Semantically Annotating CEUR-WS Workshop Proceedings with RML

Pieter Heyvaert[1], Anastasia Dimou[2]
Ruben Verborgh[2], Erik Mannens[2], and Rik Van de Walle[2]

[1] Ghent University - iMinds - Multimedia Lab,
pheyvaer.heyvaert@ugent.be
[2] Ghent University - iMinds - Multimedia Lab,
{firstname}.{surname}@ugent.be

Abstract. In this paper, we present our solution for the first task of the second Semantic Publishing Challenge. The task requires extracting and semantically annotating information regarding CEUR-WS workshops, their chairs and conference affiliations, as well as their papers and their authors, from a set of HTML-encoded workshop proceedings volumes. Our solution builds on last year's submission, while we address a number of shortcomings, assess the generated dataset for its quality and publish the queries as SPARQL query templates. This is accomplished using the RDF Mapping Language (RML) to define the mappings, the RMLProcessor to execute them, the RDFUnit to both validate the mapping documents and assess the generated dataset's quality, and The DataTank to publish the SPARQL query templates. This results in an overall improved quality of the generated dataset that is reflected in the query results.

1 Introduction

A lot of information is available on the Web through websites. However, this information is not always processable by Semantic Web enabled systems, because most HTML pages lack the required metadata. An example of such a website is CEUR-WS Workshop Proceedings (CEUR-WS)[3]. CEUR-WS is a publication service for proceedings of scientific workshops. It provides (i) a list of all the volumes indexed in a single Web page; and (ii) a detailed Web page for each volume. In need of assessing the scientific output quality, the Semantic Publishing Challenge (SPC14) was organized in 2014[4], followed by this year's edition[5] (SPC15).

In this paper, we propose a solution to solve the challenge's first task[6], which includes extracting information regarding workshops, their chairs and conference affiliations, as well as their papers and their authors, from a set of HTML-encoded tables of workshop proceedings volumes. In order to achieve this, we build on last

[3] http://ceur-ws.org/
[4] http://challenges.2014.eswc-conferences.org/index.php/SemPub/
[5] https://github.com/ceurws/lod/wiki/SemPub2015
[6] https://github.com/ceurws/lod/wiki/Task1

year's submission [1]. The solution uses the RDF Mapping language (RML)[7] [2, 3], which is a generic mapping language based on an extension over R2RML, the W3C standard for mapping relational databases into RDF. RML offers a uniform way of defining the mapping rules for data in heterogeneous formats.

We follow the same approach as last year. However, we (i) address a number of shortcomings, (ii) assess the generated dataset for its quality and (iii) publish the queries as SPARQL query templates. This is accomplished using RML (see Section 4) to define the mappings, the RMLProcessor to execute them, the RDFUnit to both validate the mapping documents and assess the generated dataset's quality (see Section 8.2), and The DataTank to publish the SPARQL query templates (see Section 8.3).

This paper that supports our submission to the SPC15 is structured as follows: we state the problem in Section 2, and give an overview of our approach in Section 3. In Section 4 we elaborate on the basis of the solution, namely RML. After defining how the data is modeled in Section 5, we elaborate on how the mapping is done in Section 6. We discuss how the queries of the task are evaluated in Section 7. In Section 8 we explain the used tools: RMLProcessor (Section 8.1), the RDFUnit (Section 8.2) and The DataTank (Section 8.3). Finally, in Section 9, we discuss our solution and its results, after which we form our conclusions.

2 Problem Statement

The conclusions of the Semantic Publishing Challenge 2014 [4] show that the submitted solutions provided satisfying results. However, they also highlight that there is still room for improvement. With the Semantic Publishing Challenge 2015, the organizers continue pursuing the objective of assessing the quality of scientific output and of evolving the dataset bootstrapped in 2014 to take also into account the wider ecosystem of publications. The challenge consists of the following three tasks:

Task 1 Extraction and assessment of workshop proceedings information,
Task 2 Extracting contextual information from the papers text in PDF, and
Task 3 Interlinking

In this paper we explain how we tackle the first task of the challenge. The participants are asked to extract information from a set of HTML tables published as Web pages in the CEUR-WS workshop proceedings. The information is obtained from the HTML pages' content which is semantically annotated and represented using the RDF data model. The extracted information is expected to answer queries about the quality of these workshops, for instance by measuring growth, longevity, and so on. The task is an extension of the SPC14's first task. The most challenging quality indicators from last year's challenge are reused. However, a number of them are defined more precisely, and new indicators are added. This results in the following three subtasks:

[7] http://rml.io

SubTask 1.1 Extract information from the HTML input pages;
SubTask 1.2 Annotate the information with appropriate ontologies and vocabularies; and
Subtask 1.3 Publish the semantically enriched representation with the RDF data model.

3 Overview of Our Approach

Our approach includes: (i) the generation of the RDF dataset and (ii) the evaluation of the SPARQL queries. The first is achieved with the following workflow:

1. define the mapping documents, using RML;
2. assess the mapping documents, using the RDFUnit;
3. generate the dataset, by executing the mappings, using the RMLProcessor;
4. assess the quality of the dataset, using theRDFUnit, and
5. publish the dataset, using The DataTank.

After the generation of the RDF dataset, the queries of the task are evaluated (see Section 7). In order to achieve this, the following are considered:

1. define the queries, using SPARQL templates, using The DataTank,
2. instantiate and execute the SPARQL queries, and
3. provide the results.

The components and output of our solution and where they can be found are summarized in Table 1.

Table 1. Submission's Output

Output	**Location**
RML mapping documents	`http://rml.io/data/SPC2015/mappings`
RDF dataset	`http://rml.io/data/SPC2015/dataset.ttl`
SPARQL templates	`http://rml.io/data/SPC2015/sparql_templates`
Query results	`http://rml.io/data/SPC2015/query_results`
List of queries on The DataTank	`http://ewi.mmlab.be/spc`

4 RML

RDF Mapping Language (RML) [2, 3] is a generic language defined to express customized mapping rules from data in heterogeneous formats to the RDF data model. RML is defined as a superset of the W3C-standardized mapping language R2RML [5], 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, making RML highly extensible towards new source formats.

4.1 Structure of an RML Mapping Document

In RML, the mapping to the RDF data model is based on one or more Triples Maps that define how RDF triples should be generated. A Triples Map consists of three main parts: (i) the Logical Source (`rml:LogicalSource`), (ii) the Subject Map, and (iii) 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 RDF triples 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 define 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).

4.2 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 stated at the Logical Source. This makes RML highly extensible towards new source formats.

At the current version of RML, the `ql:CSV`, `ql:XPath`, `ql:JSONPath` and `ql:CSS3` Reference Formulations are predefined (where *ql* is the prefix for `http://semweb.mmlab.be/ns/ql#`). For the task we use the `ql:CSS3` Reference Formulation to access the elements within the document. CSS3[8] 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[9]. CSS3 selectors can be used to refer to data in HTML documents. However, they can also be used for XML documents.

[8] `http://www.w3.org/TR/selectors/`
[9] `http://jquery.com`

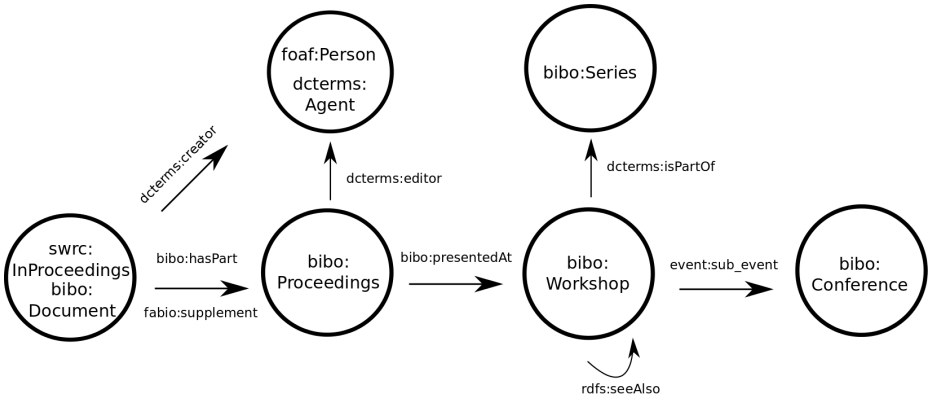

Fig. 1. An overview of the interaction between the classes and properties used to model the workshops proceedings information.

5 Data Modeling

In order to model the workshop proceedings information, we use the following ontologies:

- The Bibliographic Ontology[10] (with prefix *bibo*),
- DCMI Metadata Terms[11] (with prefix *dcterms*),
- Friend of a Friend[12] (with prefix *foaf*),
- RDF Schema[13] (with prefix *rdfs*),
- FRBR-aligned Bibliographic Ontology[14] (with prefix *fabio*)
- The Event Ontology[15] (with prefix *event*)
- Semantic Web for Research Communities[16] (with prefix *swrc*)

The classes used to determine the type of the entities are denoted in Table 2.

The properties used to annotate the entities and determine the relationships among them are denoted in Table 3. The properties listed here are not exhaustive, and for a complete overview of the used properties we refer to the mapping documents[17]. An overview of the entities and the relationships between the entities and the properties that determine them is shown in Figure 1. Overall, the modelling of the data is driven by the queries that need to be answered as part of the challenge.

We extracted information related to workshop (`bibo:Workshop`) entities from the index page. Furthermore, we extracted information that models the relationship among different workshops (`rdfs:seeAlso`) of the same series, that denotes which proceedings are presented at a workshop (`bibo:presentedAt`) and states the conference that the

[10] `http://purl.org/ontology/bibo/`
[11] `http://purl.org/dc/terms/`
[12] `http://xmlns.com/foaf/0.1/`
[13] `http://www.w3.org/2000/01/rdf-schema#`
[14] `http://purl.org/spar/fabio/`
[15] `http://purl.org/NET/c4dm/event.owl`
[16] `http://swrc.ontoware.org/ontology#`
[17] `http://rml.io/data/spc2015/mappings`

Table 2. Classes

Class	Entity
bibo:Workshop	workshop
bibo:Series	workshop series
bibo:Proceedings	proceedings of a workshop
bibo:Conference	event where a workshop took place
foaf:Person dcterms:Agent	editor of a proceedings and author of a paper
swrc:InProceedings bibo:Document	paper

Table 3. Properties

Property	Relationship
dcterms:creator	person who is author of a paper
dcterms:hasPart	proceedings that a paper belong to
fabio:supplement	proceedings that a supplemental document (e.g., invited paper) belong to
dcterms:editor	person who is editor of proceedings
bibo:presentedAt	workshops that the papers, hence, the proceedings, are presented
dcterms:isPartOf	workshop series that a workshop is part of
rdfs:seeAlso	workshop that is related to this workshop
event:sub_event	event that the workshop is a subevent of

workshop was co-located with (`dcterms:isPartOf`). To determine the workshops we iterated over the volumes, because, except for the joint volumes, all of them represent a separate workshop. Finally, the workshops related to the current one are added by following the 'see also' links in its description.

Each volume page represents a proceedings entity (`bibo:Proceedings`). This HTML page contains information about the papers (`swrc:InProceedings`, `bibo:Document`), which are connected to the proceedings (`bibo:hasPart`). We make a distinction between non-invited and invited papers (using `fabio:supplement` instead of `bibo:hasPart`). The authors (`foaf:Person`, `dcterms:Agent`) are defined (using `dcterms:creator`) of each paper, as well as the editors (`foaf:Person`, `dcterms:Agent`) of the proceedings (`dcterms:editor`). Finally, from the workshop's name its series (`bibo:Series`) is determined and the workshop's co-located event (`bibo:Conference`) is determined (using `event:sub_event`). The extraction of additional information (location, date, edition), annotated with datatype properties, is defined in the mapping documents. Due to the repetitive nature of the corresponding definitions, we refer to the mapping documents for more details.

6 Mapping CEUR-WS from HTML to RDF

The task refers to two types of HTML pages that serve as input. On the one hand it is the index page listing all the volumes, namely `http://ceur-ws.org`. On the other hand, for each volume there is an HTML page that contains more detailed information, e.g., `http://ceur-ws.org/Vol-1165/`.

6.1 Defining the Mappings

Excerpts of a mapping document for one of the volumes are indicatively presented. First, the input source (Listing 1.1, line 5) that is used by this Triples Map (Listing 1.1, line 4) is stated, together with the Reference Formulation, in this case the CSS3 selectors (Listing 1.1, line 7), that states how we refer to the input and the iterator (Listing 1.1, line 6) over which the iteration occurs, as in Listing 1.1:

```
1  @prefix rml: <http://semweb.mmlab.be/ns/rml#>.
2
3  <#VolumeMapping>
4    rml:logicalSource [
5      rml:source <http://ceur-ws.org/Vol-1128> ;
6      rml:iterator "body";
7      rml:referenceFormulation ql:CSS3 ].
```

Listing 1.1. Defining the source of a mapping for a volume page

To define how the subject of all RDF triples will be generated using this Triples Map (Listing 1.2, line 4), we define a Subject Map (Listing 1.2, line 5). A unique URI will be generated for each volume with the volume number that is present on each page. This number is addressable by the CSS3 expression `span.CEURVOLNR` (Listing 1.2, line 6). The class of the workshop is set to `swrc:Proceedings` (Listing 1.2, line 7). The definition of a complete Subject Map can be found in Listing 1.2:

```
1  @prefix rr: <http://www.w3.org/ns/r2rml#>.
2  @prefix swrc: <http://swrc.ontoware.org/ontology#> .
3
4  <#VolumeMapping>
5    rr:subjectMap [
6      rr:template "http://ceur-ws.org/{span.CEURVOLNR}/";
7      rr:class swrc:Proceedings ].
```

Listing 1.2. Defining the subject of mapping for a volume page

For each RDF triple of the volume we need to define a Predicate Object Map (Listing 1.3, line 7). In our example (see Listing 1.3), we add the predicate for the label (`rdfs:label`) to the volume (Listing 1.3, line 6). The value of the object is specified as the content of the link (`<a>`) inside the `<span>` with the class `CEURVOLTITLE`, which results in the CSS3 selector `span.CEURVOLTITLE a` (Listing 1.3, line 8). The definition of a complete Subject Map is indicatively presented at Listing 1.3:

```
1  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
2  @prefix rr: <http://www.w3.org/ns/r2rml#>.
3
4  <#VolumeMapping>
5    rr:predicateObjectMap [
6      rr:predicate rdfs:label;
7      rr:objectMap [
8        rml:reference "span.CEURVOLTITLE a" ] ].
```

Listing 1.3. Defining the Objects (as literals) for the subject for a volume page

For the object's generation, RML is not limited to literals, as in the previous example. A reference to another Triples Map (Listing 1.4, line 8), instead of an rml:reference, is used to generate resources instead of literal values. In Listing 1.4, we state that all subjects of `<#EditorMapping>` are editors (`bibo:editor`) of the volume:

```
1  @prefix bibo: <http://purl.org/ontology/bibo/>.
2  @prefix rr: <http://www.w3.org/ns/r2rml#>.
3
4  <#VolumeMapping>
5    rr:predicateObjectMap [
6      rr:predicate bibo:editor;
7      rr:objectMap [
8        rr:parentTriplesMap <#EditorMapping> ] ].
```

Listing 1.4. Defining the objects (as resources) for the editors of a volume

6.2 Executing the Mappings

Executing an RML mapping requires a mapping document that summarizes all Triples Maps and points to an input data source. The mapping document is executed by an RML processor and the corresponding RDF output is generated. Each Triples Map is processed and the defined Subject Map and Predicate Object Maps are applied to the input data. For each reference to the input HTML, the CSS3 extractor returns an extract of the data and the corresponding triples are generated. The resulting RDF can be exporting in a user-specified serialization format. This solves subtask 1.3.

Data cleansing is out of RML's scope. However, the values extracted from the input is not always exactly as desired to be represented in RDF and the situation aggravates when mapping e.g. live HTML documents on-the-fly, where neither pre-processing is possible nor being as selective as desired purely based on CSS3 expressions to retrieve extracts from HTML pages. To this end, we defined and used `rml:process`, `rml:replace` and `rml:split` to further process the values returned from the input source as defined within a mapping rule. To be more precise, `rml:process` and `rml:replace` were used to define regular expressions whenever it is required to be more selective over the returned value and replaced by a part of the value or another value. For instance, a reference to `h3 span.CEURLOCTIME` returns `Montpellier, France, May 26, 2013` and since there is no further HTML annotation, we cannot be more selective over the returned value. In these cases `rml:process` is used to define a regular expression, e.g. `([a-zA-Z]*), [a-zA-Z]*, [a-zA-Z]* [0-9]*, [0-9]*`, and `rml:replace` is used to define the part of the value that is used for a certain mapping rule, e.g., `$1`, for the aforementioned case to map the city `Montpellier`. Furthermore, `rml:split` allows to split the value based on a delimiter and to map each part separately. The possibility to chain them enables even more fine-grained selections. These adjustments contribute in solving subtask 1.2.

Challenge-Specific Adjustments In order to cope with a number of non-trivial structures of the challenge-specific HTML input sources, the default CSS3 selectors are not expressive enough. To this extent, we added the CSS3 function `:until(x)` to CSSelly[18], a Java implementation of the W3C CSS3 specification, used by the RMLProcessor. This function matches the first x found element in the HTML document.

The structure of the index page does not allow to use the default CSS3 selectors to extract the required information. However, implementing a custom function is not possible in this case, due to the extensibility limitations of CSSelly. To this extent, we reformatted[19] the index page to make it processable using the available selectors.

[18] `http://jodd.org/doc/csselly/`

[19] This tool is available at `http://rml.io/data/spc2015/reformat_tool`.

Last, a number of HTML pages contain invalid HTML syntax. To cope with this, we used JTidy[20] to produce valid versions of the HTML pages[21]. These adjustments allow to solve subtask 1.1.

7 Query Evaluation

The queries for Task 1 of the challenge can be found at `https://github.com/ceurws/lod/wiki/QueriesTask1`. Based on the description of each query, we created the corresponding SPARQL queries based on our data model (Section 5). Because of the queries templated nature, we defined our queries as SPARQL templates[22] and published them using The DataTank (Section 8.3), allowing easy access to the queries for different values. For example, the SPARQL template for the query 1.1 can be found in Listing 1.5. It is the same as the original query with exception of line 9, where `${workshop}` is added. If we want to execute the query with the value `Vol-1085` for the variable `workshop`, we consider the following URI `http://rml.io/data/spc2015/tdt/queries/q01.json?workshop=Vol-1085`. This returns the results of the query in JSON format.

```
1  PREFIX bibo: <http://purl.org/ontology/bibo/>
2  PREFIX swrc: <http://swrc.ontoware.org/ontology#>
3  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4
5  SELECT DISTINCT ?W ?name
6  WHERE {
7          ?W a swrc:Proceedings ; bibo:editor ?editor .
8          ?editor rdfs:label ?name.
9          FILTER (?W = <http://ceur-ws.org/${workshop}/>  ) }
```

Listing 1.5. Sparql Template of Query 1.1

8 Tools

The execution of our publishing workflow is accomplished based on two tools: the RMLProcessor that is used to execute the mapping definitions and generate the RDF dataset and RDFUnit that is used to validate and improve the quality of both the defined schema and the generated dataset. Besides the publishing workflow, we used another tool, The DataTank to publish the SPARQL queries.

8.1 RML Processor

Our RMLProcessor[23], implemented in Java on top of db2triples[24], was used to perform the mappings. The RMLProcessor follows the mapping-driven processing approach, namely it reads the mapping definitions as defined with RML, and executes the mapping rules to generate the corresponding RDF dataset. The RMLProcessor has a modular architecture where the extraction and mapping modules are executed independently of each other. When the RML mappings are processed, the mapping module deals with the mappings' execution as defined in the mapping document in RML syntax, while

[20] `http://jtidy.sourceforge.net/`
[21] The valid HTML pages are available at `http://rml.io/data/spc2015/valid_html`.
[22] `http://rml.io/data/spc2015/sparql_templates`
[23] `https://github.com/mmlab/RMLProcessor`
[24] `https://github.com/antidot/db2triples/`

the extraction module deals with the target languages expressions, in our case CSS3 expressions. To be more precise, the RMLProcessor uses CSSelly, a Java implementation of the W3C CSS3 specification.

8.2 RDFUnit

RDFUnit [6] is an RDF validation framework inspired by test-driven software development. In RDFUnit, every vocabulary, ontology, dataset or application can be accompanied by a set of data quality Test Cases (TCs) that ensure a basic level of quality. Assigning TCs in ontologies results in tests that can be reused by datasets sharing the same schema. All TCs are executed as SPARQL queries using a pattern-based transformation approach. In our workflow, we use RDFUnit to assure that (i) the mapping documents validate against the RML ontology, (ii) the schema, as a combination of several ontologies and vocabularies, is valid and (iii) the generated dataset does not contain violations in respect to the schema used.

8.3 The DataTank

The DataTank[25] is a RESTful data management system written in PHP and maintained by OKFN Belgium[26]. It enables publishing several data formats into Web readable formats. The source data can be stored in text based files, such as CSV, XML and JSON, or in binary structures, such as SHP files and relational databases. The DataTank reads the data out of these files and/or structures and publishes them on the Web using a URI as an identifier. It can provide the data in any format depending on the users needs, independently of the original format. Next to publishing data, The DataTank allows to publish (templated) SPARQL queries. SPARQL templates make it possible to define a variable's value at runtime (by the user). As a result, those queries have improved reusability and their scope fits well in the challenge's needs.

9 Discussion and Conclusion

It is beneficial that CSS3 selectors become part of a formalization 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 reusing same mappings over different files, the effort they put is even less. For the challenge, same mapping documents and/or definitions were re-used for different HTML input sources.

It is reasonable to consider CSS3 selectors to extract content from HTML pages because nowadays most websites use templates, formed with CSS3 selectors. Thus the content of their Web pages is structured in a similar way, which is the same point of reference as the one used by RML. This allows us to use RML mapping documents as a 'translation layer' over the published content of HTML pages.

Furthermore, as the mappings are partitioned in independent Triples Maps, data publishers can select the Triples Maps they want to execute at any time. For instance, in the case of the challenge, if violations were identified using the RDFUnit because

[25] http://thedatatank.com/

[26] http://www.openknowledge.be/

of incorrect mappings, we can isolate the Triples Map that generated those triples, correct the relevant mapping definitions and re-execute them, without affecting the rest mapping definitions or the overall dataset. This becomes even easier considering that the mappings in RML are defined as triples themselves and, thus, the triples' provenance can be tracked and used to identify the mappings and data that cause the erroneous RDF result.

Beyond re-using the same mapping documents, RML allows to 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 results of the extraction of the information from the PDF's of the papers published at the workshops, aligning with the results of Task 2. This results in enriching dataset when the two original datasets are combined.

Compared to last year's submission, we made the following improvements: (i) more information was extracted from the index page, while we keep the volume mapping documents simpler; (ii) the information extraction was focused on answering the challenge's queries; and (iii) series and workshops were modeled as separate entities, adding more semantic meaning to the resulting dataset; (iv) we use single mapping documents for multiple Web pages of the CEUR-WS HTML input sources. These improvements occur thanks to the updated syntax and the more stable release of RMLProcessor, leading to a higher number of supported queries.

Acknowledgements The described research activities were funded by Ghent University, iMinds, 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] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Laurens De Vocht, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. Extraction and Semantic Annotation of Workshop Proceedings in HTML using RML. In *Semantic Web Evaluation Challenge*, pages 114–119. Springer, 2014.

[2] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *Workshop on Linked Data on the Web*, 2014.

[3] Anastasia Dimou, Miel Vander Sande, Jason Slepicka, Pedro Szekely, Erik Mannens, Craig Knoblock, and Rik Van de Walle. Mapping Hierarchical Sources into RDF using the RML Mapping Language. In *Proceedings of the 8th IEEE International Conference on Semantic Computing*, 2014.

[4] Christoph Lange and Angelo Di Iorio. Semantic publishing challenge assessing the quality of scientific output. In Valentina Presutti, Milan Stankovic, Erik Cambria, Ivn Cantador, Angelo Di Iorio, Tommaso Di Noia, Christoph Lange, Diego Reforgiato Recupero, and Anna Tordai, editors, *Semantic Web Evaluation Challenge*, volume 475 of *Communications in Computer and Information Science*, pages 61–76. Springer International Publishing, 2014. ISBN 978-3-319-12023-2.

[5] Souripriya Das, Seema Sundara, and Richard Cyganiak. R2RML: RDB to RDF Mapping Language. Working group recommendation, W3C, September 2012. URL `http://www.w3.org/TR/r2rml/`.

[6] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. Test-driven evaluation of Linked Data Quality. In *Proceedings of the World Wide Web Conference*, pages 747–758, 2014.