An OLAP Endpoint for RDF Data Analysis Using Analysis Graphs

Median Hilal*, Christoph G. Schuetz, and Michael Schrefl

Department of Business Informatics – Data & Knowledge Engineering,
Johannes Kepler University Linz, Austria
http://www.dke.uni-linz.ac.at/

Abstract. Exploiting Resource Description Framework (RDF) data for Online Analytical Processing (OLAP), especially Linked Open Data (LOD), could allow analysts to obtain interesting insights. To conduct OLAP analysis over RDF data, analysts should know the specific semantics, structure, and querying mechanisms of such data. Furthermore, these data should ideally adhere to a multidimensional structure to be accessible to OLAP. In this demo paper, we present an OLAP endpoint that allows casual analysts to perform self-service OLAP analysis over RDF datasets. Specifically, analysts can instantiate semantic web analysis graphs, which are predefined models of the analysis processes. Semantic web analysis graphs are built on top of multidimensional structures that can be superimposed over arbitrary RDF datasets.

Keywords: Linked Open Data, Multidimensional Model, Self-Service Business Intelligence

1 Motivation

Online Analytical Processing (OLAP) systems support business analysts in their decision-making processes by allowing them to view the data at different granularities. In order to be accessible to OLAP, these data are organized in a multidimensional (MD) structure. An MD model consists of facts which are the subjects of the analysis and quantified by measures, and hierarchically-organized dimensions allowing for measure aggregation. Traditionally, OLAP systems have targeted enterprise-internal data. Yet, external RDF data, such as DBpedia and Wikidata, are an important source of knowledge that is still largely unexploited for OLAP analysis. External RDF data, however, do not correspond to a structure easily accessible to OLAP systems. The superimposition of analytical schemas is, therefore, a common strategy to render these data accessible to OLAP [2]. Ideally, these superimposed analytical schemas should adhere to an MD structure. Furthermore, the analyst should be familiar with the data structure, semantics, and query language in order to perform analysis over external

* Median Hilal is funded by Erasmus Mundus - ASSUR Program.

RDF data. RDF data sources typically feature complex and heterogeneous data models and SPARQL serves as the query language, which is not familiar to most business analysts. Consequently, mechanisms of self-service business intelligence (SSBI) are required as they facilitate the tasks of casual analysts [1].

In this demo paper[1], we present a self-service OLAP endpoint [3] using Semantic Web Analysis Graphs, which allow for expressing and exploiting interesting analysis patterns that are built on top of MD schemas which are superimposed over RDF data sources. To motivate our approach, we consider the RDF data of annotated news[2], which are available at factforge endpoint[3]. The dataset contains mentions of entities in the news; a *Mention* has a *source*, *creationDate*, *category*, *mentionsEntity*, and *confidenceScore*. Consider *Julia*, a data journalist preparing a report about trending industries in the news in a specific time period. Julia needs to identify the top mentioned industries, and for each industry, compare the number of mentions for biggest companies in terms of number of employees. Afterwards, she wants to manipulate the results by news categories. Performing that manually is misleading, cumbersome, and needs a lot of skills. Nevertheless, with our approach, the knowledge about this analysis process can be modelled as a dynamic analysis graph which refers to a superimposed MD schema. This analysis graph can be easily reused in multiple cases as it contains variables that can be bound by the analyst to concrete values without a need of technical knowledge. Consequently, *Julia* can access an OLAP endpoint, instantiate the analysis graph to suit her needs using a simple interface, and view the results. The system automatically generates the corresponding SPARQL queries to retrieve the data from sources and delivers the results [3].

2 Approach

Multidimensional schemas (MDS) can be superimposed over arbitrary RDF datasets to express MD analytical possibilities. Superimposition means that the data do not originally follow an MD structure, but this structure is rather imposed on top of data that are left unchanged. A mapping, however, relates MDS to the data in a GLAV (global-as-view) fashion using SPARQL queries. An OWL ontology is established to express the MDS metamodel. This metamodel can be instantiated to express MD schemas for particular RDF sources. The upper part of Fig. 1 shows an example instantiation of the MDS metamodel for the news example. *Mention* is the fact. *Source, Category, Entity,* and *Time* are dimensions with some having hierarchies. *ConfScore* and *NumOfMentions* are measures. *Entity* dimension illustrates specialization as *MentionedEntity* is specialized to *Person* and *Organizationn*, which is further specialized to *Company*. *Company* rolls up to *Industry* and has number of employees *NumOfEmps* as an attribute.

[1] A video demonstrating the developed prototype can be found at https://youtu.be/ymhkqla8J1I

[2] http://ontotext.com/semantic-solutions/dynamic-semantic-publishing-platform/linked-data-integration-for-global-publishers/

[3] http://factforge.net/sparql

We extend Analysis Graphs, proposed for relational data warehousing [4], to become Semantic Web Analysis Graphs (SWAG) that model the analytical process on the schema level, and are a facilitator for SSBI over MDS as they provide accesses to prepared reports [1]. The nodes of an analysis graph are analysis situations, each representing a multidimensional query. The edges of an analysis graph are navigation steps, each representing one or more OLAP operations that transform the source analysis situation into the target analysis situation. Analysis graph schemas may contain variables to be bound during instantiation. The lower part of Fig. 1 shows a simplified example of an analysis graph from the motivating scenario.

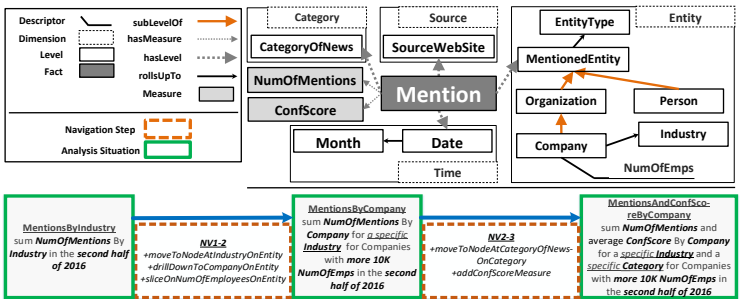

Fig. 1. Examples of MDS (up) and SWAG (down) from the motivating scenario

3 Implementation

Figure 2 illustrates the analysis process using analysis graphs. The analyst first selects an analysis situation to start from in the analysis graph. Then, the analyst can view the analysis situation specification and unbound parameters and bind them supported by auto-completion. Once the form is submitted, a SPARQL query which corresponds to the analysis situation is generated and sent to the data source's endpoint. The results are then retrieved and can be viewed in tabular form or as charts. Afterwards, the analyst can select a navigation to one of the subsequent analysis situations. Once a navigation is performed, the target analysis situation is activated with the operations corresponding to the performed navigation. The graphical user interface facilitates the process and guides the user. Figure 3 sketches the system architecture. The OLAP Endpoint is implemented as Java Server Pages (JSP). D3 force layout is used to visualize the analysis graph, while Google charts are used to visualize the results. The endpoint is controlled via the *Controller*. The overall analysis process is coordinated and performed by the *Analysis Graphs Execution Engine*. Parsing and reasoning over MDS and SWAG ontologies and their instances is performed by *OWL Handler*, which is implemented using Jena libraries. *SPARQL Query Creator* is implemented using Jena ARQ library, and is responsible for generating

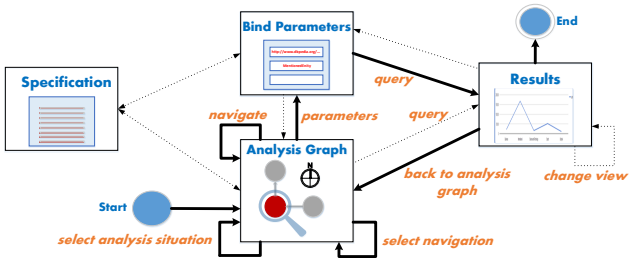

Fig. 2. User interaction process (bold arrows express the main analysis process flow)

SPARQL queries that correspond to the MD queries specified by bound analysis situations. *Connection Manager* is responsible for accessing external required files and interacting with the dataset *Endpoint*.

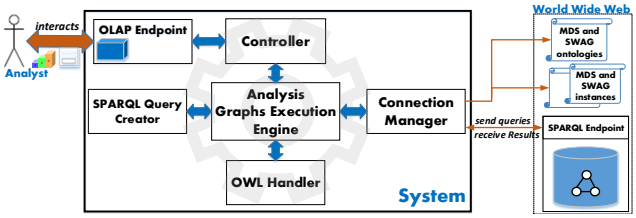

Fig. 3. System architecture

4 Future Work

We plan to enable analysts to manipulate analysis graphs on the fly, thus extending or editing the analysis situations and navigations at runtime or navigating without a predefined schema. Furthermore, we plan to conduct a user study to evaluate the usability of the approach and improve it correspondingly.

References

1. P. Alpar and M. Schulz. Self-service business intelligence. *Business & Information Systems Engineering*, 58(2):151–155, 2016.
2. D. Colazzo, F. Goasdoué, I. Manolescu, and A. Roatiş. RDF analytics: lenses over semantic graphs. In *Proceedings of the 23rd international conference on World wide web*, pages 467–478. ACM, 2014.
3. M. Hilal. A proposal for self-service OLAP endpoints for linked RDF datasets. In *European Knowledge Acquisition Workshop*, pages 245–250. Springer, 2016.
4. T. Neuböck and M. Schrefl. Modelling knowledge about data analysis processes in manufacturing. *IFAC-PapersOnLine*, 48(3):277–282, 2015.