Engineering Open Data Visualizations over the Web

Rober Morales-Chaparro, Fernando Sánchez-Figueroa, and Juan Carlos Preciado

Quercus Software Engineering Group,
University of Extremadura, Spain
{robermorales,fernando,jcpreciado}@unex.es

Abstract. Data Visualization has been traditionally one of the main pillars for understanding the information coming from Business Intelligence/Analytics based systems. While building visualizations has never been an easy task, moving data visualization for open data over the Web adds extra complexity. The variety of data sources, devices and users, together with the multidimensional nature of data and the continuous evolution of user requirements makes Data Visualization on the Web for open data more complicated as well as challenging. This paper briefly introduces a DSL for engineering open data visualizations over the Web whose aim is twofold. On the one hand, overcoming technological dependencies such as data source, data set, rendering technologies and so on. On the other hand, bringing the possibility of building useful graphics to users with domain expertise but non-technical skills.

Keywords: Open Data, Software Engineering, Data Visualization, Domain Specific Language.

1 Introduction

Available data on the Internet is growing up more and more every day, with heterogeneous formats and sources. The production of data is being accelerated with the widely use of approaches close related to open data such as linked data, big data, social data and so on. The emergence of new related fields like the Internet of Things and Smart Cities are adding fuel to the fire by providing an immense amount of data that must be processed to be transformed onto useful information and eventually knowledge.

Although the term open data is not new, it is gaining popularity with the launch of open data government initiatives such as data.gov. Currently, many public administrations and enterprises are following open data best practices and publishing their data in RDF format. This format allows users to query the data using a SPARQL endpoint.

The data resulting from these queries need to be explored, analyzed, interpreted or even communicated to/with other people with different profiles. For these actions data visualization is a key concept [1].

However, visualizing open data is not an easy task. It mainly involves issues at two different levels: technical and social. From the technical point of view, the visualization must be rendered in different platforms and devices, connected to different and heterogeneous data sources and it also should make use of the best rendering
technologies available at a given moment. Moreover, the visualization must change accordingly to the real-time changes performed on data. From an engineering point of view, it should be desirable for the visualizations to be reusable and defined as much independent as possible from all these technology issues, leading to a self-service data visualization [2]. From the social point of view, the information should be shown in different ways attending to the user changing context (user profile, company interests, colleague interests, market tendencies, social tendencies, etc.). Moreover, some kind of interactions for the user should be desirable in order to change the dimension of the data shown (multidimensional data visualization) or create new graphics in a non-technical way.

Although there is a significant number of methodologies and tools for visualizing data, the real fact is that, for different motives, not all of them are suitable for dealing with open data. This is the reason why we developed Visualligence, a model-driven, data-driven and user-driven approach for visualizing open data [3]. One of the main pillars of Visualligence is a Domain Specific Language (DSL) [4] for defining visual patterns. While an overview of the proposal is provided in [3], the presentation of the main features of the DSL is the main contribution of this paper. This DSL allows both, to capture formally the visualization process and to be used by non-experts users.

The rest of the paper is as follows. Section 2 deals with the limits of existing visualization technologies for coping with open data. Section 3 briefly introduces the main features of the mentioned DSL. Conclusions are omitted for space reasons.

2 Limitations of Existing Approaches

From the point of view of open data, this section outlines some of the problems arising in visualization solutions, which have been split in several categories:

- **Spreadsheets.** Microsoft Excel is used to store and visualize data by near to 95% of business around the World. Both Excel and other spreadsheets systems (LibreOffice, Google Drive) lack of full support of data visualization, more if we are referring to open data. They are not extensible, so we cannot make new graphics in an easy way. We are always limited to the initial graphic set. We cannot extract a configured graphic from one document to another, keeping the visualization slave of a dataset. They can connect with external sources, but only for a few technologies, none of them related to Open Data, and with a limited size of the dataset.

- **Statistical tools** such as SPSS and R have been used for years for visualizing scientific results and they can manage very large datasets. However, they usually require importing the dataset to their own format, constraining possibilities of external sources. This is not the ideal solution for managing real time updates. Although they are oriented to general-public, they do not allow for the design of new graphics by non-technical users.

- **Business Intelligence** systems are statistical packages with improved visualization. They have remote data connections but lack of dataset independence of dashboards configured: they lock all the work under the vendor technology. Extensibility options that allow the median user create or share new visualizations are missing.