Data transformations

Miroslav Blaško, Petr Křemen

October 17, 2019

1 Background

The goal of this seminar is to get familiar with data transformations and in transforming CSV files into RDF, in particular. For this purpose we will use *OntoRefine*, a GraphDB variant of OpenRefine supporting creation of RDF output.

2 Data Transformations with OpenRefine

Within this task we will transform two sheets from an XLSX document into RDF which we will use for querying. Concrete steps of the task are:

Ex. 1 — Login into GraphDB at `http://onto.fel.cvut.cz:7300` and import sheet "event type x factor" provided by XLSX document into OntoRefine within GraphDB (Hint: `/GraphDB/Import/Tabular (OntoRefine)`).

Ex. 2 — Fix issues with spelling/capitalization of "Eccairs event description" using cluster & merge method. (Hint: `ECCAIRSEvent description/Edit cells ../Cluster and edit ...`).

Ex. 3 — Analyze "Source of model description (if relevant)"column using Text facet and filtering – explore functions of this tool. (Hint: `Source of model description (if relevant)/Facet/Text facet`).

Ex. 4 — Add source type column based on different values of the analyzed column. (Hint: `Source of model description (if relevant)/Edit column/Add column based on this column ...`). You can use OpenRefine Expression Language to define new column in the following way:

```
value.replace(/^http[s]?:../,"").split("/")[0].replace(/www./,'').split('.')[0].
    replace(/$/,' documentation')
```

Ex. 5 — Remove all irrelevant rows (Hint: `Source of model description (if relevant)/Facet/Text facet`, pick blank to include only in filtering, then `All/Edit rows/Remove all matching rows`).

Answer (Ex. 5) — Solution to all the tasks up to here will be uploaded to the course web site together with this document.

Ex. 6 — Export the project into a SPARQL endpoint. (Hint: RDF button + Data/Get SPARQL endpoint).

Ex. 7 — Create new OpenRefine project by importing sheet "uniset factors" provided by XLSX, transform it appropriately and export it as a second SPARQL endpoint.

Ex. 8 — Use both SPARQL endpoints to construct a list of events together with their uniset factors.

Answer (Ex. 8) — A query similar to the following is the solution. Adjust the prefixes, column names (RDF predicates) and endpoint URLs to your case.

```
PREFIX s4u: <http://onto.fel.cvut.cz/ontologies/osw2018/s4/u/>
PREFIX s4ef: <http://onto.fel.cvut.cz/ontologies/osw2018-seminar-4/ef/>
PREFIX spif: <http://spinrdf.org/spif#>

# Example query returning RDF data
SELECT  ?eventType ?factorType {
  SERVICE <ontorefine:2132197441269> {
    SELECT DISTINCT ?factorTypeId {
      [] a s4u:Row ;
         s4u:EccairsEventFactorId ?factorTypeId .
    }
  }

  SERVICE <ontorefine:2174942986409> {
    SELECT DISTINCT ?eventType ?factorType ?factorTypeId {
      [] a s4ef:Row ;
         s4ef:EccairsEventFactorId ?factorTypeId ;
         s4ef:ECCAIRSEvent_description ?eventType ;
         s4ef:ECCAIRSEventFactor_description ?factorType ;
    }
  }
}
```

Ex. 9 — As another exercise you can solve the ČSSZ integration task from the first tutorial using OpenRefine and check the results.

3 Other related tools

- **S-pipes** – available at `https://kbss.felk.cvut.cz/gitblit/tree/s-pipes.git`.
- **RDFpro** – available at `http://rdfpro.fbk.eu/`.
- **ETL LinkedPipes** – available at `https://etl.linkedpipes.com/`.