Demonstration of Using a Domain-Specific Visual Modeler for Building Semantic Queries

Gábor Simon, Dániel Palatinszky and Gergely Mezei

Department of Automation and Applied Informatics, Budapest University of Technology and Economics
simon.gabor@aut.bme.hu, pd1326@hszk.bme.hu, gergely.mezei@aut.bme.hu

Abstract. In the age of Big Data, data mining, exploring and extracting meaningful information from massive datasets became a natural need. Currently this is a natural habitat for data scientists who have the knowledge about standards, tools and query languages. While more and more datasets were made available to the public, the exploration and extraction tools for non-technical users remained limited. Usually only simple keyword search or browsing through pre-determined navigation paths are offered to this audience. The aim of our research is to create an intuitive, user friendly visual query editing solution that is usable to create queries for semantic web databases. We designed a domain-specific language and built a fully customized, web-based editing workbench upon it. The workbench achieves high usability and comprehensibility.

Keywords: semantic dataset, visual query, Wikidata

1 Introduction

It can be a challenging task to get the simplest piece of information from a massive dataset without the support of a visual interface even for a user with far-reaching domain knowledge. Data stores can be addressed in some kind of specialized query language, but these languages require a not necessarily technical user to master a highly technical language and translate his natural language query to something that seems to be a cryptic and unnatural script for him. To solve this issue, various kinds of higher level interfaces were developed as surveyed recently in [7] in order to close the gap between the mindset of the user and the comprehension of the data store [4]. All these interfaces are balanced between expressiveness, flexibility and usability. Most of the highly usable tools restrict user queries to match some of the predetermined query patterns [6], while the other side of the spectrum is represented by highly technical editors visualizing concepts and verbs directly from the underlying textual query language [1]. Our research aims to raise the level of customization by offering greater flexibility in expressing the queries, but at the same time keeping the visual language simple enough to be understood by average, non-technical people.

Visual domain-specific modeling environments provide solution in a similar, but more generic problem space. These environments offer a highly customizable and flexible interface for arbitrary textual or visual domain-specific languages (DSLs). The core of our solution is a metamodel-based, general purpose visual domain-specific modeling system, the Visual Modeling and Transformation System (VMTS [2]). We have created a custom domain-specific language capable of describing queries for the semantic data and customized the web-based editing interface to meet the challenges in this domain. In this paper, we introduce our editing environment referred to as SemEx (Semantic Extensions for VMTS) and show its mechanisms by using an example.

2 The SemEx Visual Query Environment

The first step of our work was to create a language describing the queries. Since our solution is based on a metamodeling system, the language is defined by a metamodel. Our metamodel was created on the SPARQL query language [5] in mind, however it does not capture the full expressive power of the SPARQL 1.1 grammar, as we focused our efforts to support the queries that can be expressed using basic graph patterns (BGP).

In the SemEx metamodel the outermost concept is *Query*, which represents a semantic query. A *Query* consists of *Query Elements*. There are several types of *Query Element*s: *Subject*, *Statement*, *Property* and *Object*. *Subject* is a known or unknown entity the users have and/or need information about. *Statement*s can be added to a *Subject* to encode the known or missing pieces of information. A single *Statement* represents an atomic piece of information about a *Subject*, and it contains a single *Property* and a single *Object*. The *Statement* **st** added to *Subject* **s** with *Property* **p** and *Object* **o** encodes the knowledge that **s** has a property **p** with the value of **o**. Moreover, we also distinguish *Primitive*s that are specialized *Object*s holding a literal value.

Query Elements can be either defined or undefined. Defined elements hold a reference to a particular entity in the dataset or in case of *Primitives*, they have a literal value. In contrast, undefined elements specify an alias that can be referenced in the query results.

SemEx provides a web based interface to build a query in the form of a graph. Nodes have customized visualization encapsulating a *Subject* and its *Statements*, while an edge between two nodes encode the knowledge that both the nodes denote the same entity or value.

To define a *Query Element* (e.g. *Subject* or *Property*), users are aided with a simple filter interface that helps them to choose a concrete entity. The filter is implemented as an auto-complete search box. In case of *Primitives* (literal values) input fields are used. For undefined query elements, users can type in an alias. It is possible to express equality between two elements by either using the same alias, or by connecting the elements graphically. An edge between two subnodes encode the knowledge that both the subnodes denote the same entity or value.

Once the visual query is built by the user, SemEx can generate a SPARQL query from the model graph. The generated query can be inspected in a syntax highlighted panel that also supports manual editing of the final query. Finally, the query can be executed from the environments on the specified SPARQL endpoint and the results are visualized in a tabular form.

We tested our tool with one of the major publicly accessible semantic dataset Wikidata [8] through utilizing its public SPARQL query service [3]. To test our approach, we have built queries with complexity equivalent to the ones presented in the user study of [6].

As a case study we present the visual model constructed to answer a question with maximum complexity that users encountered through the user study. From this type of question results a query with long (at least 3) linear chain of connected elements and contains at least two undefined elements of the same type. Paper [6] refers to this expressiveness category as "*Long with branching and type III cycle (T6)*". We constructed the following category T6 task: "Find the names of all the people that directed a science fiction movie and had a role in a film that has an actor with a given name "George" who was born in the European continent and plays in a rock band." The resulting visual model is depicted in Figure 1. For more models of the same case study, please refer to the SemEx page on [2].

Through the demo session we are going to build similar queries interactively, driven by the input from the audience.

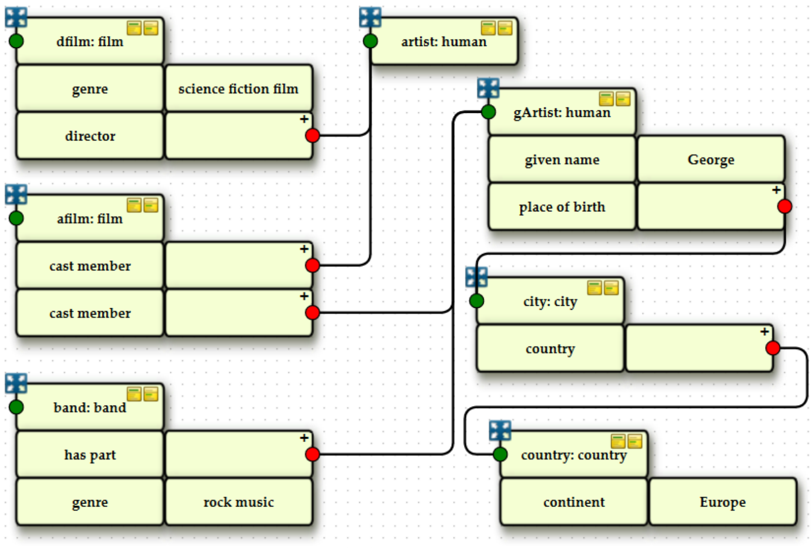

Fig. 1. Visual model of a T6-type sample query

3 Conclusions

Massive semantic datasets like Wikidata provide simple yet efficient tools for everyone to browse and edit semantic data. However,even for the simplest queries, users have to learn a textual query language. In this paper, we introduced a new approach to create and execute visual queries on semantic datasets. We recognized that crafting a query visually is essentially equivalent to the process of editing a special visual model. As a result we were able to build our tool on top of an existing visual modeling framework, VMTS. We mapped the structure of SPARQL queries to a metamodel and thus created a domain-specific language for this domain. Using the metamodel, VMTS provided an initial editing environment, which we customized in order to improve usability. The result, SemEx, offers a clean, accessible and uncluttered web interface to formulate information retrieval tasks with ease. Moreover, harnessing the capabilities of the underlying VMTS infrastructure, users can collaborate real-time through the query formulation process, making SemEx a social data mining solution for people without programming skills. Currently, the metamodel supports only queries that can be translated to basic RDF graph patterns. We are working on adding support for more complex query concepts, like group graph patterns or refined relations (e.g. negation) between entities and values. As we built our approach upon a visual modeling system, increasing the expressiveness is a natural process: we enrich the metamodel and add more customization to the user interface. As a result, we can offer a more capable tool with the continuous support of the base services provided by the underlying modeling infrastructure.

Acknowledgments This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Széchényi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

References

1. OpenLink iSPARQL, https://www.openlinksw.com/isparql
2. VMTS website, http://vmts.aut.bme.hu
3. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing Wikidata to the linked data web. In: Proceedings of the 13th International Semantic Web Conference (ISWC'14). LNCS, vol. 8796, pp. 50–65. Springer (2014)
4. Freitas, A., Curry, E., Oliveira, J.G., O'Riain, S.: Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends. IEEE Internet Computing 16(1), 24–33 (2012)
5. Harris, S., Seaborne, A., Prud'hommeaux, E.: Sparql 1.1 query language. W3C recommendation 21(10) (2013)
6. Soylu, A., Giese, M., Jimenez-Ruiz, E., Vega-Gorgojo, G., Horrocks, I.: Experiencing OptiqueVQS: A multi-paradigm and ontology-based visual query system for end users. Univers. Access Inf. Soc. 15(1), 129–152 (2016)
7. Vega-Gorgojo, G., Slaughter, L., Giese, M., Heggestøyl, S., Soylu, A., Waaler, A.: Visual query interfaces for semantic datasets: An evaluation study. Web Semantics: Science, Services and Agents on the World Wide Web 39, 81–96 (2016)
8. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM 57, 78–85 (2014)