Querying and Semantically Integrating Spreadsheet Collections with XLWrap-Server

Use Cases and Mapping Design Patterns

Andreas Langegger
Johannes Kepler University Linz, Institute for Application Oriented Knowledge Processing
Altenberger Straße 69
4040 Linz, Austria
al@jku.at

Wolfram Wöß
Johannes Kepler University Linz, Institute for Application Oriented Knowledge Processing
Altenberger Straße 69
4040 Linz, Austria
wolfram.woess@jku.at

ABSTRACT

In this demo we will present XLWrap-Server, which is a wrapper for collections of spreadsheets providing a SPARQL and Linked Data interface similar to D2R-Server. It is based on XLWrap, a novel approach for generating RDF graphs of arbitrary complexity from spreadsheets with different layouts. To our best knowledge, XLWrap is the first spreadsheet wrapper, supporting cross tables and tables where data is not aligned in rows. It features a full expression algebra based on the syntax of OpenOffice Calc which can be easily extended by users and it supports Microsoft Excel, Open Document, and large CSV spreadsheets. XLWrap-Server can be used to integrate information from a collection of spreadsheets. We will show several use-cases and mapping design patterns in our demonstration.

1. INTRODUCTION

The translation of information stored in various legacy information systems to RDF is an important requirement of many Semantic Web applications. While for the Web of Data relational databases are considered to be the most important legacy data sources, in case of corporate Semantic Web applications, spreadsheets play a similar important role. Spreadsheets are frequently used by people in companies, organizations, and research institutions to share, exchange, and store data. Whenever there is no database in place, spreadsheets are often the primary fall-back tool for maintaining structured information.

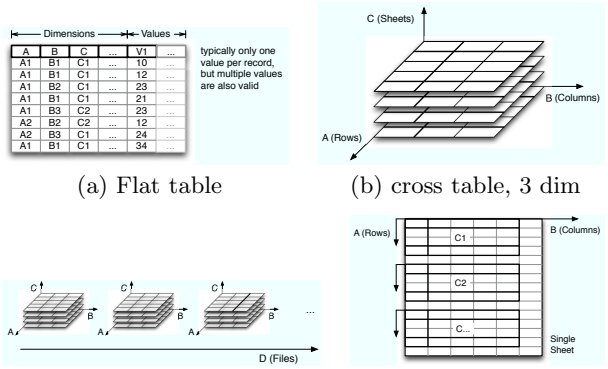
Currently available spreadsheet wrappers treat spreadsheets as flat tables like single database relations or CSV files. In this paper we will present a novel mapping approach for spreadsheets which is based on template graphs similar to RDF123 [2]. However, the XLWrap mapping approach is not based on a simple row oriented iteration of tables. It allows to define template mappings as RDF graphs and to repeat them based on various extensible shift and repeat operations in order to map arbitrary layouts including multi-dimensional cross tables and spreadsheets over multiple files. XLWrap supports expressions to reference cells and ranges from template graphs including sheet ranges and absolute references to other external spreadsheets.

XLWrap-Server provides a SPARQL endpoint as well as a Linked Data browser similar to D2R-Server [1]. It observes a configurable directory for mapping files and whenever a mapping file or one of the referred spreadsheet files of the mapping is added, modified, or removed, it automatically updates and caches the corresponding parts of the result graph. The setup procedure is a matter of starting the server and putting mapping files into the observation folder. XLWrap-Server is also considered to become a practical tool for experimenting with linked data. It can be used to quickly expose information via SPARQL while editing it in a human-friendly way. XLWrap can be downloaded from its homepage at `http://xlwrap.sourceforge.net`.

2. INFORMATION REPRESENTATION IN SPREADSHEETS

When mapping spreadsheets to RDF, it is important to distinguish between the *information model* and the *representation model*, which is used to represent information within a spreadsheet. The *information model* is defined implicitly by the semantics of the entailed information. The resulting RDF graph should as closely as possible reflect that information model, as for example, expenditures by category, year, and sub-division or personal information about employees. The actual information representation as an RDF graph is a subject of data modeling and there are many ways how to represent expenditures in RDF. XLWrap does not enforce any fixed rules and structures that depend on the representation of the information model in the spreadsheet. Concerning the *representation model*, three different layouts depicted in Figure 1 could be identified, whereas the third one is a hybrid approach of the first two.

One-dimensional flat table: In this layout (Figure 1(a)) information is represented, regardless of its dimensionality, in a flat table with a single column (*or* row) heading. It is used to represent flat data but also multi-



(a) Flat table

(b) cross table, 3 dim

(c) cross table, 4 dim

(d) cross table, 3 dim on a single sheet

Figure 1: spreadsheet information representation

dimensional information represented in de-normalized tables.

Cross tables: In this layout (Figure 1(b)) information is organized in cross tables, which may span multiple columns, rows, sheets, and even files and directories (Figure 1(c)). Instead of a single column/row header, cross tables have multiple headers, one for each dimension. Because a single sheet is already restricted to columns and rows, cross tables are often represented by repeating similar table layouts on the same sheet as depicted in Figure 1(d).

Hybrid layouts: Finally, there is the possibility of combining flat tables with cross tables (e.g. de-normalized tables can be repeated across several sheets or files).

Except for RDF123, all existing wrappers create exactly one RDF resource per row and to our best knowledge all existing wrappers, including RDF123, use a limited row-by-row based transformation approach. XLWrap supports hybrid layouts and is capable of translating multi-dimensional information, independent of the representation model, into arbitrary user-defined RDF graphs.

3. MAPPING APPROACH

A detailed description of the XLWrap mapping approach can be found in [3] and online at `http://xlwrap.sourceforge.net`. XLWrap mappings are based on template graphs, which may contain XLWrap expressions including cell references similar to a typical spreadsheet application such as Microsoft Excel or OpenOffice Calc. An example for such a template graph is depicted in Listing 1. Template graphs are repeatedly applied on one or more work sheets and files in order to produce the target graph. Depending on the representation of the information stored in the spreadsheet, which may be flat tables or cross tables over multiple sheets and files, each template graph is moved across sheets and subsequently applied for different combinations of cells. The way how template graphs are moved is specified by a sequence of transform operations. Currently, there exists shift operations for columns, rows, and sheets and repeat operations for a set of sheets or files. For each of these operations, an optional *cell range restriction* can be specified, which restricts the transform operation to a specific range, i.e. only range references within the restriction are transformed. Furthermore, an optional logical *condition* can be specified which is evaluated before a transformed template graph is applied. If it evaluates to false, the transform operation is skipped and XLWrap continues with the next stage of the following transform operation.

Listing 1: Example template graph in Turtle syntax.

```
:Revenues {
  [ xl:uri "'http://example.org/revenue_' &
    URLENCODE(
      SHEETNAME(A1) & '_' & B2 & '_' & A4
    )"^^xl:Expr ] a ex:Revenue ;
  ex:country    "DBP_COUNTRY(SHEETNAME(A1))"^^xl:Expr ;
  ex:year       "DBP_YEAR(B2)"^^xl:Expr ;
  ex:product    "A4"^^xl:Expr ;
  ex:itemsSold  "B4"^^xl:Expr ;
  ex:revenue    "C4"^^xl:Expr .
}
```

XLWrap supports the unification of anonymous instances (blank nodes) by the special property `xl:id`. All resources with equal IDs will get equal blank node IDs in the target graph.

The syntax of XLWrap expressions is similar to expressions in OpenOffice Calc and Microsoft Excel. The set of available functions can be extended easily by users. In this way, it is possible to attach to other software components, databases, etc. and dynamically influence the transformation process (e.g. value translations based on database tables).

4. MAPPING DESIGN PATTERNS

As part of the demo track we will show several use cases and introduce a collection of mapping design patterns we have started to publish online at `http://xlwrap.sourceforge.net/patterns.html`. The list is continuously updated based on the experiences of users of XLWrap.

Acknowledgments

This work is funded by the Austrian BMBWK (Federal Ministry for Education, Science and Culture), contract GZ BMWF-10.220/0002-II/10/2007.

5. REFERENCES

[1] R. Cyganiak and C. Bizer. D2R Server – Publishing Relational Databases on the Web as SPARQL Endpoints. In *Developers Track at the 15th International World Wide Web Conference (WWW2006)*, Edinburgh, Scotland, May 2006.

[2] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi. RDF123: From Spreadsheets to RDF. In *7th International Semantic Web Conference (ISWC2008)*, October 2008.

[3] A. Langegger and W. Wöß. XLWrap – Querying and Integrating Arbitrary Spreadsheets with SPARQL. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009), Washington D.C., LNCS*. Springer, 2009.