Eurographics Conference on Visualization (EuroVis), Posters Track (2016)
T. Isenberg and F. Sadlo (Guest Editors)

Styrian Diversity Visualisation: Visualising Statistical Open Data with a Lean Web App and Data Server

K. Andrews[1], T. Traunmüller[1], T. Wolkinger[2], E. Goldgruber[2], R. Gutounig[2] and J. Ausserhofer[2]

[1]IICM, Graz University of Technology, Austria
[2]FH Joanneum, Graz, Austria

Abstract

Statistical open data is usually provided only in the form of spreadsheets or CSV files. The developers of open data apps must either restrict themselves to managable bite-sized chunks of data, which can be consumed (read, parsed, and held in memory) in one go, or must install and maintain their own data server which the app can query on demand.
The Styrian Diversity Visualisation (in German "Steirische Vielfalt Visualisiert" or SVV) project demonstrates the use of a dedicated data server (triple store) to host large amounts of statistical open data. The SVV web app queries the data server dynamically using SPARQL queries to obtain exactly the data required at that particular time, greatly simplifying its internal logic. There is no need to parse and store entire data sets in memory.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—D.2.12 [Software Engineering]: Interoperability—Data Mapping

1. Introduction

Most statistical open datasets are made available as spreadsheet files (CSV, XLS) hosted on a data portal [Eur15; Lan15; OGD15; Sta15]. However, only relatively small datasets can be loaded directly from a data portal into an app, be parsed by the app, and then held in memory by the app, particularly on a mobile device with limited memory. When an app can fetch data on demand from a server over an API, the app can remain lean and slim, concentrating on its inherent purpose.

2. Linked Open Data and Data Servers

The emerging technologies of the semantic web [SHB06] provide a standardised infrastructure to host statistical open data on a data server. Data servers, or triple stores [Seq13], are databases which store data as RDF triples [W3C14], and are queried using SPARQL queries [W3C13]. Numerous implementations of triple stores are available, both open source and commercial. For smaller projects, Fuseki [Apa16] is often used. For larger projects, Virtuoso Open Source Edition [Ope16] is a good choice.

With a data server, an app can remain lean. There is no need for code to parse CSV files and no need to hold entire data sets in memory. Whenever data is needed, a corresponding SPARQL query is issued to the data server. An example is shown in Figure 1. The desired subset of data (and *only* that subset) is then returned to the app, typically in JSON or RDF format.

Of course, the extra functionality and flexibility of a data server does not come for free. The open dataset must be transformed into (RDF or TTL) triples and imported into the data server. Procedures must be in place to consistently name data fields and provide for versioning. Running a data server also requires regular backups, maintenance procedures, security updates, and monitoring, all of

```
prefix communes: <http://data.steiermark.at/communes#>
prefix countries: <http://data.steiermark.at/countries#>
prefix country_born: <http://data.steiermark.at/people/country_born#>
SELECT ?commune_name ?commune_code (sum(?commune_count) as ?count)
WHERE {
 GRAPH <http://data.steiermark.at/2014-01-01/people/country_born> {
  ?dataset country_born:communeCode ?commune_code .
  ?dataset country_born:sex 2 .
  ?dataset country_born:ageGroup 4 .
  ?dataset country_born:countryCode ?country_code .
  ?dataset country_born:count ?commune_count .
 }
 GRAPH <http://data.steiermark.at/2014-01-01/communes> {
  ?none2 communes:communeCode ?commune_code .
  ?none2 communes:communeName ?commune_name .
 }
}
ORDER BY asc(?commune_name)
```

Figure 1: *An example SPARQL query. Corresponding data records are returned to the app in JSON or RDF format.*

DOI: 10.2312/eurp.20161150

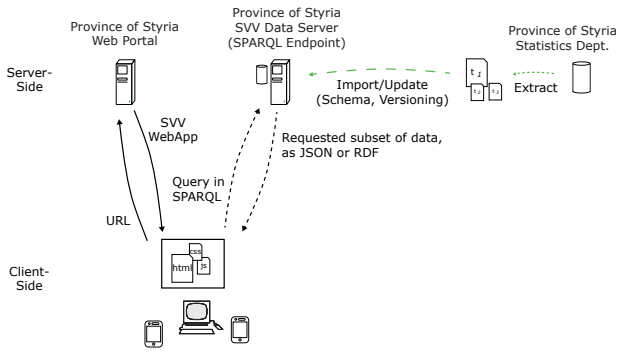

Figure 2: *The client-server architecture of the SVV web app. The user enters a URL into their web browser to request the web app. The web app is downloaded into the web browser and runs there. The app requests any data it needs on demand from a dedicated data server using SPARQL queries.*

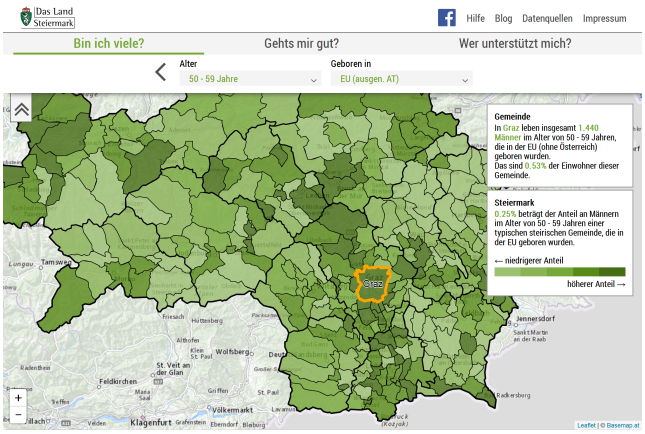

Figure 3: *The SVV web app is built around three scenarios and the principle of progressive disclosure. A choropleth map of corresponding statistical open data is successively refined and updated.*

which require significant commitment on the part of hosting organisations.

3. The Styrian Diversity Visualisation Project

The Styrian Diversity Visualisation (SVV) project comprises a web app and a data server, as shown in Figure 2. The user enters the URL of the web app into their web browser, and it is downloaded into the browser to run. As and when the webapp needs data, it requests it on demand by sending a SPARQL query to a dedicated data server. The data requested (and only that subset) is then returned to the web app, which displays it in an appropriate visualisation.

The user interface is built around three scenarios and the principle of progressive disclosure. The three scenarios are:

1. *"Bin ich viele?" ("How Many am I?")*: The user is prompted to enter their place of residence, gender, age range, and place of birth, in that order. In return, a choropleth map of Styria is successively updated to reflect the distribution of other people in the same demographic.
2. *"Geht's mir gut?" ("How am I Doing?")*: The user is prompted to enter their place of residence, employment status, and gender. In return, a choropleth map of corresponding income and employment statistics is successively refined and updated.
3. *"Wer unterstützt mich?" ("Who Can Support Me?")*: Markers are displayed showing the locations of individual integration partners.

As the user reveals more about themselves, the SVV app requests the refined data and regenerates the choropleth map accordingly.

The SVV web app is written in HTML5 with CSS3 and Javascript as a responsive web app [Mar14]. The web app adapts to the characteristics of the display device. On narrow screens, some blocks of content are relocated, menus are collapsed and placed behind a button, font sizes and spacing are reduced, and so forth. On devices which support touch interaction, appropriate code is loaded to support touch events.

On the data server side, an instance of Virtuoso Open Source Edition (VOS) is running on a Linux server with 16 gb of RAM and 4 cores. Individual data sets are prepared and then converted into LOD format (triples) using LODRefine [Zem13]. They are then uploaded onto the VOS data server. The VOS server is configured to use its standard port 8890 and http as the SPARQL endpoint to which client apps can send SPARQL queries. Scripts are used to semi-automate the process of uploading later versions and updates of the same data.

The choropleth map visualisation in the web app is realised using the Leaflet open source Javascript library [Aga16]. Leaflet, in turn, fetches map tiles from the basemap.at [bas16] map server. District outlines are overlaid as vector graphics.

4. Related Work

The "Qui Sommes Nous?" ("Who Are We?") project by the city of Rennes [Dat14; Ren15] uses similar data and a similar user interface. However, the Rennes visualisation is built on quite different technology. Data are loaded as multiple individual CSV files, which are then parsed and held in memory in the web browser. Graphical output is performed using the D3.js [Bos15] library, which essentially means dynamically injecting SVG nodes into the DOM.

5. Concluding Remarks

The Styrian Diversity Visualisation (SVV) project visualises the diversity of inhabitants of the Province of Styria (Land Steiermark) using open data served from a data server. The code developed during the project will be released as open source and the project documentation will be made publicly available. The project is intended to serve as a case study of how to develop lean visualisation web apps for large statistical open datasets. The project is due to launch to the public in May 2016. The current prototype is available at `http://projects.iicm.tugraz.at/opendata/svv/preview/`.

References

[Aga16] Vladimir Agafonkin. *Leaflet*. 8th Apr 2016. http://leafletjs.com/ (cited on page 2).

[Apa16] Apache. *Fuseki2*. Apache Jena Project. 8th Apr 2016. http://jena.apache.org/documentation/fuseki2/ (cited on page 1).

[bas16] basemap. *basemap.at*. City of Vienna and the Austrian Provinces. 8th Apr 2016. http://basemap.at/ (cited on page 2).

[Bos15] Mike Bostock. *D3: Data-Driven Documents*. May 2015. http://d3js.org/ (cited on page 2).

[Dat14] Dataveyes. *A Data-Driven Snapshot of the Population of Rennes*. blog post. Jun 2014. http://dataveyes.com/#!/en/case-studies/rennes-metropole (cited on page 2).

[Eur15] Eurostat. *Eurostat*. European Commission. 24th Jun 2015. http://ec.europa.eu/eurostat/ (cited on page 1).

[Lan15] Land Steiermark. *Open Government Data, Land Steiermark*. 24th Jun 2015. http://data.steiermark.at/ (cited on page 1).

[Mar14] Ethan Marcotte. *Responsive Web Design*. 2nd edition. A Book Apart, 2nd Dec 2014. 153 pages. ISBN 1937557189. http://abookapart.com/products/responsive-web-design (cited on page 2).

[OGD15] OGD. *Open Government Data Austria*. Bundeskanzleramt. 24th Jun 2015. http://data.gv.at/ (cited on page 1).

[Ope16] OpenLink. *Virtuoso Open-Source Edition*. OpenLink Software. 5th Apr 2016. https://github.com/openlink/virtuoso-opensource (cited on page 1).

[Ren15] Rennes. *Qui Sommes Nous?* Rennes Métropole. 23rd Jun 2015. http://dataviz.rennesmetropole.fr/quisommesnous/ (cited on page 2).

[Seq13] Juan Sequeda. *Introduction to: Triplestores*. dataversity. 31st Jan 2013. http://dataversity.net/introduction-to-triplestores/ (cited on page 1).

[SHB06] Nigel Shadbolt, Wendy Hall and Tim Berners-Lee. "The Semantic Web Revisited". In: *IEEE Intelligent Systems* 3.21 (Jan 2006), pages 96–101. ISSN 1541-1672. doi:10.1109/MIS.2006.62 (cited on page 1).

[Sta15] Stadt Graz. *Open Government Data Graz*. 24th Jun 2015. http://data.graz.gv.at/ (cited on page 1).

[W3C13] W3C. *SPARQL 1.1 Overview*. W3C Recommendation. 21st Mar 2013. http://w3.org/TR/sparql11-overview/ (cited on page 1).

[W3C14] W3C. *Resource Description Framework (RDF)*. W3C RDF Working Group. 25th Feb 2014. http://w3.org/RDF/ (cited on page 1).

[Zem13] Zemanta. *LODRefine: LOD-enabled Version of OpenRefine*. 5th Oct 2013. http://sourceforge.net/projects/lodrefine/ (cited on page 2).