CHAPTER 8

CMD2RDF: Building a Bridge from CLARIN to Linked Open Data

Menzo Windhouwer[a,1], Eko Indarto[b] and Daan Broeder[a]
[a]Meertens Institute, [b]Data Archiving and Networked Services (DANS)

ABSTRACT

Metadata can be represented in many different ways. CLARIN's Component Metadata Infrastructure (CMDI) uses the eXtensible Markup Language (XML) as the representation format for metadata records. However, the Resource Description Format (RDF) as used by Linked Open Data (LOD) is gaining more popularity. RDF has interesting potential for queries that involve both metadata about and the content of linguistic resources. This chapter describes the implementation of a mapping for records in CMDI from XML to RDF and experiments to assess the potential of this representation.

8.1 Introduction

Metadata has always been a key issue for libraries and archives and thus has a long history (M-Files, 2016). Throughout the ages the physical form and, more recently, the digital representation of metadata has changed, i.e., adapted to the standard current at that time. When the CLARIN preparatory phase started in 2007 the eXtensible Markup Language (XML; Bray et al., 2008) was the current standard. CLARIN's metadata standard as implemented in the Component Metadata Infrastructure (CMDI; Broeder et al., 2012; CLARIN, 2016a) is thus also based on XML as the representation format for metadata. However, the Resource Description Format (RDF; Cyganiak, Wood and Lanthaler, 2014) as used, for example, by the Linguistic Linked Open Data (LLOD) cloud (Chiarcos et al., 2012; LIDER project, 2016) is gaining more popularity. RDF provides an

[1] Corresponding author: menzo.windhouwer@meertens.knaw.nl

How to cite this book chapter:
Windhouwer, M, Indarto, E and Broeder, D. 2017. CMD2RDF: Building a Bridge from CLARIN to Linked Open Data. In: Odijk, J and van Hessen, A. (eds.) *CLARIN in the Low Countries*, Pp. 95–103. London: Ubiquity Press. DOI: https://doi.org/10.5334/bbi.8.

interesting potential for queries that involve both metadata about and the content of linguistic resources, as both metadata and content can be collected and queried in a set of connected graphs. In the CMD2RDF project CLARIN-NL (2016) CLARIN-NL sponsored the actual implementation of the mapping from Component Metadata (CMD) to RDF, which has been proposed by Durco and Windhouwer (2014a), and the services to provide access to the resulting RDF. This enables the CLARIN community to experiment with RDF representations of the CMD records, and to get a sense of its potential and the opportunities for cross fertilisation with other Linked Data resources like those found in the LLOD cloud

The results of this project are described in the main part of this chapter. The first two sections provide a short summary of both CMDI and the Linked Data paradigm, and the chapter ends with the current status of CMD2RDF and future plans for it.

8.2 The Component Metadata Infrastructure

The basic building blocks of CMDI are, not surprisingly, components. A component focuses on a specific aspect of a (linguistic) resource and groups together metadata elements, which can be used to capture information, and other components. For example, an *address* component contains the elements *street*, *city* and *country.* This component could be reused by a *contact person* or an *organisation* component. The infrastructure provides a Component Registry for metadata modellers to share and reuse components. The registry is accompanied by an editor, which allows adapting components to specific needs or creating completely new ones. A modeller in the end creates metadata profiles, i.e., a collection of metadata components, targeted at a specific resource type, e.g., a historic text or an audio recording of an endangered language. A CMD profile is a tree-based structure where the nodes are components, from which one is the root of the tree, and the leaves are elements. This tree can be very naturally mapped to XML and thus an XML Schema (XSD; Gao, Sperberg-McQueen and Thompson, 2012) can be used to validate whether a CMD record is compliant with a specific profile. In CLARIN various tools, e.g. online and offline editors, have been developed to create and maintain valid CMD records (also known as metadata descriptions). This core of CMDI, the Component Metadata model, is visualised in Figure 8.1 and has been standardised by ISO *Technical Committee 37* (ISO 24622-1, 2015)

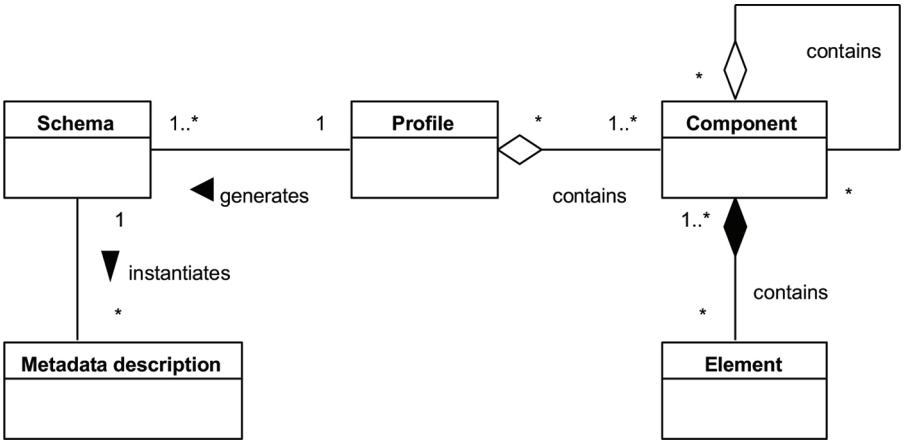

Figure 8.1: Component Metadata model (ISO 24622-1, 2015).

CLARIN centres offer the CMD records they create for harvesting via the Open Archives Initiative's Protocol for Metadata Harvesting (OAI-PMH; Lagoze and Van de Sompel, 2015). Central CLARIN services, like the Virtual Language Observatory (VLO; CLARIN, 2016b), provide access to the full set of harvested CMD records.

8.3 Linked Open Data

Linked Data, open or closed, has become increasingly popular. In this paradigm graphs are constructed out of triples consisting of a subject, a predicate and an object. The object of a triple can be the subject of another triple thus building the graph. All parts of the triple can be identified (nodes, i.e., subjects or objects) or typed (nodes or edges, i.e., predicates) with an Internationalized Resource Identifier (IRI; Dürst and Suignard, 2005), most commonly a Uniform Resource Location (URL; Berners-Lee, Masinter and McCahill, 1994). A coherent vocabulary of types is commonly described in an RDF Schema (RDFS; Brickley and Guha, 2014) or extensions thereof. Many RDF vocabularies (Open Knowledge Foundation, 2016) exist and some are frequently reused. Graphs are linked with each other when they share an IRI. In this way large graphs like the Linked Open Data (LOD) cloud Cyganiak and Jentzsch (2016) and the Linguistic Linked Open Data (LLOD) cloud (LIDER project, 2016) can be identified.

Access to (parts of) these graphs is mostly provided in two ways: 1) as downloads in one or more of the various RDF serialisations, and/or 2) via SPARQL (W3C SPARQL Working Group, 2013) query endpoints. In the latter case the graphs are in general stored in a triple store, i.e., a system for managing (large) sets of triples equivalent to Relational DataBase Management Systems (RDBMS) for structured data.

8.4 The CMD2RDF Bridge

The aim of the CMD2RDF project has been to bring all of the CLARIN CMD record collection to the Linked Data cloud. For this the XML-based records have to be transformed into RDF without loss of information (note that this goal is different from the approach taken by an aggregator like LingHub (McCrae et al. 2015), where only a subset of the information, i.e. in the case of LingHub the set already mapped to Dublin Core (DC; Dublin Core Metadata Initiative 2016) by the OAI-PMH provider, is transformed to RDF). The flexibility of CMDI also means that in such a generic transformation a fixed metadata RDF Schema, like the Data Catalog Vocabulary (DCAT; Maali and Erickson, 2014), is not directly applicable as it would require hand-crafted and maintained mappings to the fixed schema for every CMD profile encountered. But as shown below more generic RDF vocabularies do play a role in transformation. These graphs should also be accessible, either as a download or via a SPARQL endpoint. The next subsections describe the approaches taken to tackle these issues.

8.4.1 The Component Model and RDF

A CMD record is an instance of a CMD profile, which in its turn is an instance of the CMD model. Next to the profile-specific part each record also uses a generic envelope, e.g. to provide information on the resources involved. For all these levels and parts an RDF equivalent has to be created The following description is short, i.e. highlights some issues, the design choices made to resolve them and consists mainly of examples, but Durco and Windhouwer (2014a) gives a full description of the mapping of all these levels and parts.

In the CMD model the main building block, the CMD component naturally corresponds to an RDFS class. A CMD profile can be seen as a specialisation of component, so it is a subclass of the RDFS class for component. It seems natural to map a CMD element to an RDF property. However, a CMD element is more complex than an RDF property, i.e., it can carry additional information in the form of attributes. To be able to retain this information in the mapping a CMD element also has to be mapped to an RDFS class. In RDF, as opposed to XML, the nesting of CMD components or elements in a CMD component needs a predicate. For this the very generic *contains* predicate is introduced. To retain consistency attributes are modelled in a similar way as elements. This results in the following RDF Schema:

```
cmdm:Component a rdfs:Class .
cmdm:Profile rdfs:subClassOf cmdm:Component .
cmdm:Element a rdfs:Class .
cmdm:Attribute a rdfs:Class .

cmdm:contains
    a rdf:Property ;
    rdfs:domain cmdm:Component ;
    rdfs:range cmdm:Component, cmdm:Element .

cmdm:containsAttribute
    a rdf:Property ;
    rdfs:domain cmdm:Component, cmdm:Element ;
    rdfs:range cmdm:Attribute .

cmdm:hasElementValue
    a rdf:Property ;
    rdfs:domain cmdm:Element, rdfs:Literal .

cmdm:hasAttributeValue
    a rdf:Property ;
    rdfs:domain cmdm:Attribute ;
    rdfs:range rdfs:Literal .
```

Based on this mapping of the CMD model a specific component can be transformed into RDF. For example:

```
cmd1:collection
      a cmdm:       Profile ;
      rdfs:label    "collection" .
cmd2:Actor
      a cmdm:       Component ;
      rdfs:label    "Actor" .
cmd2:Actor_Languages
      a             cmdm:Component ;
      rdfs:label    "Languages" .
cmd2:Actor_Languages_Language
      a             cmdm:Element ;
      rdfs:label    "Language" .
```

where the `cmd1:` and `cmd2:` prefixes are bound to component-specific IRIs, i.e., the URL to the component specification in the CMDI Component Registry.

A complicating matter is that although a component or element has a unique name among its siblings, within a single component specification a name can very well be ambiguous – so context has to be taken into account. This is done by adding the context to the IRI of a component or

element; e.g., `cmd2:Actor_Languages_Language` represents a *Language* element nested in a *Languages* component which itself is nested in a reusable *Actor* component.[2]

Now that a CMD profile can be transformed into an RDF Schema an actual CMD record can also be transformed. The core of such a record is formed by its instantiation of the component hierarchy allowed by the profile:

```
_:collection1   a           cmd1:collection .
_:actor1        a           cmd2:Actor .
_:languages1    a           cmd2:Actor_Languages .
_:language1     a           cmd2:Actor_Languages_Language .

_:collection1   cmdm:contains           _:actor1 .
_:actor1        cmdm:contains           _:languages1 .
_:languages1    cmdm:contains           _:language1 .
_:language1     cmdm:hasElementValue    "nld" .
```

In this example the hierarchy is instantiated using RDF blank nodes, but the IRI of a record extended with a local unique identifier can also be used.

In a CMD record the profile-specific payload is placed inside a generic CMD envelope, which contains information about the resources involved and metadata about the records themselves, e.g. who has created them and when. This part is also mapped to RDF. And as it is more generic it was possible to reuse existing RDF vocabularies: Dublin Core for the metadata, Open Annotation (OA; W3C Web Annotation Working Group, 2016) for the relation between the profile-specific part and the resources, and the Open Archives Initiative's Object Reuse and Exchange vocabulary (ORE; Open Archives Initiative, 2016) for the relationships of the record with other CMD records.

8.4.2 From Harvesting CMD to Providing RDF

Using the RDF mapping described above any CMD record can be transformed. However, to be of actual use the continuously evolving CLARIN-wide collection of CMD records would have to become available in the Linked Data cloud. To achieve this goal the system architecture depicted in Figure 8.2 was implemented in the CMD2RDF project.

CMD records provided by the CLARIN centres are regularly harvested by the CLARIN OAI-PMH harvester. As the harvester currently does not support incremental harvests, and since even if it did all centres would still not necessarily support them, the CMD2RDF conversion pipeline determines which records are new or updated and transforms those into RDF. These RDF records and the RDFS of the components and profiles involved are stored in the Virtuoso triple store (OpenLink Software, 2016). Virtuoso supports a SPARQL endpoint and RESTful access to the RDF graphs, which each correspond to a CMD record. CMD2RDF does put a proxy in front of those to be able to (potentially) control the access, e.g. to prevent too heavy SPARQL queries. The resulting service is available at:

catalog.clarin.eu/ds/cmd2rdf

Another important aspect of the CMD2RDF conversion pipeline is the ability to also enrich the CMD or RDF representations. This makes it possible to introduce links to other datasets, i.e., determine the place of a CMD record in the Linked Open Data cloud and especially in the Linguistic Linked Open Data cloud.

[2] This is the only place where the implementation differs from the mapping described in Durco and Windhouwer (2014a): as the dot ('.') has a special meaning in many RDF representations and also in SPARQL its use as a separator for the context turned out to be problematic and was replaced by an underscore ('_').

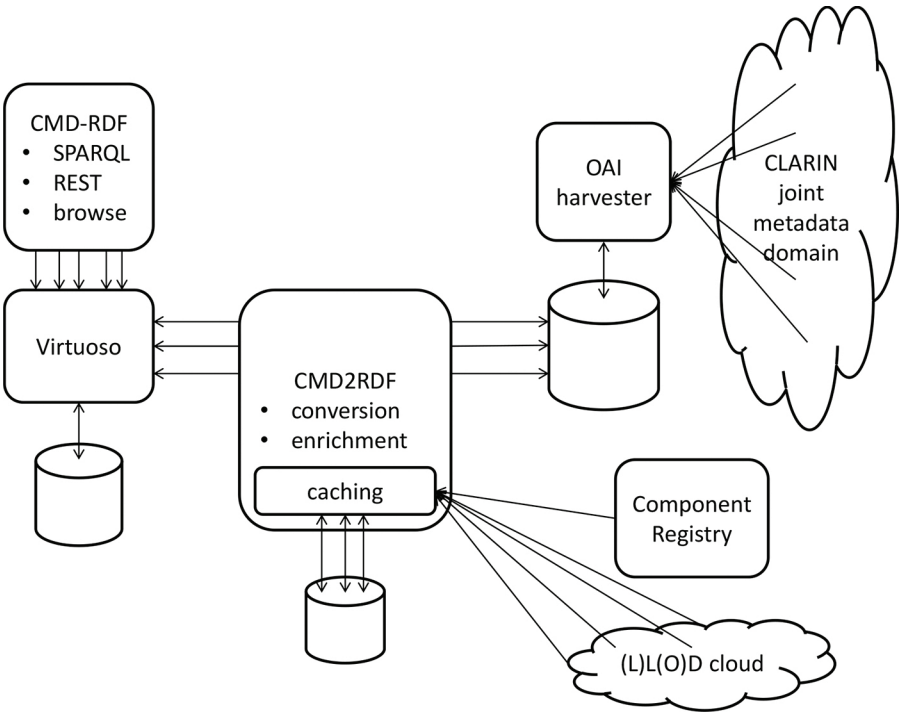

Figure 8.2: The CMD2RDF system architecture (catalog.clarin.eu/ds/cmd2rdf).

8.5 CMD2RDF and LLOD

In the CMD2RDF system architecture CMD records can be enriched with links to other LLOD datasets. The main linking pins for linguistic datasets are of course languages. The most prominent set of language codes is ISO 639:3 (Summer Institute for Linguistics, 2016), which is represented by DBpedia (2016) IRIs in the LOD cloud. Due to the heterogeneous nature of CMDI these codes can appear anywhere in a CMD record. However, due to the semantic network (Durco and Windhouwer, 2014b) that overlays the CLARIN collection of CMD record these places can be identified. Currently CMD2RDF uses the approach used for the VLO facet mapping (Van Uytvanck, Stehouwer and Lampen, 2012) and includes the resulting facets explicitly. To retain the original value next to the IRI identified by the enrichment process the `cmdm:hasElementEntity` predicate (which gets subclassed by specific enrichments like the VLO facets) was introduced:

```
<hdl:123/456>
    vlo:hasFacetISO6393ElementValue     "nld" ;
    vlo:hasFacetISO6393ElementEntity
        <http://dbpedia.org/resource/ISO_639:nld> .
```

As a showcase the WALS dataset (Dryer and Haspelmath, 2013) was also loaded into Virtuoso. Now SPARQL queries can be issued that involve both CMD records and linguistic content, i.e., WALS. The following query is an example of this:

```
SELECT DISTINCT ?resource ?mimetype ?language ?value
WHERE {
  ?feature dcterms:references wals:9A .
  ?feature dcterms:hasPart/rdfs:label ?value .
  ?feature ^dcterms:isReferencedBy/owl:sameAs ?language
```

```
  GRAPH ?g {
    ?cmd vlo:hasFacetISO6393ElementEntity ?language .
    ?cmd oa:hasTarget ?resource .
    ?resource cmdm:hasMimeType ?mimetype .
  }
}
```

This query returns the locations (`?resource`) of multimedia (`?mimetype`) resources for languages (`?language` – from the RDF graph `?g`, which represents the CMD record `?cmd`) where the WALS contains information (`?value`) on a typological feature (`?feature`), i.e., the distribution of the sound *η* (the velar nasal, which is WALS feature 9A). The example SPARQL queries at catalog.clarin.eu/ds/cmd2rdf include this query so its current result can be inspected there.

Similar queries that cross (multiple times) the boundaries between metadata and content can easily be envisioned. For example, the new Lexicon Model for Ontologies (Ontolex; Cimiano, McCrae and Buitelaar, 2016), which is an RDF-based model, would enable one to query for the word for a concept, e.g., *peace* or *love* in a specific language, and via CMD2RDF time segments in annotated media could be found where this word in uttered. Several lexica are available in Ontolex or its RDF-based predecessors, but the use of RDF for time-based annotations is not so common.

The example query also shows that still quite intimate knowledge of the usage of specific RDF vocabularies by the involved datasets is needed, but this is to be expected for structured queries where one has to know the structure, as opposed to full text or facetted search Writing a SPARQL query like this is a task for a technically savvy and adventurous user, so for the average user easier interfaces will need to be provided. The CMD2RDF service does include a general RDF browser, which allows some basic interaction with the SPARQL endpoint, but for more domain-specific interaction expert user interfaces with more built-in knowledge of the used vocabularies are needed.

8.6 Current Status and Future Plans

For a while the CMD2RDF service has been hosted by the Max Planck Institute for Psycholinguistics, but due to strategic decisions by this CLARIN centre the service had to be moved, and, as a medium-term solution, is now hosted by the Meertens Institute However, the generic CLARIN URL redirect at catalog.clarin.eu/ds/cmd2rdf will take any user to the current host.

In the new Dutch CLARIAH (2016) project, which covers both linguistics and the broader Digital Humanities, there is an agreement to use RDF as a lingua-franca and to merge information obtained from different sources. The CLARIAH approach for the linguistics work package will be based on the CMD Infrastructure for compatibility with CLARIN; however, it will also offer Linked Data via the CMD2RDF service for use by others.

To also enable the discovery and use of interesting resources created within non-linguistic work packages in CLARIAH, an inverse procedure, i.e., RDF2CMD, is required, which if sufficiently scalable, will also make the Linked Data for Language Resources (LR) outside CLARIAH available for CLARIN.

With respect to the procedure to facilitate this transformation of RDF encoded LR metadata the plan is to investigate a number of different strategies. All strategies will start with a PID (Persistent IDentifier) or URI (Uniform Resource Identifier) of a LR and then search from a suitable source, e.g. a SPARQL endpoint, RDF data set, for statements related to this resource. The collected RDF statements are aggregated and processed. The RDF2CMD mapping can then use, for example, the following strategies:

- *Comparison strategy:* the collected RDF is compared to a number of RDF templates that were derived from a set of records, which instantiate recommended CMD profiles. A suitable

proximity measure will then select the closest template after which the original CMD profile can be instantiated with the correct values.
- *Building strategy:* the collected RDF is inspected and every triple considered for implying a component or element in a dedicated CMD profile. The generated profile may be unique and can be 'shaved' of linguistically uninteresting non-linguistic adornments.

Minimal functionality should be supporting roundtrip conversion from a CMD record to RDF and back to CMD without loss of information, but the 'perfect' translation from Dublin Core RDF statements to the CMD Dublin Core profile should also be mandatory – a requirement which can be extended to some other popular metadata schemas.

In the proximity measure the semantic registries, e.g. the CLARIN Concept Registry (Schuurman et al., 2016), the Dublin Core metadata elements and terms, and special Linked Data repositories like Schema.org (2016) and sameas.org (2016), will play an important role.

8.7 Conclusion

This first full-fledged implementation of the mapping of Component Metadata to Linked Data already enables powerful queries that cross the line between metadata and content, which is in general prominent in the traditional metadata domain but less so in Linked Data. The future plans outlined will make it possible to more easily switch back and forth between these XML and RDF-based approaches, making the information on language resources available in the CLARIN infrastructure more widely available.

Acknowledgements

The authors would like to thank Matej Durco for the initial work on the mapping from CMD to RDF, which formed the foundation for the CLARIN-NL CMD2RDF project in which we were able to extend this into an actual working system architecture.

References

T. Berners-Lee, L. Masinter and M. McCahill (1994). *Uniform Resource Locators (URL)*. IETF. December, 1994.

T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler and F. Yergeau (2008). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C. November 26, 2008.

D. Brickley and R.V. Guha (2014). *RDF Schema 1.1*. W3C. February 25, 2014.

D. Broeder, M. Windhouwer, D. Van Uytvanck, T. Goosen and T. Trippel (2012). CMDI: a Component Metadata Infrastructure. In the *Proceedings of the Metadata 2012 Workshop on Describing Language Resources with Metadata: Towards Flexibility and Interoperability in the Documentation of Language Resources*. LREC 2012. Istanbul, Turkey, May 22, 2012.

C. Chiarcos, S. Hellmann, S. Nordhoff, S. Moran, R. Littauer, J. Eckle-Kohler, I. Gurevych, S. Hartmann, M. Matuschek and C.M. Meyer (2012). *The Open Linguistics Working Group*. LREC 2012. Istanbul, Turkey, May 23–25, 2012.

P. Cimiano, J.P. McCrae and P. Buitelaar (2016). *Lexicon Model for Ontologies: Community Report*. W3C Ontology-Lexicon Community Group. May 10, 2016.

CLARIAH (2016). clariah.nl. Accessed on February 12, 2016.

CLARIN (2016a). *Component Metadata*, www.clarin.eu/cmdi. Accessed on January 18, 2016.

CLARIN (2016b). *Virtual Language Observatory*, vlo.clarin.eu. Accessed on January 18, 2016.

CLARIN-NL (2016). *CMD2RDF data,* portal.clarin.nl/node/4226. Accessed on January 18, 2016.

R. Cyganiak and A. Jentzsch (2016). *The Linking Open Data cloud diagram*, lod-cloud.net. Accessed on January 18, 2016.

R. Cyganiak, D. Wood and M. Lanthaler (2014). *RDF 1.1 Concepts and Abstract Syntax*. W3C. February 25, 2014.

DBpedia (2016). wiki.dbpedia.org. Accessed on January 19, 2016.

M.S. Dryer and M. Haspelmath (eds.) (2013). *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology. 2013.

Dublin Core Metadata Initiative (2016). dublincore.org. Accessed on February 12, 2016.

M. Durco and M. Windhouwer (2014a). From CLARIN Component Metadata to Linked Open Data. In *Proceedings of the third Workshop on Linked Data in Linguistics: Multilingual Knowledge Resources and Natural Language Processing* (LDL 2014). LREC 2014. Reykjavik, Iceland, May 27, 2014.

M. Durco and M. Windhouwer (2014b). *The CMD Cloud*. LREC 2014. Reykjavik, Iceland, May 28–30, 2014.

M. Dürst and M. Suignard (2005). *Internationalized Resource Identifiers (IRIs)*. IETF. January, 2005.

S. Gao, C.M. Sperberg-McQueen and H.S. Thompson (2012). *W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures*. W3C. April 5, 2012.

ISO 24622-1 (2015), *Language resource management - Component Metadata Infrastructure (CMDI) - Part 1: The Component Metadata Model*. ISO. January 20, 2015.

C. Lagoze and H. Van de Sompel (2015). *The Open Archives Initiative Protocol for Metadata Harvesting*. OAI, January 8, 2015.

LIDER project (2016). *Linguistic Linked Open Data*, linguistic-lod.org. Accessed on January 18, 2016.

F. Maali and J. Erickson (2014). *Data Catalog Vocabulary* (DCAT). W3C. January 16, 2014.

J.P. McCrae, P. Cimiano, V. Rodriguez-Doncel, D. Vila-Suero, J. Gracia, L. Matteis, R. Navigli, A. Abele, G. Vulcu and P. Buitelaar (2015). Reconciling Heterogeneous Descriptions of Language Resources. *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications* (LDL 2015). ACL-IJCNLP 2015. Beijing, China, July, 2015.

M-Files (2016). *The history of Metadata*, m-files.com/en/infographic-the-history-of-metadata. Accessed on January 18, 2016.

Open Archives Initiative (2016). *Object Reuse and Exchange,* www.openarchives.org/ore/. Accessed on February 12, 2016.

Open Knowledge Foundation (2016). *Linked Open Vocabularies*, lov.oknf.org. Accessed on January 19, 2016.

OpenLink Software (2016). *Virtuoso Open-Source Edition*. virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/. Accessed on January 19, 2016.

<sameAs> (2016). sameas.org. Accessed on February 12, 2016.

Schema.org (2016). schema.org. Accessed on February 12, 2016.

I. Schuurman, M. Windhouwer, O. Ohren and D. Zeman (2016). CLARIN Concept Registry: the new semantic registry In K. De Smedt (ed.), *Selected Papers from the CLARIN 2015 Conference* Linköping Electronic Conference Proceedings April, 2016.

Summer Institute for Linguistics (2016). *ISO 639-3*, www-01.sil.org/iso639-3/. Accessed on January 19, 2016.

D. Van Uytvanck, H. Stehouwer and L. Lampen (2012). *Semantic metadata mapping in practice: the Virtual Language Observatory*. LREC 2012. Istanbul, Turkey, May 23–25, 2012.

W3C SPARQL Working Group (2013). *SPARQL 1.1 Overview*. W3C. March 21. 2013.

W3C Web Annotation Working Group (2016). www.w3.org/annotation/. W3C. Accessed on January 18, 2016.