Extraction and Semantic Annotation of Workshop Proceedings in HTML using RML

Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Laurens De Vocht, Ruben Verborgh, Erik Mannens, and Rik Van de Walle

Ghent University - iMinds - Multimedia Lab,
firstname.lastname@ugent.be

Abstract. Despite the significant number of existing tools, incorporating data into the Linked Open Data cloud remains complicated; hence discouraging data owners to publish their data as Linked Data. Unlocking the semantics of published data, even if they are not provided by the data owners, can contribute to surpass the barriers posed by the low availability of Linked Data and come closer to the realisation of the envisaged Semantic Web. RML, a generic mapping language based on an extension over R2RML, the W3C standard for mapping relational databases into RDF, offers a uniform way of defining the mapping rules for data in heterogeneous formats. In this paper, we present how we adjusted our prototype RML Processor, taking advantage of RML's scalability, to extract and map data of workshop proceedings published in HTML to the RDF data model for the Semantic Publishing Challenge needs.

1 Introduction

Data owners lack of incentives to publish their data in a format processable by Semantic Web clients, partly because incorporating data into the Linked Open Data still remains complicated. *Generic solutions* fail to efficiently support them, as it is impossible to predict every potential input, while *case-specific solutions*, in their turn, need individual investment and they are not reused at the end. Furthermore, most of the existing solutions are *source-specific*. Only few tools provide mappings from different source formats to RDF; but even those tools actually employ separate *source-centric approaches* for each of the supported formats. Thus, whenever a new need to map data from a source in an arbitrary format emerges, the whole implementation is developed from scratch.

The low availability of Linked Data, mainly caused by data owners who do not publish their data as Linked Data for different reasons, remains a barrier to the realisation of the Semantic Web. There are a lot of data published as Open Data but even more data are published on Web pages and only a few of them have any semantic annotations. Unlocking the semantics of this data is of high importance if we want to be able to query their content. Therefore, we need solutions that allow us easily to get the published data in RDF, even if the data providers do not publish them as such.

Regarding data published in Web pages, most of the proposed solutions rely on the page's DOM or on processing the HTML source as XML documents which implies that they should be well-formatted documents. To deal with invalid HTML documents, Coetzee [1], for instance, balances the tags and validates the model before performing the mappings. However prior cleansing and *re-formatting* is not always possible, especially when performing mappings *on-the-fly*.

In an effort to address, among others, the aforementioned issues, we defined RML [2] in our previous works. In the frame of the Semantic Publishing challenge[1], selected computer science workshop proceedings published with the CEUR-WS.org open access service were mapped in RDF in order to answer more complicated queries related to the quality of the workshops. To address the challenge of semantically annotating the content of HTML pages, we exploited and proved RML's extensibility and flexibility. Our RML Processor implementation[2], which was configured so far to map data in CSV, XML and JSON formats, was extended further to support mapping of data in HTML to the RDF data model.

2 Defining RML mappings

The *RDF Mapping language* (RML)[3] is a generic language defined to express customized mapping rules from data in heterogeneous formats to the RDF data model [2]. RML is defined as a superset of the W3C-standardized mapping language R2RML, extending its applicability and broadening its scope. RML keeps the mapping definitions as in R2RML and follows the same syntax, providing a generic way of defining the mappings that is easily transferable to cover references to other data structures, combined with case-specific extensions. RML considers that sets of sources that all together describe a certain domain, can be mapped to RDF in a combined and uniform way, while the mapping definitions may be re-used across different sources that describe the same domain.

Structure of an RML mapping

In RML, the mapping to the RDF data model is based on one or more Triples Maps. A Triples Map consists of three main parts: the Logical Source (`rr:LogicalSource`), the Subject Map and zero or more Predicate-Object Maps. The Subject Map (`rr:SubjectMap`) defines the rule that generates unique identifiers (URIs) for the resources which are mapped and is used as the subject of all the RDF triples that are generated from this Triples Map. A Predicate-Object Map consists of Predicate Maps, which define the rule that generates the triple's predicate and Object Maps or Referencing Object Maps, which defines the rule that generates the triple's object. The Subject Map, the Predicate Map and the Object Map are Term Maps, namely rules that generate an RDF term (an IRI, a blank node or a literal).

[1] `http://2014.eswc-conferences.org/semantic-publishing-challenge`
[2] `https://github.com/mmlab/RMLProcessor`
[3] `http://semweb.mmlab.be/rml`

Leveraging HTML with RML

A Logical Source (`rml:LogicalSource`) is used to determine the input source with the data to be mapped. RML deals with different data serializations which use different ways to refer to their content. Thus RML considers that any reference to the Logical Source should be defined in a form relevant to the input data, e.g. XPATH for XML files or JSONPATH for JSON files. The Reference Formulation (`rml:referenceFormulation`) indicates the formulation (for instance, a standard or a query language) to refer to its data. Any reference to the data of the input source must be valid expressions according to the Reference Formulation defined at the Logical Source. This makes RML highly extensible towards new source formats.

At the current version of RML, the `ql:CSV`, `ql:XPath` and `ql:JSONPath` Reference Formulations are predefined while the `ql:CSS3` was introduced for the challenge's needs as we chose the Selectors Level 3 expressions (CSS3)[4] to access the elements within the document. CSS3 selectors are standardized by W3C, they are easily used and broadly-known as they are used for selecting the HTML elements both for cascading styles and for jQuery[5]. CSS3 selectors can be used not only to refer to data in HTML documents but they could also be used for XML documents.

Defining RML documents for CEUR proceedings

```
<a href="http://salad2013.linkedservices.org/">
  <span class="CEURVOLACRONYM">SALAD 2013</span></a>
<h3><span class="CEURVOLEDITOR">Ruben Verborgh</span><br>
    <span class="CEURVOLEDITOR">Maria Maleshkova</span><br>
    ....</h3>

<#VolumeMapping>
  rr:subjectMap
  [ rr:template "http://ceur-ws.org/{span.CEURVOLNR}/";
    rr:class bibo:Volume ];
  rr:predicateObjectMap
  [ rr:predicate bibo:shortTitle;
    rr:objectMap [ rml:reference "span.CEURVOLACRONYM"; ] ];
  rr:predicateObjectMap
  [ rr:predicate bibo:editor;
    rr:objectMap [ rr:template "http://ceur-ws.org/person/{span.CEURVOLEDITOR}"; ] ].

<http://ceur-ws.org/Vol-1056/> a bibo:Volume ;
  bibo:shortTitle "SALAD 2013" ;
  bibo:editor <http://ceur-ws.org/person/Ruben%20Verborgh> ;
  bibo:editor <http://ceur-ws.org/person/Maria%20Maleshkova> ;
  ....
```

Listing 1.1. An extract of Vol-996 of CEUR proceedings, following by an extract of the RML document that generates the triples specified.

The vocabularies used to describe the domain were selected to be aligned with the annotations provided in the case of volumes that already included RDFa annotations and considering vocabularies relevant to the domain as listed at `http://linkeduniversities.org/lu/index.php/vocabularies/`. The RML document for the *challenge* can be found at `http://ewi.mmlab.be/spc_dataset/mappings/`.

[4] `http://www.w3.org/TR/selectors/`

[5] `http://jquery.com`

3 Performing Mappings to RDF with RML

Defining and executing a mapping with RML requires the user to provide an *input dataset* to be mapped and the *mapping document* according to which the mapping will be executed to generate the corresponding RDF *output dataset*. Data cleansing is out of RML's scope and should be performed in advance. Baring in mind that such data cleansing is not always possible, e.g. mapping live HTML documents *on-the-fly*, regular expressions were preferred to be used whenever it is required to be more selective over the returned values. For instance, a reference to `h3 span.CEURLOCTIME` returns `Montpellier, France, May 26, 2013` for the aforementioned example and, as there is no further HTML annotation, regular expressions are required to select parts of the returned value to be mapped separately(e.g. city).

Performing HTML to RDF mappings with the RML processor

Our prototype RML processor[6], implemented in Java, was used but, for the challenge needs, we extended it to leverage also HTML documents. We used CSSelly[7], a Java implementation of the W3C CSS3 specification. The HTML documents were stored locally and mapped as the RML processor was implemented so far with the scope of mapping files owned by data publishers and existing locally to the system. The definition of RML though allows to refer to resources even if they are published on the web and be retrieved as Web resources instead of local files.

The core functionality of the processor is used as such, we only added the CSS3 selectors to access the HTML input. Each defined Triples Maps is processed in a consecutive order and the defined Subject Map and Predicate-Object Maps are applied. For each reference to the input HTML, the HTML extractor returns an extract of the data. If a regular expression is specified, it is applied over the returned value and the corresponding triples are generated. The output dataset for the challenge can be found at `http://ewi.mmlab.be/spc_dataset`.

4 Discussion and Conclusions

It is beneficial that CSS3 selectors become part of a formalisation that performs mappings of data in HTML. Considering that the RML processor takes care of executing the mappings while the CSS3 extractor parses the document, the data publishers' contribution is limited in providing only the mapping document. As RML enables the re-use of the same mappings over different files, the effort they put is even less. In the case of the challenge, the same mapping documents were used to define the mappings for different HTML input sources.

This happens because most of the websites use templates thus the content of their pages is structured in a similar way, which is defined using CSS3 selectors, the same point of reference as the one used by RML. This allows us to use RML

[6] `https://github.com/mmlab/RMLProcessor`
[7] `http://jodd.org/doc/csselly/`

mapping documents as a "translation layer" over the published content and extract the content. Furthermore, as the mappings are partitioned in independent Triples Maps, data owners can select the Triples Maps they want to execute at any time. This provides them with the flexibility to execute only a part of the mappings at any time. For instance, if they identify a faulty mapping to their RDF output, they can isolate the Triples Map that generated those triples, correct it and re-execute it without affecting the rest of the dataset.

This becomes even more valuable considering that the mappings in RML are defined as triples themselves. The triples' provenance can be tracked and used to identify the mappings and data that cause the "faulty" RDF result [3]. Last, the mapping rules are interoperable; any tool that supports RML can process them either to execute them, as our RML Processor does or to refine them, e.g. by importing them to an application, such as Karma[8] or OpenRefine[9].

Beyond re-using the same mapping documents, data publishers can combine data from different input sources either they are in the same format or not. This leads to enhanced results as integration of data from different sources occurs during the mapping and relations between data appearing in different resources can be defined instead of interlinking them afterwards. For instance, the proceedings appearing in HTML can be mapped in an integrated fashion with the XML versions of the papers published at the workshops, enriching the resulting dataset with properties defined considering the combination of the two documents.

To sum up, this solution proves the scalability of the RML, as it was successfully extended to define mappings from data in HTML to the RDF data model.

Acknowledgments.

The described research activities were funded by Ghent University, the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research Flanders (FWO Flanders), and the European Union.

References

1. P. Coetzee. Sparqplug: Generating linked data from legacy html, sparql and the dom, 2008.
2. A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle. Rml: A generic language for integrated rdf mappings of heterogeneous data. In *Workshop on Linked Data on the Web*, 2013.
3. A. Dimou, M. Vander Sande, T. De Nies, R. Verborgh, E. Mannens, and R. Van de Walle. Rdf mapping rules refinements according to data consumers feedback. In *2nd International World Wide Web Conference, Poster Track Proceedings*, 2014.

[8] http://www.isi.edu/integration/karma/
[9] http://openrefine.org/