Tripcel: Exploring RDF Graphs using the Spreadsheet Metaphor

Bernhard Schandl
University of Vienna, Dept. of Distributed and Multimedia Systems
Liebiggasse 4/3-4, A-1010 Wien, Austria
bernhard.schandl@univie.ac.at

ABSTRACT

Spreadsheet tools are often used in business and private scenarios in order to collect and store data, and to explore and analyze these data by executing functions and aggregations. They allow users to incrementally compose calculations by filling cells with formulas that are evaluated against data in the sheet, whereas expressions can be nested via cell references. In this paper we present Tripcel, a tool that applies the spreadsheet concept to RDF. It allows users to formulate expressions over the contents of an RDF graph, to arrange these expressions in a grid, and to interactively inspect their evaluation results. Thus it can be used to perform analysis tasks over large data sets within an understandable and familiar interface.

Categories and Subject Descriptors

H.4.1 [**Information Systems**]: Office Automation—*spreadsheets*; H.5.2 [**Information Systems**]: User Interfaces—*interaction styles, theory and methods*

General Terms

Algorithms, Design, Human Factors

Keywords

Spreadsheets, Semantic Web, query language, data analysis

1. INTRODUCTION

The basic information unit of RDF is the triple, which denotes either a specific type of relationship between two entities, or a specific attribute of an entity. Consequently, tools to explore, navigate, and analyze RDF data have been designed based on the triple concept: they represent triples either in graphical form (e.g., by rendering an edge between two nodes) or in textual and tabular form (e.g., by representing triples in a subject/predicate/object table).

Such interfaces are only partially useful when one wants to analyse existing data in a more structured way. For such tasks, query languages like SPARQL can be used: these allow users to precisely specify their information needs, can be evaluated and optimized by query execution environments, and return their results in structured form. However, complex queries are difficult to author and to test, and tool support for query authoring is relatively weak.

In this respect, we can observe a parallelism to mathematics and business applications, where spreadsheet tools are used to break down complex calculations into smaller formulas. These can be distributed in a two-dimensional grid of cells, and cells can be referenced within formulas so that the results of other expressions can be re-used. This paradigm allows users to incrementally compose their analysis framework, and to immediately observe the effects of modifications to formulas or data.

We have applied the spreadsheet concept to RDF and present *Tripcel*, a tool that allows users to edit expressions over RDF language elements (i.e., resources and literals), to arrange these expressions in a spreadsheet, and to evaluate them against an RDF graph (the *background graph*). This tool allows users to quickly perform simple queries over RDF data, or to compose complex calculations in an iterative and intuitive manner.

2. FUNCTIONS AND EXPRESSIONS

Tripcel[1] is based on RDFunctions [1], a formal framework for expressing functions over RDF language elements (URIs, blank nodes, and literals). RDFunctions essentially transform a set of input elements to a set of output elements under consideration of the contents of an RDF graph that influences the way the transformation is performed. The most important difference to RDF query languages is that RDFunctions can be arbitrarily nested: the input data of an RDFunctions is not an RDF graph, but an RDF language element, as is its output.

For a detailed specification of RDFunctions we refer to [1]; here we give one example of an RDFunction expression that returns all resources in the background graph G whose `rdf:type` is `swrc:Publication`:

$$invproperty_G^{\texttt{rdf:type}}(resource^{\texttt{swrc:Publication}}(\cdot)) \qquad (1)$$

A number of RDFunctions have already been defined, including functions that return elements (resources, properties, literals) from the background graph, functions that return property values for resources, and aggregate functions.

[1]The Tripcel prototype as well as a tutorial can be downloaded from `http://www.ifs.univie.ac.at/schandl/2009/06/tripcel`.

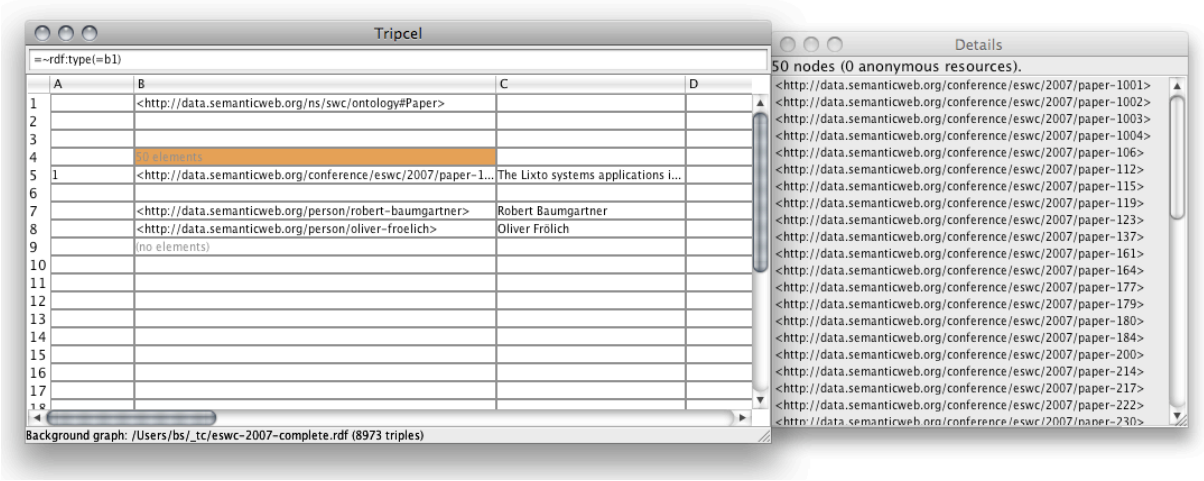

Figure 1: Tripcel screenshot: spreadsheet window (left) and detail window (right)

Additionally, a subset of SPARQL `FILTER` operators is supported.

RDFunctions are similar to RDF query languages in that they return sets of RDF language elements (like, e.g., a SPARQL `SELECT` query). They are also similar to spreadsheet expressions with the major difference that while spreadsheet formulas always return zero or one result, RDFunction expressions may return an arbitrary number of results, as shown in the previous example.

To allow users to edit RDFunction expressions an intuitive syntax has been defined [1]. Its design is oriented towards the syntax that is typically used for spreadsheet formulas (e.g., expressions are preceded by a '=' sign) and has been enriched with elements that allow to represent RDF language elements (e.g., angle brackets are used to denote URIs). For example, (1) can be written as

```
=~rdf:type(<swrc:Publication>)          (2)
```

Tripcel is based on the spreadsheet metaphor, which means that expressions can be distributed across cells. Each cell is identified by a combination of letters (identifying the column) and a number (identifying the row). Assuming that (2) has been written into cell `C4`, we can re-use its results and apply another function to it, this time returning all `foaf:name` properties of all `dc:creator`s of the resources retrieved before:

```
=foaf:name(=dc:creator(=c4))            (3)
```

Similarly, the number of all publication's authors in the RDF graph can be identified by using an aggregate function:

```
=count(=dc:creator(=c4))                (4)
```

The Tripcel prototype has been implemented as a standalone application based on Java and the Jena Semantic Web framework[2]; hence every RDF graph that can be accessed through the Jena/ARQ API can be used as background graph. Currently, only reading RDF graphs from files is implemented; because of SPARQL's limited expressivity, Tripcel can operate on a background graph via SPARQL only in a limited manner (e.g., no aggregate functions). This can be solved either by using proprietary SPARQL extensions, or by adopting the needed features into the next SPARQL version.

Tripcel expressions can be edited through a user interface that resembles well-known spreadsheet applications (cf. Figure 1). The main window allows users to navigate through the grid of cells, and to directly edit formulas by clicking on the respective cell. Formulas are immediately evaluated after each change; for this purpose Tripcel maintains an in-memory dependency graph and efficiently propagates changes throughout the entire sheet. If an expression evaluates to exactly one result, the result is rendered directly in the cell. However, if a cell expression results in more than one value the number of results is displayed. In this case the detail window can be used to inspect the full list of results.

Tripcel maintains the formulas that were entered into a sheet separate from the background graph, hence allowing users to store the sheet using an RDF-based serialization and apply it to a different background graph without modifications.

3. CONCLUSIONS

Tripcel is a tool for the analysis of RDF data based on the spreadsheet paradigm, which allows users to iteratively compose complex queries, and immediately inspect their results. We have tested the usability of Tripcel against experts in the field of Semantic Web technologies who were asked to perform a tutorial and answer a questionnaire[3]. Our test persons considered the tool to be highly useful, especially for experts [1]. However, to lower the entry barrier for end users and to integrate calculations over RDF more tightly with other types of data we aim to further improve the tool and integrate it into standard office software.

4. REFERENCES

[1] B. Schandl. Functions over RDF Language Elements. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, 2009.

[2]Jena Semantic Web Framework: `http://jena.sourceforge.net`

[3]Available from `http://www.ifs.univie.ac.at/schandl/2009/06/tripcel`.