[Semantic Data Web Technologies Lab, University of Bonn, 2016]

Semantic Lifting Of Budget Data

Software requirements for ABuDaT (administrative budget data transformer)

[Group: E, Mentor: Fathoni Musyaffa]

Students: Florian Weile, Tatiana Novikova, Aberham Gebreyohannes
Samuel Y. Ayele

Table of Contents

1. Introduction

This document presents the requirements specification for ABuDaT and describes the overall design including functional and nonfunctional requirements of the system.

2. List of acronyms

ABuDaT - administrative budget data transformer

UML - Unified Modeling Language

CSV - comma separated values file

XML - Extensible Markup Language

ETL - Extract Transform Load

RDF - Resource Description Framework

OLAP - Online Analytical Processing

3. Description

ABuDaT is a software system that enables users to transform real world administrative budget data into the openbudgets.eu RDF data model. Transforming the raw data published by the authorities from various formats such as tabular (.csv) or hierarchical (.xml) and in various languages involves multiple steps which are described in this document.

Subject: User Interface for transformation.
Purpose: To foster transparency in administrative budgeting, make budgets comparable on a global scale and improve participatory budgeting by the citizen.
Object: governmental raw data published in various formats and different languages.

The overall goal is to have budget data available in openbudgets.eu RDF data format.

4. ETL Process

ABuDaT will implement an ETL Process. ETL (Extract Transform Load) refers to a process in the field of data warehousing. It involves three steps:

1. **Extraction** of data from possibly heterogeneous data sources.
2. **Transformation** of the data for storing in a certain target format, in order to be able to query and analyse it.
3. **Loading** it into some target data store [MA].

ABuDaT's main focus is on the first phase of the ETL process, the extraction. Considering budget data in various formats and various languages, our goal is to provide an easy to use interface for the extraction and also for transformation of the important fields from these datasets. The target format for ABuDaT is the OpenBudgets.eu RDF data model [OBeu1]. The transformed data will then be validated through available open source tools as a backend and loaded into an RDF triple store - a purpose-built database for the storage and retrieval of triples through semantic queries [Rush].

ABuDaT will use the open-source ETL cycle tool LinkedPipes http://etl.linkedpipes.com/ as a backend trough it's REST API. LinkedPipes realizes a transformation process that consists of interconnected components [Lin].

5. Openbudgets.eu data model

The opendudgets.eu data model is the result of a survey and knowledge elicitation performed in cooperation with domain experts. It uses the RDF Data Cube Vocabulary, a data structure representing an OLAP data cube using the RDF format. The vocabulary defines a schema made up of classes and properties specifying the structure of the data to be published [OBeu2, PSA].

The Data Cube vocabulary is focused purely on the publication of multidimensional data on the web. The OpenBudgets.eu data model includes the following components:

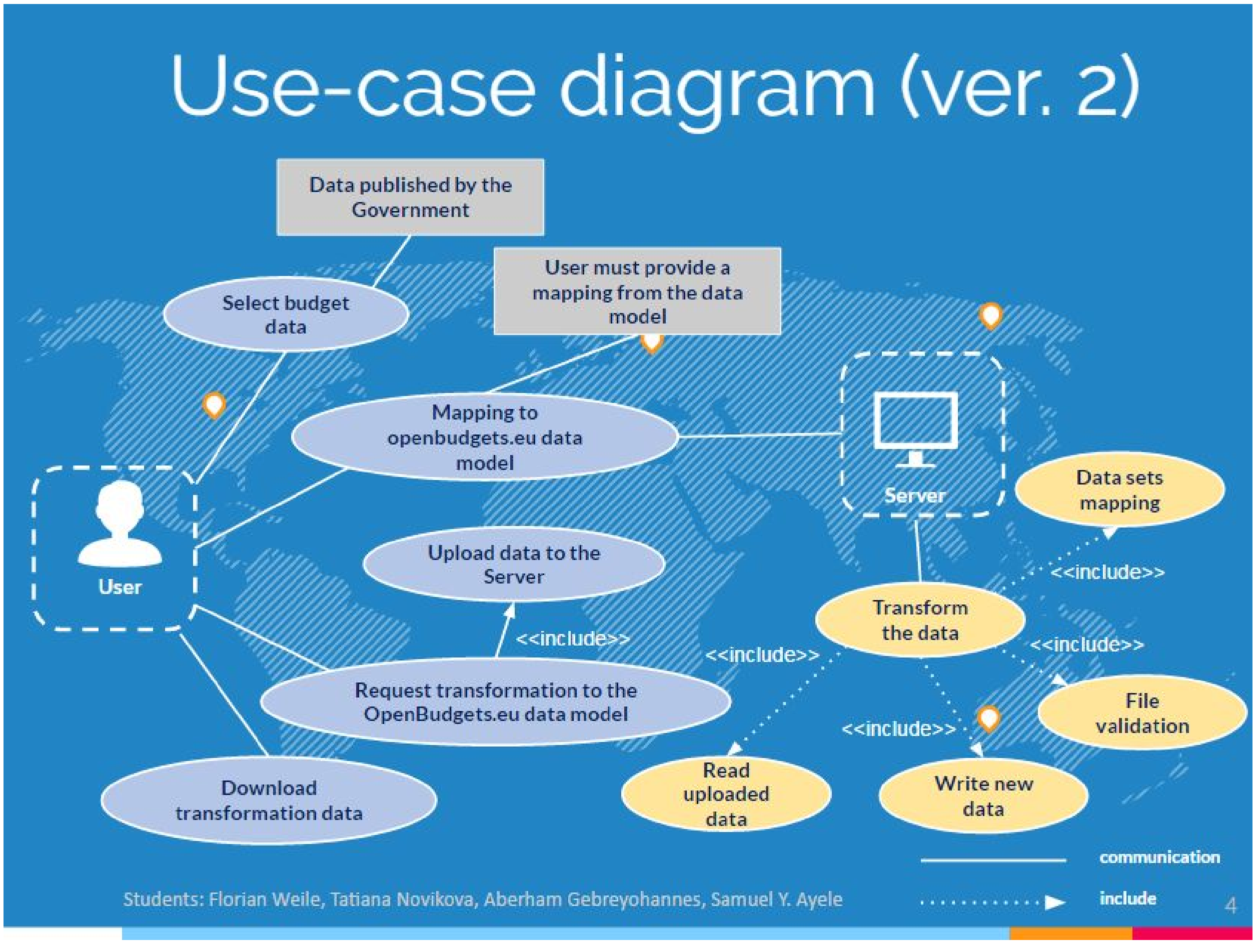
- **A semantic data model.** The semantic data model for (open) budget data integrates datasets. This integration allows for analysis and comparison across government levels, regions, and countries. These comparisons and analysis shed light on anomalies within and between datasets.
- **Libraries of visualisation, data-mining, and comparative analysis tools.** Visualisation tools help to present data in different granularity and modalities (spatial, temporal, administrative), clarifying and strengthening the stories and contributing to the analysis [OBeu3].

6. Use Case Diagram

User: A UML actor representing the role of a person, object or device that interacts with the system.

Server: A UML actor representing the role of the server.

Use Case Diagram: A UML behavior diagram that visually describes the functional requirements of a proposed system and shows the relationships between User, Server and Use Cases with the help of Associations (connectors).



Use-case diagram (ver. 2)

Data published by the Government

User must provide a mapping from the data model

Select budget data

Mapping to openbudgets.eu data model

Upload data to the Server

Request transformation to the OpenBudgets.eu data model

Download transformation data

User

Server

Data sets mapping

Transform the data

File validation

Read uploaded data

Write new data

<<include>>

<<include>>

<<include>>

<<include>>

<<include>>

communication

include

Students: Florian Weile, Tatiana Novikova, Aberham Gebreyohannes, Samuel Y. Ayele

4

Figure 1: Use-Case Diagram

7. Use Case Scenario

Budget data is being published by the government in various formats (e.g. xml, csv) and is written in the official language of the respective countries. The user may download the budget data.

Transforming the data involves multiple steps:

1. **Selecting the budget data (User)**
   - The user will be able to select a file from his / her local storage in order to upload it to the system. Alternatively the user will be able to provide a URL as input to the system.
2. **Providing a mapping (User)**
   - The user will have to provide a mapping from the model and name space of the respective input data to the OpenBudgets.eu data model.
3. **Requesting the transformation (User)**
   - **Uploading the file**
     - In order to transform the data, the file will be uploaded to the server. Alternatively, the URL provided by the user will be sent to the server.
4. **Transforming the data (Server)**
   - **Read the data**
     - The server will read the data provided by the user.
   - **Read the mapping**
     - The server will read the data mapping provided by the user.
   - **Apply the mapping**
     - The server will apply the mapping to the data.
   - **Write the result**
     - The server will write the resulting RDF data to file.
   - **Validate the result**
     - The server will validate the resulting RDF data.
5. **Download the transformed data (User)**
   - The user will be able to download the data

8. Functional Requirements

REQ_ID	REQ_NAME	DESCRIPTION	PRIORITY
1	Data Transformation	Use open-source tools to perform the extraction, transformation, loading, and quality assurance of the datasets (OBEU RDF data model).	1
2	User Interface	Implement an easy to use web interface.	1
3	Data export	Data export into OBEU RDF format.	1
4	Validation of output	Validation of the transformation result using an open source RDF-Data-Cube-Validator	2
5	Publication of output	Publish transformation results to a triple store provided by the customer	2
6	Data Sets	Data need to be interlinked with Linked Open Data sets (e.g. DBpedia)	3

9. Non-Functional Requirements

REQ_ID	REQ_NAME	DESCRIPTION	PRIORITY
1	**Usability**	User-friendly interface. The UI should be intuitive and easy to use even for novice users.	1
2	**Technical System Documentation**	The documentation should contain architecture, functionality and usage.	1
3	**Robustness**	The UI should be able to deal with errors and abnormalities in the input data.	2
4	**Open Source**	The system should be release under an open source license for further development by others	1

10. References

[Lin] http://etl.linkedpipes.com/documentation/

[MA] Larissa Terpeluk Moss, S. Atre, Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-support Application, Addison-Wesley Professional, 2003

[OBeu1] http://openbudgets.eu/assets/deliverables/D1.2.pdf

[OBeu2] http://openbudgets.eu/assets/deliverables/D1.1.pdf

[OBeu3] http://openbudgets.eu/about/objectives/

[PSA] Tassilo Pellegrini, Harald Sack, Sören Auer, Linked Enterprise Data: Management und Bewirtschaftung vernetzter Unternehmensdaten mit Semantic Web Technologien, Springer 2014

[Rush] TripleStore, Jack Rusher, Semantic Web Advanced Development for Europe (SWAD-Europe), Workshop on Semantic Web Storage and Retrieval - Position Papers