LinDA - Visualising and Exploring Linked Data

Klaudia Thellmann, Fabrizio Orlandi, Sören Auer

University of Bonn & Fraunhofer IAIS, Bonn, Germany
klaudia.thellmann@iais.fraunhofer.de, orlandi@iai.uni-bonn.de,
auer@cs.uni-bonn.de

Abstract. The main goal of our work in the context of the LinDA (Linked Data Analytics) project is to offer small and medium sized enterprises (SMEs) possibilities for integrating and consuming data by using Linked Data technologies. One of the major challenges of this project consists in providing user-friendly means of exploring and visualising Linked Data. To achieve this, a Semantic Web application has been created, based on state-of-the-art linked data visualisation approaches, which allows a largely automatic matching and binding of data to visualisations. Hence, in this demo paper we demonstrate the potential of a visualisation framework which is capable of dealing with different data formats, serialisations and Semantic Web ontologies.

Keywords: linked open data, open data consumption, visualisation

1 Introduction

The increasing number of publicly available datasets poses a challenge regarding the integration and consumption of information. The aim of the LinDA project[1] is to make the benefits of Linked Open Data accessible to SMEs and data providers by providing libraries for Open Data consumption. One of the main tasks in this context is to build an ecosystem of tools for visualising Linked Data to assist SMEs in their daily tasks by hiding complexity through automation and an intuitive user interface. To complete this task, a generic visualisation workflow[2] is being implemented based on state-of-the-art Linked Data visualisation approaches [1][3]. Most existing approaches are only usable by a technical audience or limited to certain domains or data representations [2]. Voigt et al. propose a generic approach for visualisation selection in form of a faceted browser that imposes on the user the task of describing the visualisation at an unfamiliar level of abstraction.

By taking the well-established visualisation tool Tableau Public[3] as an upper boundary regarding complexity we aim to find a balance between generality and ease of use. Hence, we aim at improving on existing approaches and supporting the user in visualising arbitrary Linked Data through automatic visualisation

[1] http://linda-project.eu/
[2] A demonstration of the prototype is available at http://goo.gl/bSgvjn
[3] http://www.tableausoftware.com/public/

recommendation and pre-configuration and intuitive customization. For this we consider using the visualisation ontology proposed by Voigt et al. besides the other components for conversion, data analysis and (pre-)selection developed within the LinDA project in order to realize and evaluate a generic and user-centered visualisation workflow.

2 LinDA Visualisation Workflow

The visualisation workflow, as depicted in Figure 1, is used for supporting the user in selecting and configuring visualisations and consists of the following steps:

1. **Select data:** The user starts with the selection of the dataset she intends to visualise (Fig. 2, left). The input data needs to be either in RDF or tabular format in order to proceed to the next step.
2. **Select visualisation:** Based on the content and format of the data and the semantic descriptions of the available visualisation widgets, a ranking of possible visualisations is computed and presented to the user (Fig. 2, right).
3. **Configure visualisation:** After choosing a visualisation from the list of recommended visualisations, the user proceeds to the configuration step. Here, she needs to provide the input necessary for the application in order to map the data to the chosen visualisation (Fig. 3, top). Data conversion may be performed automatically at this stage in case the selected visualisation widget requires a different format.
4. **Visualise:** Finally, following the configuration step, the visualisation and exploration phase of the input data can be performed (Fig. 3, bottom). At this stage, the user can further customise the visualisation or export it in different formats, share or publish it or save it for later reuse.

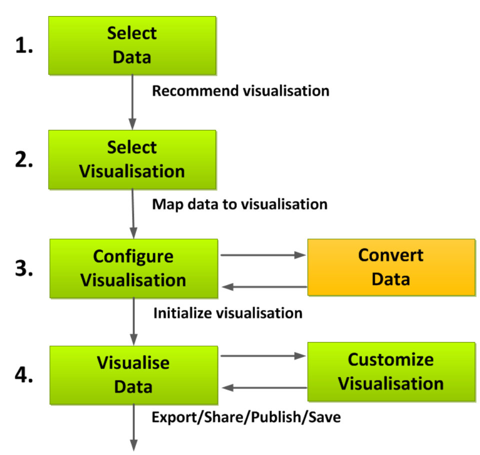

Fig. 1. The Linked Data Visualisation Workflow.

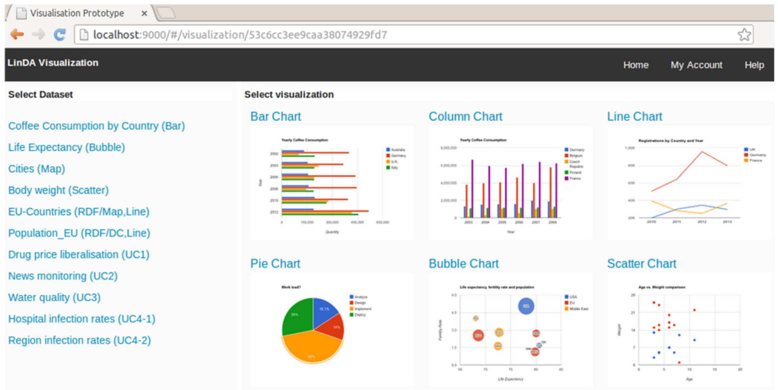

Fig. 2. Datasource and visualisation widget selection

In the following sections, we briefly introduce the approaches for the automatic suggestion of suitable visualisations and the automatic mapping of input data to a visualisation being developed.

Recommending Visualisations In order to rank suitable visualisations for RDF input data, a similarity measure between the vocabularies used in the input data and the vocabularies supported by the visualisation widgets is calculated. For RDF input data the list of visualisations can also include visualisations with different input formats if corresponding converters are present. If the data is in a non-RDF format, the recommendation is based solely on the format, without similarity calculation or other recommendation strategies.

Mapping Data to Visualisations The approach used for automatically mapping data to visualisations is an extension of the state-of-the-art LDVM [1]. Each visualisation is described by a visualisation model which consists of (1) an input data type, e.g. cube (table), network or tree, and (2) configuration options defining structure and layout of the visualisation. For instance, a bar chart has the cube data model and has the mapping of CSV columns or RDF properties to vertical and horizontal axes as structure options and the axis labels as layout options. The approach for mapping data to visualisations consists of the following steps: First, the appropriate input data type is determined based on the content and format of the data. Then, a configuration form is composed based on the configuration options of the visualisation, which can be pre-configured automatically if the data is in RDF format (Fig. 3). The content of the configuration form can vary depending on the input data type (e.g. for CSV data, the axes of a bar chart can be mapped to the columns of the input file, while for RDF data, the mapping is realised according to the properties of the instances of a class selected by the user). A pre-configuration is created in both cases, but as in CSV little semantic information is present, it can only be created on a low level. In case of RDF, however, it can be automatically inferred from the RDF

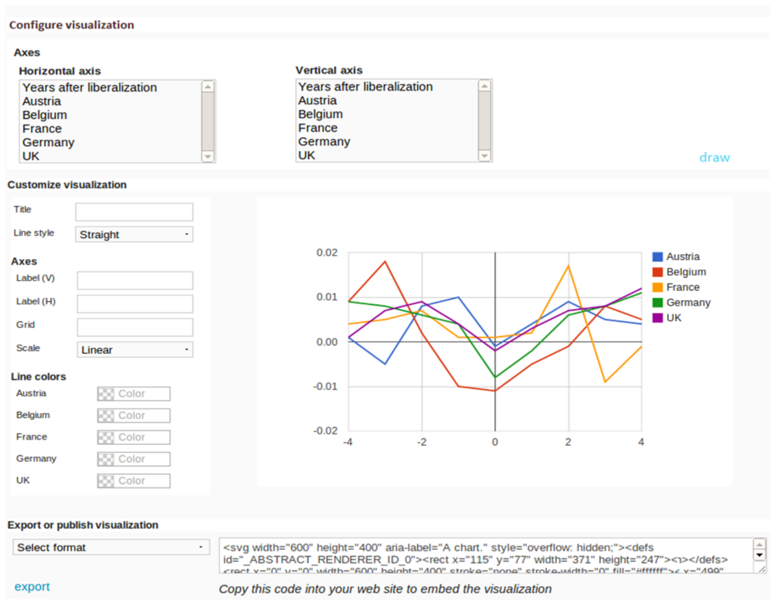

Fig. 3. Configuration and visualisation

vocabulary of the input data by linking properties of a visualisation to properties of a vocabulary.

3 Conclusion and Future Work

In this demo paper, we have introduced a generic approach for automatically suggesting visualisations and binding data of different formats and vocabularies, with the goal of providing SMEs with an intuitive way of exploring and visualising data, especially Linked Data. In the future we plan to: (1) expanding the range of input datatypes, e.g. networks, trees (2) introduce a generic ontology for describing data and visualisation widgets (3) improve the user experience by introducing intuitive configuration form templates for selecting and exploring RDF data, and (4) conduct an extensive evaluation through a user study.

References

1. Josep Maria Brunetti, Sören Auer, Roberto García, Jakub Klímek, and Martin Nečaský. Formal linked data visualization model. In *Proc. IIWAS*, IIWAS '13, pages 309–318, NY, 2013. ACM.
2. Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2(2):89–124, 2011.
3. Martin Voigt, Stefan Pietschmann, and Klaus Meißner. A semantics-based, end-user-centered information visualization process for semantic web data. In *Semantic Models for Adaptive Interactive Systems*, pages 83–107. Springer, 2013.