Processing RDF Data in UnifiedViews

Tomáš Knap[1,2] ⋆

[1] University of Economics, Prague, Czech Republic
[2] Charles University in Prague,
Faculty of Mathematics and Physics, Dept. of Software Engineering
Malostranské nám. 25, 118 00 Prague, Czech Republic
`tomas.knap@mff.cuni.cz`

Abstract. UnifiedViews is an Extract-Transform-Load (ETL) framework that allows users – publishers, consumers, or analysts – to define, execute, monitor, debug, schedule, and share RDF data processing tasks. The data processing tasks may use custom plugins created by users. UnifiedViews differs from other ETL frameworks by natively supporting RDF data and ontologies. The practical demonstration of UnifiedViews at the conference will (1) clearly demonstrate how UnifiedViews helps RDF/Linked Data users with RDF data processing (2) and show the real instance of UnifiedViews with tens of data processing tasks and DPUs motivated by real data processing use cases.

1 Introduction

There are lots of tools used by the RDF/Linked Data community[3], which may support various phases of RDF data processing; e.g., a user may use *any23*[4] for extraction of non-RDF data and its conversion to RDF data, *Virtuoso*[5] database for storing RDF data and executing SPARQL (Update) queries [1, 2], *Silk* [4] for RDF data linkage, or *Cr-batch*[6] for RDF data fusion.

Nevertheless, the user who is preparing a *data processing task* producing a data mart typically has to (1) configure every such tool using a different configurator, (2) implement a script for retrieving source data, (3) write his own script holding the set of SPARQL Update queries refining the data, (4) implement custom transformers which, e.g., enrich processed data with the data in his knowledge base, (5) write his own script executing the tools in the required order, so that every tool has all desired inputs when being launched, (6) prepare a scheduling script, which ensures that the task is executed regularly, and (7) extend his script with notification capabilities, such as sending an email in case of an error during task execution. Furthermore, in case of errors in data processing, the user has no support for debugging the tasks.

⋆ This work was supported by EU ICT FP7 under No.257943 (LOD2 project) and by projects SVV-2014-260100 and PRVOUK.

[3] http://semanticweb.org/wiki/Tools

[4] https://any23.apache.org/

[5] http://virtuoso.openlinksw.com/

[6] https://github.com/mifeet/cr-batch

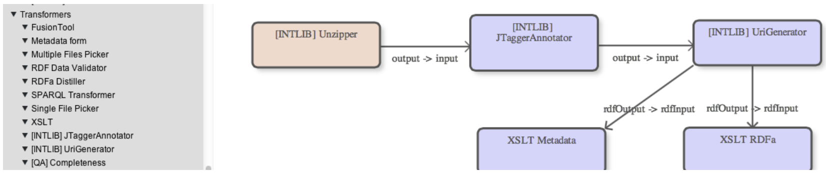

Fig. 1. UnifiedViews Framework – Definition of a Data Processing Task

2 UnifiedViews

To address these problems, we developed UnifiedViews, an Extract-Transform-Load (ETL) framework, where the concept of data processing task is a central concept. Another central concept is the native support for RDF data format and ontologies.

A *data processing task* (or simply task) consists of one or more data processing units. A *data processing unit* (DPU) encapsulates certain business logic needed when processing data (e.g., one DPU may extract data from a SPARQL endpoint or apply a SPARQL query). Every DPU must define its required/optional inputs and produced outputs. UnifiedViews supports exchange of RDF data between DPUs. Every tool produced by RDF/Linked Data community can be used in UnifiedViews as a DPU, if a simple wrapper is provided[7].

UnifiedViews allows users to define and adjust data processing tasks, using graphical user interface (an excerpt is depicted in Figure 1). Every user may also define their custom DPUs, or share DPUs provided by others together with their configurations. DPUs may be drag&dropped on the canvas where the data processing task is constructed. Data flow between two DPUs is denoted as an edge on the canvas (see Figure 1); a label on the edge clarifies which outputs of a DPU are mapped to which inputs of another DPU. UnifiedViews natively supports exchange of RDF data between DPUs; apart from that, files and folders may be exchanged between DPUs.

UnifiedViews takes care of task schedulling, a user may configure UnifiedViews to get notifications about errors in the tasks' executions; user may also get daily summaries about the tasks executed. UnifiedViews ensures that DPUs are executed in the proper order, so that all DPUs have proper required inputs when being launched. UnifiedViews provides users with the debugging capabilities – a user may browse and query (using SPARQL query language) the RDF inputs to and RDF outputs from any DPU. UnifiedViews allows users to share DPUs and tasks as needed.

Paper [3] contains discussion about related work and lists projects where UnifiedViews is used. The code of UnifiedViews is available at GitHub[8] under a combination of GPLv3 and LGPLv3 license[9].

[7] https://grips.semantic-web.at/display/UDDOC/Creation+of+Plugins
[8] https://github.com/UnifiedViews
[9] http://www.gnu.org/licenses/gpl.txt, http://www.gnu.org/licenses/lgpl.txt

3 UnifiedViews - The Demo

The demo of the tool is available at http://odcs.xrg.cz:8080/unifiedviews. You can use the account guest/guest to test the framework. When you log in, you can see tasks (menu item Pipelines) and DPUs (menu item DPU Templates) available in the framework. You can, for example, run *DBpedia* pipeline, which is extracting data about Prague from DBpedia, and supplement the pipeline with *SPARQL Transformer* DPU executing certain SPARQL (Update) queries on top of the extracted data.

The practical demonstration of UnifiedViews at the conference will (1) clearly demonstrate how UnifiedViews helps RDF/Linked Data users with RDF data processing (2) and show the real instance of UnifiedViews with tens of data processing tasks and DPUs motivated by real use cases.

References

1. S. H. Garlik, A. Seaborne, and E. Prud'hommeaux. SPARQL 1.1 Query Language. W3C Recommendation, 2013. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/, Retrieved 20/03/2014.
2. P. Gearon, A. Passant, and A. Polleres. SPARQL 1.1 Update. Technical report, W3C, 2013. Published online on March 21st, 2013 at http://www.w3.org/TR/2013/REC-sparql11-update-20130321/, Retrieved 20/03/2014.
3. T. Knap, M. Kukhar, B. Macháč, P. Škoda, J. Tomeš, and J. Vojt. UnifiedViews: An ETL Framework for Sustainable RDF Data Processing. In *Extended Semantic Web Conference (ESWC 2014)*, Anissaras, Crete, Greece, 2014. Springer.
4. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Silk - A Link Discovery Framework for the Web of Data. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web (LDOW)*, Madrid, Spain, 2009. CEUR-WS.org.