CLARIN-DSS “Dutch Ships and Sailors”

DELIVERABLE D11: DOCUMENTATION OF TOOL CHAIN

Author:
Victor de Boer (VU Amsterdam)

Deliverable type: SOFTWARE

Version	Date	Author	Description
0.1	16-10-2013	Victor de Boer	First version
0.2	19-4-2014	V. de Boer	Tools described

This deliverable describes the tool chain used in the conversion of the datasets.

TOOLS

We here build on and include previous work as described in de Boer et al. (2012)[1]

To convert the datasets into Linked Data, we here describe the general methodology. The result of the workflow process is the collection metadata in semantic format (RDF). Links are established between vocabulary terms used in the collections.

The methodology is built on the ClioPatria semantic server (http://cliopatria.swi-prolog.org). ClioPatria provides feedback to the user about intermediary or final RDF output in the form of an RDF browser and by providing statistics on the various RDF graphs. This feedback is crucial for the intended interactivity. The approach takes the form of a modular workflow, supported by two tools.Both the XMLRDF and Amalgame are packages of the ClioPatria semantic web toolkit. ClioPatria itself is based on SWI-Prolog and XMLRDF can therefore use its expressiveness for more complex conversions.

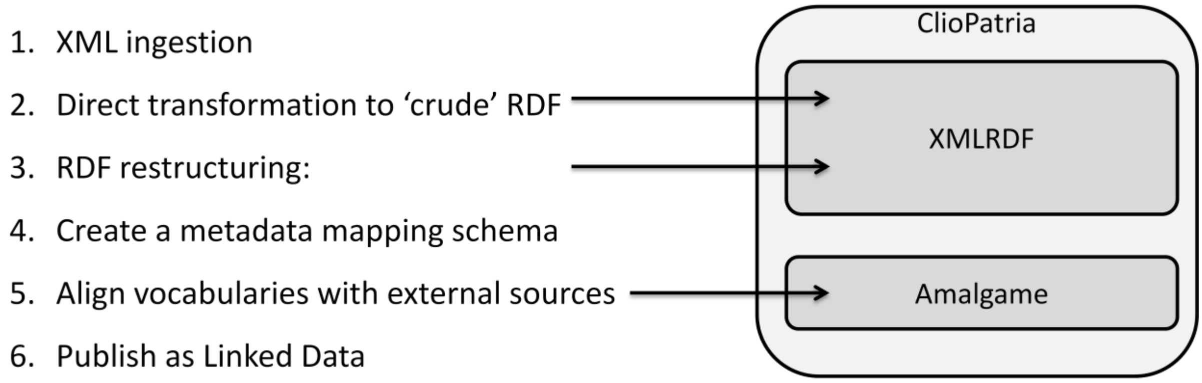
1. XML ingestion
2. Direct transformation to ‘crude’ RDF
3. RDF restructuring:
4. Create a metadata mapping schema
5. Align vocabularies with external sources
6. Publish as Linked Data
ClioPatria
XMLRDF
Amalgame

FIG 1: TOOL CHAIN

[1] Victor de Boer, Jan Wielemaker, Judith van Gent, Michiel Hildebrand, Antoine Isaac, Jacco van Ossenbruggen, Guus Schreiber. Supporting Linked Data Production for Cultural Heritage institutes: The Amsterdam Museum Case Study. In: Proceedings of the 9th Extended Semantic Web Conference (ESWC 2012) Heraklion, Greece. May 27 – 31

In the first step of this workflow, we ingest the XML into the ClioPatria environment. In the second step, the XML is converted to crude RDF format. This is done using the XMLRDF tool, which is documented below. This crude RDF is then rewritten in RDF adhering to the chosen metadata format, which is done using graph rewrite rules. These rules are executed by the XMLRDF tool to produce the final RDF representation of the collection metadata

Next, the user can provide an RDFS metadata schema which relates the produced classes and properties to the metadata schema of choice. The XMLRDF tool provides support for this by presenting the user with a schema template based on the RDF data loaded. In Step 5, links are established between vocabulary concepts that are used in the collection metadata and other vocabularies. This is done using the Amalgame tool, (http://semanticweb.cs.vu.nl/amalgame [2].

XMLRDF

[From http://semanticweb.cs.vu.nl/xmlrdf/]

XMLRDF is designed for converting XML documents with a fairly consistent structure, such as database dumps, to RDF. The key idea is to split the process into two steps:

1. Create RDF using a generic lossless conversion process
2. Apply a set of rewrite rules to turn the generated RDF into a proper semantic model.

The tool relies on ClioPatria's capabilities for exploring both the initial conversion and the final RDF datasets. The rule-set is written in an abstract committed-choice language. This language is compiled into Prolog rules for execution. The abstract language provides the possibility to use arbitrary Prolog code, both for guarding the ruleset and (notably) for rewriting literals.

The scripts

The XMLRDF scripts used in this project are found online at https://github.com/biktorrr/dss/tree/master/script

Amalgame

[from http://semanticweb.cs.vu.nl/amalgame]

Amalgame (AMsterdam ALignment GenerAtion MEtatool) is a tool for finding, evaluating and managing vocabulary alignments. We explicitly do not aim to produce 'yet another alignment method' but rather seek to combine existing matching techniques and methods such as those developed within the context of the Ontology Alignment Evaluation Initiative (OAEI), in which different alignment methods can be combined using a workflow setup. The Amalgame Alignment server features

1. A workflow composition functionality, where various alignment generators can be positioned. Their resulting mapping sets can be used as input for filtering methods, other alignment generators or combined into overlap sets.
2. A statistics function, where statistics for alignment sets will be shown
3. An evaluation tool, where subsets of alignments can be evaluated manually

[2] van Ossenbruggen, J., Hildebrand, M., de Boer, V.: Interactive vocabulary alignment. In Gradmann, S., Borri, F., Meghini, C., Schuldt, H., eds.: TPDL. Volume 6966 of Lecture Notes in Computer Science., Springer (2011) 296–307

Provenance

The provenance of the data is documented using the Provenance ontology (http://www.w3.org/TR/prov-o/). The provenance of DSS can be found at http://www.dutchshipsandsailors.nl/data/browse/list_graph?graph=http://purl.org/collections/nl/dss/dss_provenance.ttl . We list the Prov-O software agents and human agents below in their RDF Turtle form:

```
<http://cliopatria.swi-prolog.org/>
        a prov:SoftwareAgent ;
        rdfs:label "ClioPatria" .

packs:amalgame
        a prov:SoftwareAgent ;
        rdfs:label "Amalgame alignment platform" .

packs:xmlrdf
        a prov:SoftwareAgent ;
        rdfs:label "XMLRDF conversion tool" .

<AndreaBravoBalado>
        a prov:Agent ,
          foaf:Person ;
        rdfs:label "Andrea Bravo Balado" ;
        foaf:pastProject <http://dutchshipsandsailors.nl> .

<JurLeinenga>
        a prov:Agent ,
          foaf:Person ;
        rdfs:label "Jur Leinenga" ;
        foaf:pastProject <http://dutchshipsandsailors.nl> .

<MatthiasVanRossum>
        a prov:Agent ,
          foaf:Person ;
        rdfs:label "Matthias van Rossum" ;
        foaf:pastProject <http://dutchshipsandsailors.nl> .

<RobinPonstein>
        a prov:Agent ,
          foaf:Person ;
        rdfs:label "Robin Ponstein" ;
        foaf:pastProject <http://dutchshipsandsailors.nl> .

<VictorDeBoer>
        a prov:Agent ,
          foaf:Person ;
        rdfs:label "Victor de Boer" ;
        owl:sameAs <http://www.few.vu.nl/~vbr240/foaf.rdf> ;
        foaf:pastProject <http://dutchshipsandsailors.nl> .
```

Misc. scripts

The current toolchain expects XML input. However, in some cases other types of input are provided and we use simple scripts to produce XML versions of the input data. More specifically we use

1. A csv to XML script. A version is implemented in Python and can be found at https://github.com/biktorrr/dss/tree/master/script/CsvTwoXml
2. Excel to XML. MS Excel has a built in functionality to save a worksheet as XML
3. SQL dump to XML: We used a XAMPP server to rebuild a SQL database based on the SQL dump provided. The XAMPP server software provides built in export to XML