UnifiedViews: An ETL Framework for Sustainable RDF Data Processing

Tomáš Knap[1,2], Maria Kukhar[1], Bohuslav Macháč[1], Petr Škoda[1], Jiří Tomeš[1], and Ján Vojt[1] *

[1] Charles University in Prague,
Faculty of Mathematics and Physics, Dept. of Software Engineering
Malostranské nám. 25, 118 00 Prague, Czech Republic
[2] EEA s.r.o, Vlašská 349/15, 11800 Prague, Czech Republic
tomas.knap@mff.cuni.cz

Abstract. We present UnifiedViews, an Extract-Transform-Load (ETL) framework that allows users to define, execute, monitor, debug, schedule, and share ETL data processing tasks, which may employ custom plugins created by users. UnifiedViews differs from other ETL frameworks by natively supporting RDF data and ontologies. We are persuaded that UnifiedViews helps RDF/Linked Data consumers to address the problem of sustainable RDF data processing; we support such statement by introducing list of projects and other activities where UnifiedViews is successfully exploited.

1 Introduction and Basic Concepts of UnifiedViews

The advent of Linked Data [1] accelerates the evolution of the Web into an exponentially growing information space where the unprecedented volume of data offers information consumers a level of information integration that has up to now not been possible.

Suppose a consumer building a data mart integrating information from various RDF and non-RDF sources. There are lots of tools used by the RDF/Linked Data community[3], which may support various phases of the data processing; e.g., a consumer 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 [2, 3], *Silk* [5] for RDF data linkage, or *Cr-batch*[6] for RDF data fusion. Nevertheless, the consumer who is preparing a *data processing task* producing the desired data mart typically has to (1) configure every

* This work was supported by Seventh Framework Programme of the European Union under Grant Agreement number 611358 and by Specific Research Project at Charles University in Prague under number SVV-2014-260100.

[3] http://semanticweb.org/wiki/Tools
[4] https://any23.apache.org/
[5] http://virtuoso.openlinksw.com/
[6] https://github.com/mifeet/cr-batch

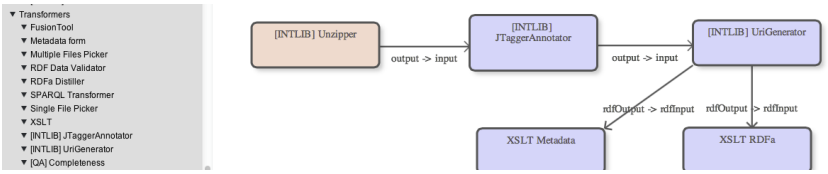

Fig. 1. UnifiedViews Framework – Definition of a Data Processing Task

such tool properly (using a different configuration for every tool), (2) implement a script for downloading and unpacking certain 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.

Maintenance of such data processing tasks is challenging. Suppose for example that a consumer defines tens of data processing tasks, which should run every week. Further, suppose that certain data processing task does not work as expected. To find the problem, the consumer typically has to browse/query the RDF data outputted by certain tool; to realise that, he has to manually launch the required tool with the problematic configuration and load the outputted RDF data to the store, such as Virtuoso, supporting browse/query capabilities. Furthermore, when other consumers prepare similar data processing tasks, they cannot use the already prepared scripts, they cannot use the tools' configurations already prepared by the consumer.

The general problem RDF/Linked Data consumers are facing is that they have to write most of the logic to define, execute, monitor, schedule, and share the data processing tasks themselves. Furthermore, consumers do not get any support regarding the debugging of the tasks. 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 consumer

[7] https://grips.semantic-web.at/display/UDDOC/Creation+of+Plugins

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.

The code of UnifiedViews is available at https://github.com/UnifiedViews/Core under a combination of GPLv3 and LGPLv3 license[8]. The demo of the tool is available at http://odcs.xrg.cz:8080/uv-demo. You can use the account eswc/eswc to work with the framework.

2 Related Work

There are plenty of ETL frameworks for preparing tabular data to be loaded to data warehouses, some of them are also opensource[9] – for example Clover ETL (community edition)[10]. In all these frameworks custom DPUs may be created in some way, but the disadvantage of these non-RDF ETL frameworks is that there is no support for RDF data format and ontologies in the framework itself. As a result, these non-RDF ETL frameworks are, e.g., not prepared to suggest ontological terms in DPU configurations, a feature important when preparing SPARQL queries or mappings of the table columns to RDF predicates. Furthermore, these frameworks do not have a native support for exchanging RDF data between DPUs; also the existing DPUs do not support RDF data format, URIs for identifying things according to Linked Data principles. Therefore, further, we discuss the related work in the area of RDF ETL frameworks.

ODCleanStore (Version 1)[11], was the original Linked data management framework, which was used as an inspiration for ODCleanStore (Version 2)[12], the student's project implemented at Charles University in Prague and defended in March 2014 . UnifiedViews is based on ODCleanStore (Version 2). Linked

[8] http://www.gnu.org/licenses/gpl.txt, http://www.gnu.org/licenses/lgpl.txt
[9] http://sourceforge.net/directory/business-enterprise/enterprise/data-warehousing/etl/
[10] http://www.cloveretl.com/products/community-edition
[11] http://sourceforge.net/projects/odcleanstore/
[12] https://github.com/mff-uk/ODCS/

Data Manager (LDM)[13] is a Java based Linked (Open) Data Management suite to schedule and monitor required ETL tasks for web-based Linked Open Data portals and data integration scenarios. LDM was developed by Semantic Web Company in Austria[14]. They currently decided to replace LDM, used by their clients, with UnifiedViews and further continue to maintain UnifiedViews together with the Czech Linked Data company, Semantica.cz[15].

DERI Pipes[16] is an engine and graphical environment for general Web Data transformations. DERI Pipes supports creation of custom DPUs; however, an adjustment of the core is needed when new DPU is added, which is not acceptable; in UnifiedViews, it is possible to reload DPUs as the framework is running. DERI Pipes also does not provide any solution for library version clashes; on the other hand, in UnifiedViews, DPUs are loaded as OSGi bundles, thus, it is possible to use two DPUs requiring two different versions of the same dependency (library) and no clashes arise. In DERI pipes, it is not possible to debug inputs and outputs of DPUs.

Linked Data Integration Framework (LDIF)[4] is an open-source Linked Data integration framework that can be used to transform Web data. The framework consists of a predefined set of DPUs, which may be influenced by their configuration; however, new DPUs cannot be easily added[17]. LDIF provides user interface to monitor results of executed tasks.; however, when compared with UnifiedViews, LDIF does not provide any graphical user interface for defining and scheduling tasks, managing DPUs, browsing and querying inputs from and output to the DPUs, and managing users and their roles in the framework. LDIF also does not provide any possibility to share pipelines/DPUs among users. On the other hand, LDIF provides possibility to run tasks using Hadoop[18].

3 Impact of the UnifiedViews Framework

The goal of the *OpenData.cz initiative*[19] is to extract, transform and publish Czech open data in the form of Linked Data, so that the initiative contributes to the Czech Linked (Open) Data cloud. For this effort, UnifiedViews framework is successfully used since September 2013.

Project INTLIB[20] aims at extracting (1) references between legislation documents, such as decisions and acts, (2) entities (e.g., a citizen, a president) defined by these documents and (3) the rights and obligations of these extracted entities. UnifiedViews is used in INTLIB to extract data from selected sources of legislation documents, convert it to RDF data, and provide it as Linked Data.

[13] https://github.com/lodms/lodms-core
[14] http://www.semantic-web.at
[15] http://semantica.cz/en/
[16] http://pipes.deri.org/
[17] http://ldif.wbsg.de/
[18] http://hadoop.apache.org/
[19] http://opendata.cz
[20] http://www.isvav.cz/projectDetail.do?rowId=TA02010182

COMSODE FP7 project[21] has the goal to create a publication platform for publishing (linked) open data. UnifiedViews is used there as the core tool for converting hundreds of original datasets to RDF/Linked Data.

UnifiedViews framework is being integrated to the stack of tools produced by the *LOD2 project*[22]. As a result, anybody using tools from LOD2 stack, such as Virtuoso and Silk, has also the possibility to use UnifiedViews.

UnifiedViews framework is intended to be used for commercial purposes by companies Semantica.cz, Czech Republic, and Semantic Web Company, Austria, to help their customers to prepare and process RDF data.

4 Conclusions

We presented UnifiedViews, an ETL framework with a native support for processing RDF data. The framework allows to define, execute, monitor, debug, schedule, and share data processing tasks. UnifiedViews also allows users to create custom plugins - data processing units. We are persuaded that UnifiedViews is a matured tool, which addresses the major problem of RDF/Linked Data consumers – the problem of sustainable RDF data processing; we support such statement by introducing list of projects where UnifiedViews is successfully used and mention two commercial exploitations of the tool.

The practical demonstration of UnifiedViews at the conference will clearly demonstrate how UnifiedViews can help RDF/Linked Data consumers and show the real instance of UnifiedViews with tens of data processing tasks and DPUs motivated by real use cases.

References

1. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1 – 22, 2009.
2. 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.
3. 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.
4. A. Schultz, A. Matteini, R. Isele, C. Bizer, and C. Becker. LDIF : Linked Data Integration Framework. In *Proceedings of the Second International Workshop on Consuming Linked Data (COLD)*, Bonn, Germany, 2011. CEUR-WS.org.
5. 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.

[21] http://www.comsode.eu/
[22] http://lod2.eu/