataset that specifies geo-spatial resources with all the properties and respective values; it would be much easier to have only the identifier (URI) of this concepts and look their geo-spatial properties (latitude, longitude) into Linked Open datasets like DBpedia[1] [1].

Currently there exist approaches to overcome this kind of situations. As for applications for displaying RDF concepts enriched with geospatial properties, Map4RDF [7] is a tool for faceted exploring, visualizing and interacting with RDF geospatial datasets. QMap application [5] is able to extract information from HTML selected datasources, create RDF instances using wrappers and RDF Schemas, and with a SPARQL engine extract relevant data to be displayed on a map. LinkedGeoData Browser [2] allows to browse the world, analyze nodes and create facets to allow filtering. Each facet will display related elements on the map. On the other hand, developing tools that help developers display their RDF datasets on a Web page, there is no evidence of an already developed,

[1] https://dbpedia.org/sparql

working and tested library to do so.

RDF2Map library focus on the problem of a developer that wants to make use of linked Data to provide to his final user an interface with a map displaying the representation of geo-spatial concepts, using an RDF file written in Turtle format. RDF2Map is very straightforward to use, as importing a JavaScript library inside a HTML Web page. The library is flexible in the sense that it allows the developer to create his own RDF geo-spatial concepts representations in a Turtle file, taking into account that these concepts do not necessarily exists on the Semantic Web; and the library will display them in the map.

The paper is structured in 6 sections. Section 2 presents concepts for understanding the problem. In Section 3, we cover an overview of related work and then we present RDF2Map library, its implementation and use cases, in Sections 4 and 5; finally in Section 6, conclusions and future work are presented.

2 Background

Semantic Web [4] is a technology that allows sharing data. It offers a common format, i.e. RDF, for integrating data taken from different sources, as well as a language for describing how the data is related. RDF (Resource Description Framework)[2] is the standard model used in Semantic Web, where data and their relationships are represented by means of triples. Data that is published on the Web requires an URI (Uniform Resource Identifier) [6], an Unicode string of characters. Each geospatial concept is described using geospatial vocabularies; Basic Geo Vocabulary[3] offers properties `geo:lat` and `geo:long`, for denoting latitude and longitude of a geospatial resource; it can also be used more general vocabularies like, Friend of a Friend[4], that has properties such as `foaf:name` to denote the name, and `foaf:homepage` to denote the Web page link of a resource.

3 Related Work

There are few approaches that focus on representation of RDF geospatial resources on a map. One of these is Map4RDF [7], an application that allows the user to search facets and visualize geospatial information from a selected dataset. The backend of the application, the Web Server, connects to a given triplestore through the SPARQL endpoint for retrieving instances of the chosen facet. The faceted browsing interface gathers the information and displays the points on a map. In contrast to RDF2Map library, it is an application, and there are no signs that it can be integrated into an HTML Web page, for displaying relevant data to a developer's client.

[2] `https://www.w3.org/RDF/`
[3] `https://www.w3.org/2003/01/geo/`
[4] `http://www.foaf-project.org/`

Another approach is LinkedGeoData Browser [2], an application for browsing and editing LinkedGeoData[5] points. It behaves similarly to Map4RDF, however LinkedGeoData Browser displays markers and polygons on the map; Map4RDF claims that it displays, LineStrings, Polygons, Markers, etc.
In contrast to these two applications, QMap [5] claims that information is extracted from HTML data sources and not from RDF datasets. The information is translated into RDF instaces using wrappers and RDF Schemas. This RDF instances are queried by a SPARQL engine providing the query results and finally being displayed as markers in a map.

As can be seen, the problem of having a library that will help developers to automatically display RDF geospatial information on a map, without further processing, is still an open issue.

4 Proposed Approach

In this section, we describe our solution. RDF2Map library aims to help developers to illustrate/display RDF geospatial information on a map, without further processing the data.

4.1 Problem Statement and Proposed Solution

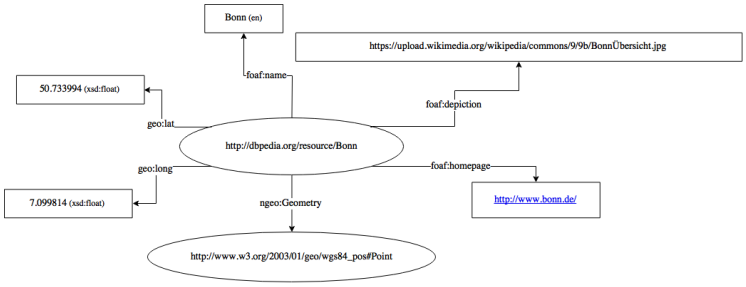
After evaluating related work, we noticed there is a lack in libraries for processing and displaying RDF geospatial concepts in a map. For this reason, we developed RDF2Map library[6]. A JavaScript library for processing geospatial concepts, from a Turtle file, and displaying them in a map.

RDF2Map works with two open-source JavaScript libraries, Leaflet[7] for interactive maps and RDFStore[8] for processing information. RDFStore supports SPARQL querying version 1.0 and data manipulation.
One of the main features of RDF2Map is the capability to extract remote information from the DBPedia SPARQL Endpoint, using the URI of a geospatial concept, and then with the extracted coordinates (latitude and longitude) display them in the map. These information can be displayed in different shapes and forms, due to Leaflet allows working with markers, polylines, polygons and pop-ups.

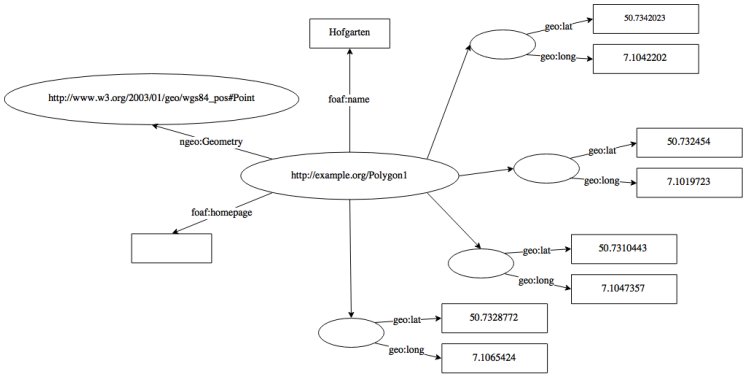
4.2 Implementation

RDF2Map library uses a subset of RDF properties, described on two data models. Figure 1a shows the data model for representing geospatial concepts dis-

[5] `http://linkedgeodata.org/Datasets`
[6] `https://github.com/atrillos/Rdf2Map_library`
[7] `http://leafletjs.com`
[8] `https://github.com/antoniogarrote/rdfstore-js`



(a) RDF2Map library data model.



(b) RDF2Map library data model for Polygons and Polylines.

Fig. 1. RDF2Map data models.

played as markers and icons. And figure 1b represents the data model for polygons and polylines geospatial concepts.
RDF2Map library receives a Turtle file with all geospatial concepts to be shown in the map. The library creates a store using RDFStore, where the concepts will be stored; then the concepts are extracted and those which data can be requested remotely, will be requested and loaded into the store, consequently markers, icons, paths and/or polygons will be generated. The map will be updated and all geospatial information will be displayed. As you can see in figure 2, the activity diagram.

Once RDF2Map finishes processing all geospatial concepts from the file, the map will be updated presenting the geospatial concepts, see figure 3a for an example. RDF2Map library, also offers clustering only for two types of geospatial concepts, markers and icons. Figure 3b is an example of how clustering will be shown on the map. The final user can interact with the markers, icons, polygons and paths; when selected, a pop-up will appear displaying name and homepage for each concept.
When a geospatial concept is described as icon, it will be shown in the map with the image extracted using the property `foaf:depiction`, as shown in figure 3a.

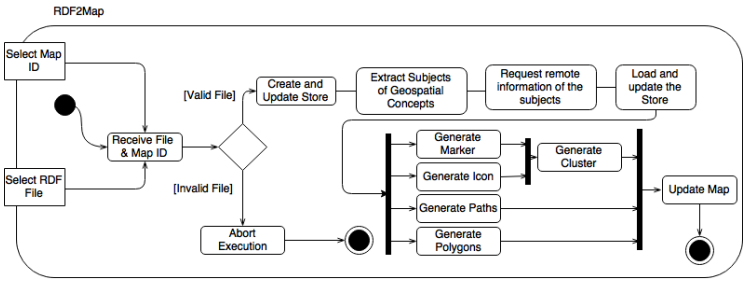
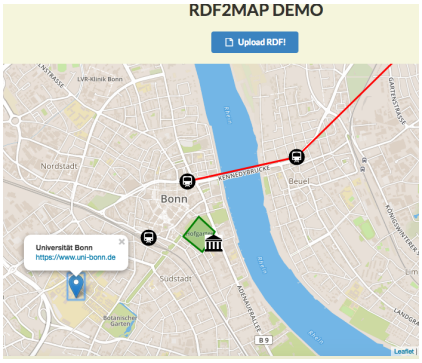
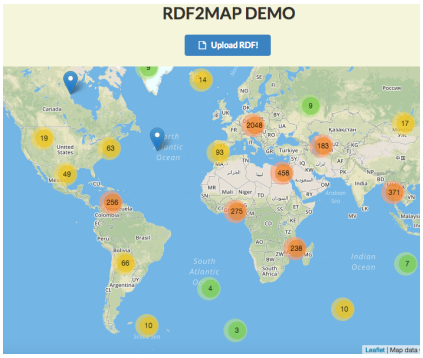


Fig. 2. Activity diagram of RDF2Map library.



(a) Sample of map displaying markers, icons, polygons and paths.



(b) Sample of clustering working on a map.

Fig. 3. Examples.

For polygons and paths, the property `dbc:Color` describe a specific color to personalize them; in case this property is not specified the Leaflet default color will be used.

RDF2Map was tested to evaluate its performance when receiving a file with different quantity of input concepts. The tests were performed on a laptop with Intel Corei7 6700HQ processor and 4GB RAM.
Table 1 shows the times (in millisecond) that each process performed by RDF2Map takes. *loadTime* is the time that the RDFStore takes to load the data from the turtle file. *requestTime*, represents the time that the library takes when requesting information remotely; and *ProcessingTime* is the time, RDF2Map takes to process and display all geospatial concepts (markers, icons, polygons and paths) in the map. The figure 4 represents the execution relative times for the test.

# Concepts in file	loadTime	requestTime	ProcessingTime	# Concepts displayed
10	35.95	474.78	25.835	10
100	175.395	610.33	60.449	100
1000	1369.775	1854.108	271.885	1000
2000	2936.82	2598.87	572.34	2000
4000	6030.48	2650.354	1158.064	4000
8000	12512.215	4383.68	2812.709	8000
10000	14772.42	6123.982	2937.875	10000

Table 1. Performance test results in ms.

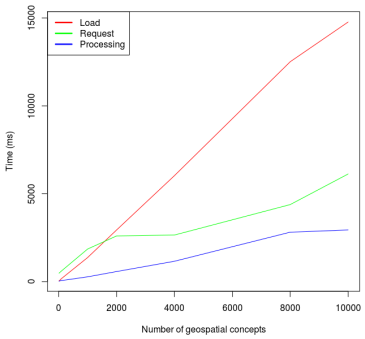


Fig. 4. Activity diagram of RDF2Map library.

5 Use Cases

RDF2Map library can be used downloading `rdf2map.js` from RDF2Map on GitHub, and importing it into an HTML file; at the same time it is important to download and import RDFStore and Leaflet libraries, as shown bellow.

```
<script type="text/javascript" src="resources/rdfstore.js"></
    script>
<script type="text/javascript" src="resources/leaflet.js"></
    script>
<link rel="stylesheet" type="text/css" href="resources/
    leaflet.css">
<link rel="stylesheet" type="text/css" href="resources/styles
    .css">
<link rel="stylesheet" type="text/css" href="resources/
    MarkerCluster.css">
<link rel="stylesheet" type="text/css" href="resources/
    MarkerCluster.Default.css">
<script type="text/javascript" src="resources/leaflet.
    markercluster.js"></script>
<script type="text/javascript" src="resources/rdf2map.js"></
    script>
```

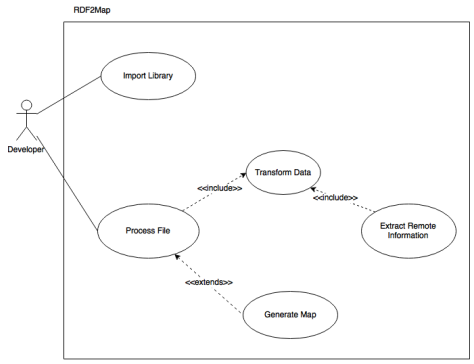


Fig. 5. Use case diagram for RDF2Map library.

The general use case of RDF2Map (Figure 5), a user imports the library and then calls RDF2Map function `bindFileInput` with two parameters, the input file and the map id. Next, we will describe two possible use cases for using RDF2Map. The first use case is related for a a specific company and the second is for didactic use.

5.1 Use Case 1:

A company (e.g. Rewe) desires to inform its users the location of their stores on their Web page, but this information is not published on the Web.
The developer creates a RDF file, with the geospatial information for each store, as shown below.

```
@prefix ns0:    <http://geovocab.org/geometry#> .
@prefix ns1:    <http://linkedgeodata.org/triplify/> .
@prefix ns2:    <http://linkedgeodata.org/ontology/> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .
@prefix xsd:    <http://www.w3.org/2001/XMLSchema#> .
@prefix geo:    <http://www.w3.org/2003/01/geo/wgs84_pos#> .
ns1:node1329330946      ns0:Geometry    ns2:Icon ;
foaf:homepage   "https://www.rewe.de/" ;
foaf:depiction  "https://i.imgur.com/yH6LWPn.png" ;
geo:lat 53.133327100000002474 ;
geo:long        8.1904960000000013309 ;
foaf:name       "REWE" .
ns1:node2545718183      ns0:Geometry    ns2:Icon ;
foaf:homepage   "https://www.rewe.de/" ;
foaf:depiction  "https://i.imgur.com/yH6LWPn.png" ;
geo:lat 48.116935300000001519 ;
geo:long        11.525668500000000094 ;
foaf:name       "REWE" .
ns1:node1574280669      ns0:Geometry    ns2:Icon ;
foaf:homepage   "https://www.rewe.de/" ;
```

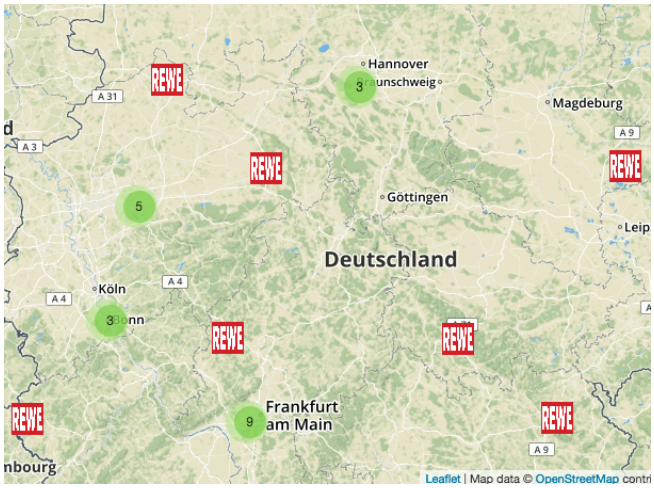


Fig. 6. Map for Use Case 1.

```
foaf:depiction  "https://i.imgur.com/yH6LWPn.png" ;
geo:lat 50.100445400000005236 ;
geo:long        8.6584660000000006619 ;
foaf:name       "REWE" .
...
```

Then the developer imports RDF2Map, RDFStore and Leaflet libraries into his HTML file, and calls `RDF2Map.bindFileInput`. Every time the company's Web page is open, the map will display the stores (Figure 6)

5.2 Use Case 2:

An institute wants to teach their students with an interactive map, all the countries that exist and have existed; so a developer creates an RDF file, with the URIs of all the countries from the DBPedia Endpoint, as shown below.

```
@prefix ns0:    <http://geovocab.org/geometry#> .
@prefix dbr:    <http://dbpedia.org/resource/> .
@prefix geo:    <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
dbr:Abbasid_Caliphate   ns0:Geometry    geo:Point .
dbr:Almohad_Caliphate   ns0:Geometry    geo:Point .
dbr:Arab_League ns0:Geometry    geo:Point .
dbr:Cape_Colony ns0:Geometry    geo:Point .
dbr:Central_Tibetan_Administration      ns0:Geometry    geo:
    Point .
dbr:Dacia       ns0:Geometry    geo:Point .
dbr:Democratic_Republic_of_Afghanistan  ns0:Geometry    geo:
    Point .
```

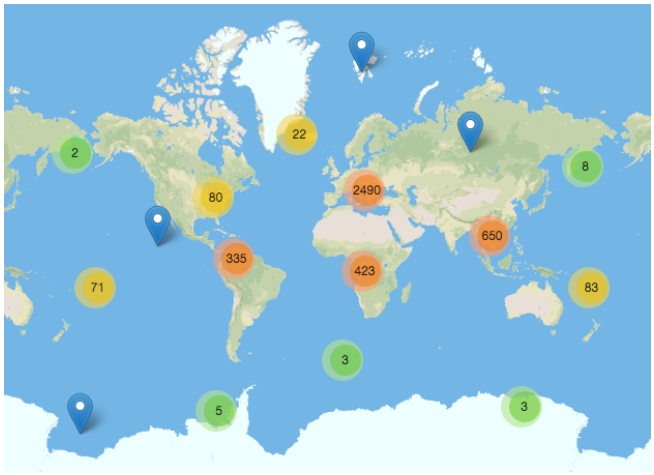


Fig. 7. Map for Use Case 2.

```
dbr:Duchy_of_Lorraine    ns0:Geometry     geo:Point .
dbr:Duchy_of_Mecklenburg-Strelitz        ns0:Geometry     geo:
    Point .
dbr:Fatimid_Caliphate    ns0:Geometry     geo:Point .
...
```

Later the developer, imports RDF2Map, RDFStore and Leaflet libraries into the HTML file and the information will be displayed. Since there have existed a lot of countries that have been divided or unified during history, the map will be clustered as shown in figure 7.

In both use cases the implementation using RDF2Map in the developer project is simple. The performance of the library for these cases takes some time (milliseconds) to process files containing more than 1000 geospatial concepts but RDF2Map ensures that all the concepts which are in the Turtle file has been inserted in the map.

6 Conclusions and Future Work

RDF2Map library is probably one of the first attempt to offer a library for displaying geospatial concepts on a map. It offers features like the ability to extract data remotely from the Web, or locally from a Turtle file and showing the results in a map. RDF2Map permits adding geo-points represented as markers, icons, polygons and or lines (paths).

RDF2Map library, currently is limited to accept files written in Turtle format, and extract information from one source, DBPedia endpoint. Thus in the future, we plan to extend the library to include multiple SPARQL endpoints, as well as, accepting other RDF serializations formats (e.g. N3, RDFa, NQuads, etc.). At the same time, we plan to make RDF2Map more flexible so that through the

API, the developer can choose which information about the geospatial concepts wants to obtain and display.

References

1. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735, 2007.
2. Sören Auer and Jens Lehmann. Linkedgeodata–collaboratively created geo-information for the semantic web. *Semantic Web Challenge, ISWC*, 2009.
3. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227, 2009.
4. I Herman. The semantic web home page. `http://www.w3.org/2001/sw/`, 2006.
5. Frederik Hogenboom, Alexander Hogenboom, Ricardo van Gelder, Viorel Milea, Flavius Frasincar, and Uzay Kaymak. Qmap: an rdf-based queryable world map. In *Third international conference on knowledge management in organizations (KMO 2008)*, pages 99–110. Vaasan Yliopiston Julkaisuja, 2008.
6. J. J Klyne, G. Carroll. Rdf semantics. `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`, 2004.
7. Alexander de Leon, Filip Wisniewki, Boris Villazón-Terrazas, and Oscar Corcho. Map4rdf-faceted browser for geospatial datasets. 2012.