Visualizing RDF Data Cubes using the Linked Data Visualization Model[⋆]

Jiří Helmich[1 2], Jakub Klímek[3 2], and Martin Nečaský[1]

[1] Charles University in Prague, Faculty of Mathematics and Physics
Malostranské nám. 25, 118 00 Praha 1, Czech Republic
{helmich, necasky}@ksi.mff.cuni.cz
[2] University of Economics, Prague
Nám. W. Churchilla 4, 130 67 Praha 3, Czech Republic
[3] Czech Technical University in Prague, Faculty of Information Technology
Thákurova 9, 160 00 Praha 6, Czech Republic
klimek@fit.cvut.cz

Abstract. Data Cube represents one of the basic means for storing, processing and analyzing statistical data. Recently, the RDF Data Cube Vocabulary became a W3C recommendation and at the same time interesting datasets using it started to appear. Along with them appeared the need for compatible visualization tools. The Linked Data Visualisation Model is a formalism focused on this area and is implemented by Payola, a framework for analysis and visualization of Linked Data. In this paper, we present capabilities of LDVM and Payola to visualize RDF Data Cubes as well as other statistical datasets not yet compatible with the Data Cube Vocabulary. We also compare our approach to CubeViz, which is a visualization tool specialized on RDF Data Cube visualizations.

Keywords: Linked Data, RDF, visualization, data cube

1 Introduction

Data analysts are accustomed to making projections from multi-dimensional datasets to low-dimensional ones using aggregations, slicing and dicing known from OLAP[3]. Those can be easily visualized by well-known and widely implemented techniques like charts, timelines, map visualizations, etc. More and more stakeholders including governments and scientific groups are publishing their datasets in a form of Linked Data[4]. Our goal is to apply the well-known visualization techniques which are understandable by non-expert users and use the Data Cube Vocabulary (DCV)[5] W3C Recommendation to achieve it. An expert user prepares a data cube and a non-expert one is provided with an easy way of exploring the cube with simple faceted visualization tools. In this paper, we demonstrate that our Linked Data Visualization Model enables us to create a flexible solution for RDF Data Cube visualizations that fit into a bigger, more general framework.

[⋆] The research is supported in part by the EU ICT FP7 under No.257943, LOD2 project and in part by project SVV-2014-260100.

[4] http://wiki.planet-data.eu/web/Datasets

[5] http://www.w3.org/TR/2014/REC-vocab-data-cube-20140116/

2 Linked Data Visualization Model

In our previous work we defined the Linked Data Visualization Model (LDVM) [1], an abstract visualization process customized for the specifics of Linked Data. LDVM allows users to create data visualization pipelines that consist of four stages: Source Data, Analytical Abstraction, Visualization Abstraction and View.

Source Data allows a user to define a custom transformation to prepare an arbitrary dataset for further stages, which require their input to be RDF. In this paper we only consider RDF data sources such as RDF files or SPARQL endpoints, e.g. DBPedia.

The *Analytical Abstraction* enables the user to specify analytical operators that extract data to be processed from a data source and then transform it to create the desired analysis. The transformation can also compute additional characteristics or even generate a new multi-dimensional dataset. For example, we can create a statistical dataset from DBPedia by querying for resources of type `dbpedia-owl:City` and using data from their properties such as `dbpedia-owl:populationAsOf` for a dimension and `dbpedia-owl:populationTotal` for a measure. Further analytical steps could be performed within this stage, e.g. filtering cities from a specific country.

In the *Visualization Abstraction* stage of LDVM we need to prepare the analytical data to be compatible with our Data Cube visualizer. In the case of the analytical data already being described by DCV, this stage can be skipped. Otherwise, we would have to use a LDVM transformer to convert non-DCV statistical data to DCV as it is the format required by our visualizer. This stage is what allows users to reuse statistical analyses with results in various formats without rewriting them simply by appending an appropriate transformer.

In *View Stage*, DCV-compliant data is passed to a visualizer which creates a user-friendly data cube visualization. Based on dimension links to SDMX and SKOS concepts, a visualizer can generate more sophisticated facets in order to let the user to slice and dice the data cube. A proper visualizer should contain the well-known data cube visualization techniques and in Payola, our LDVM implementation, we have such a visualizer.

3 Mapping non-Data Cube data to Data Cube

While experimenting with statistical data, we have encountered Linked Data datasets which contain statistical data, but do not use DCV. Since we have a visualizer using DCV, we implemented a tool, which is capable of mapping RDF non-cube data to a form compliant with DCV as a plugin usable in LDVM analyzers. While creating a new LDVM analyzer in Payola, a user is able to create a new instance of the DCV analytical plugin. On its input the plugin recieves arbitrary RDF data and based on a user-defined pattern, it maps the data to a specified DCV data structure definition. A user is asked to supply a URL containing at least one DCV data structure definition (DSD) in RDF. The user is presented with a list of available DSDs and after selecting one, a new analytical plugin is created for this DSD. This plugin can then be used by other Payola users without the need for specifying the URL with DSD and becomes a part of our extensible library of reusable DCV analyzers.

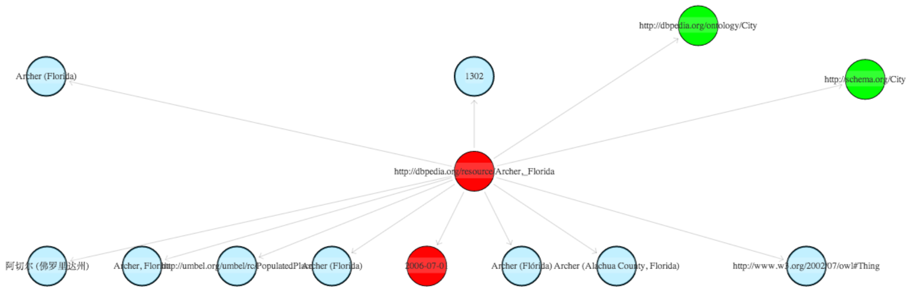

http://dbpedia.org/ontology/City
Archer (Florida)
1302
http://schema.org/City
http://dbpedia.org/resource/Archer,_Florida
阿切尔 (佛罗里达州)
Archer, Florida
http://umbel.org/umbel/rc/PopulatedPlace
Archer (Florida)
2006-07-01
Archer (Flórida)
Archer (Alachua County, Florida)
http://www.w3.org/2002/07/owl#Thing

Fig. 1. User inputs a mapping pattern

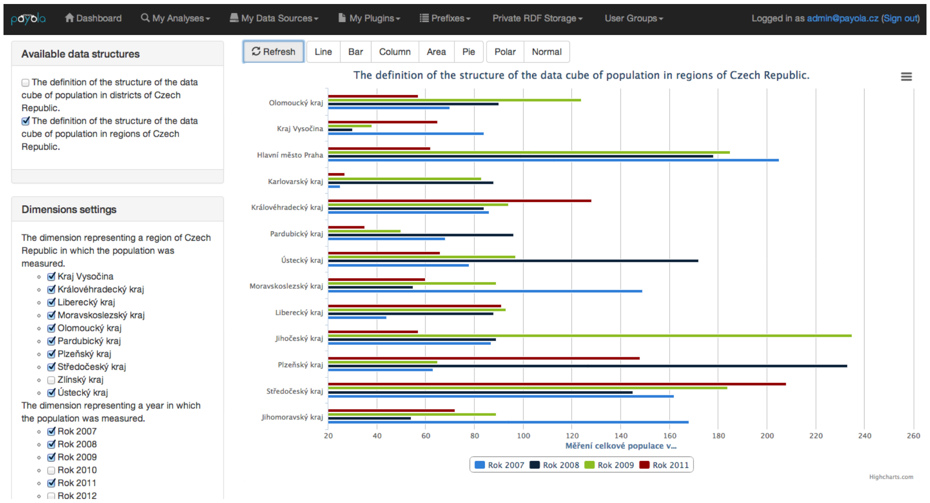
To be able to map an arbitrary dataset into a form compliant with DCV, the plugin needs the user to specify the data mapping. Based on DCV, this could be partially automated in the future. As can be seen in Figure 1 the process is based on the *query-by-example* principle. The plugin shows the user a generic graph visualization based on a preview of the input which will be processed by the DCV analytical plugin. It lets them to select a pattern: step by step, they are asked by the application to mark a vertex, which represents one of dimensions/measures/attributes of the chosen DSD (red vertices). To narrow down the volume of the results or to be able to specify more sophisticated patterns, the user is also able to mark vertices (green ones), which refine the pattern, but do not represent any DSD component. Based on the given example, the plugin produces a SPARQL query. When executed against a SPARQL endpoint, it creates new links between existing resources and components of the DSD.

The resulting plugin can be used in various ways in an LDVM analyzer. Connected directly to a data source it works as a filter and transformer which selects only data related to the specified DSD and maps it to DCV at the same time. It could also be beneficial for a user to use the plugin as an inner analytical operator to filter and map processed data since using DCV it becomes snowflake-shaped and can be easier to work with in further analytical steps. Or, as a final plugin of an analyzer, it can transform results of a non-DCV analysis into DCV in the same way a visualization transformation does.

4 Payola and CubeViz

Payola and *CubeViz* represent visualization tools that use DCV. Both of them use the Highcharts library to deliver user-friendly visualizations (line, bar, column, area and pie charts) (see Figure 2) and enable users to obtain a permanent link to a created visualization. When sent to a non-expert user, the link enables them to view a DCV-based visualization without any knowledge of Linked Data or DCV in an environment of a faceted browser. In addition, CubeViz provides a packing layout visualization of SKOS hierarchies using the d3js library. Such a visualizer is, however, also present in Payola but not as a part of the DCV visualizer as it can be also used in a more general way for non-DCV data.

Faceted capabilities of the two tools enable a user to slice a DCV cube, which means that they are enabled to select multiple values of two dimensions, one value from the rest of dimensions and choose a single measure. Configuring facets in such a way makes the tools load a 2-dimensional table, which is visualized by the aforementioned techniques. Both tools are technically capable of dicing (produces sub-cubes), but do not offer a way of visualizing more than 2 dimensions at a time.



Available data structures

☐ The definition of the structure of the data cube of population in districts of Czech Republic.
☑ The definition of the structure of the data cube of population in regions of Czech Republic.

Dimensions settings

The dimension representing a region of Czech Republic in which the population was measured.

- ☑ Kraj Vysočina
- ☑ Královéhradecký kraj
- ☑ Liberecký kraj
- ☑ Moravskoslezský kraj
- ☑ Olomoucký kraj
- ☑ Pardubický kraj
- ☑ Plzeňský kraj
- ☑ Středočeský kraj
- ☐ Zlínský kraj
- ☑ Ústecký kraj

The dimension representing a year in which the population was measured.

- ☑ Rok 2007
- ☑ Rok 2008
- ☑ Rok 2009
- ☐ Rok 2010
- ☑ Rok 2011
- ☐ Rok 2012

Fig. 2. An example of a visualization prepared in Payola. The four-dimensional cube is based on Czech Statistical Office data.

A DCV-based dataset could be visualized in both Payola and CubeViz with no additional transformations involved. The difference is that in Payola, any statistical RDF data can be transformed and visualized using the same data cube visualizer. In theory, CubeViz could even be used as an instance of a LDVM visualizer proving that LDVM is a more general and reusable framework. This could be achieved by supplying it with a DCV compatible LDVM visualization abstraction produced by a data cube LDVM pipeline. However, at the time of writing this paper, CubeViz was unstable and was crashing when loading data from our SPARQL endpoints so we could not finish evaluating this possibility.

5 Related Work

Tools like *OLAP2DataCube* [6] and *Tables* [6] enable users to convert non-RDF statistical datasets to DCV. Compared to Payola mapping process, they have a different input data type (relational data instead of RDF). In the phase of mapping data to DCV, they also rely on user input (selecting from a list or even using a custom DSL). From the group of more general visualization tools we name *VisualBox*[7] and *Exhibit* [4],

[6] http://idi.fundacionctic.org/tabels/
[7] https://github.com/alangrafu/visualbox

which are JavaScript based libraries that are not DCV capable and require the user to have scripting abilities. *GeoGlobe*[8] and map4rdf[9] visualize spatial statistical data from a fixed dataset. Also *Rhizomer* [2] offers multi-dimensional data visualizations (maps for spatial data, timeline, charts, etc.) without involving DCV. Payola and CubeViz rely on DCV as well as *Olap4ld* [5], which is an implementation of the Open Java API for OLAP and while converting OLAP operations to SPARQL, it introduces OLAP-to-SPARQL analytical approach. *Linked Statistical Data Analysis*[10] presents results of SDMX-ML transformations into DCV. It enables a user to visualize correlations over a fixed statistical datasets prepared by a set of custom analytical and transofmation scripts[11].

6 Conclusions

In this demo we present the Payola Data Cube Vocabulary mapping plugin that demonstrates how DCV can be utilized throughout the stages of LDVM. For the View Stage of LDVM we implemented a DCV visualizer in Payola that is capable of visualizing DCV datasets and provides a user with facets with slicing and dicing of data cubes. A sample DCV visualization is located at `http://vis.payola.cz/dcv_czso`. Compared to CubeViz, which is another tool for RDF Data Cube visualization, Payola, thanks to being a LDVM implementation, offers a wider range of usage scenarios. One of those scenarios is visualizing statistical data that is not described by DCV simply by mapping it to DCV as a part of a standard LDVM pipeline.

References

1. J. M. Brunetti, S. Auer, R. García, J. Klímek, and M. Nečaský. Formal Linked Data Visualization Model. In *Proceedings of the 15th International Conference on Information Integration and Web-based Applications & Services (IIWAS'13)*, pages 309–318, 2013.
2. J. M. Brunetti, R. García, and S. Auer. From overview to facets and pivoting for interactive exploration of semantic web data. *Int. J. Semantic Web Inf. Syst.*, 9(1):1–20, 2013.
3. S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. *SIGMOD Record*, 26(1):65–74, 1997.
4. D. F. Huynh, D. R. Karger, and R. C. Miller. Exhibit: lightweight structured data publishing. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 737–746, New York, NY, USA, 2007. ACM.
5. B. Kämpgen and A. Harth. No size fits all ? running the star schema benchmark with sparql and rdf aggregate views. In *ESWC 2013, LNCS 7882*, pages 290–304, Heidelberg, Mai 2013. Springer.
6. P. E. Salas, M. Martin, F. M. D. Mota, K. Breitman, S. Auer, and M. A. Casanova. Olap2datacube: An ontowiki plugin for statistical data publishing. In *Proceedings of the 2nd Workshop on Developing Tools as Plug-ins*, TOPI 2012, New York, NY, USA, 2012. ACM.

[8] `http://data.i2g.pl/insigos/hz-geo/globe/`

[9] `http://oegdev.dia.fi.upm.es/map4rdf/`

[10] `http://stats.270a.info/`

[11] `https://github.com/csarven/publishing-statistical-linked-data/blob/master/csarven.publishing-statistical-linked-data.pdf?raw=true`