

The META-SHARE OWL ontology for the interoperability of linguistic datasets on the Web

Philipp Cimiano¹, Jorge Gracia², Penny Labropoulou³, John P. McCrae¹,
V́ctor Rodŕguez Doncel², and Marta Villegas⁴

¹ Cognitive Interaction Technology, Excellence Cluster, Bielefeld University, Germany
{cimiano, jmccrae}@cit-ec.uni-bielefeld.de

² Ontology Engineering Group, Universidad Polit́cnica de Madrid, Spain
{jgracia, vrodriguez}@fi.upm.es

³ ILSP/Athena R.C., Athens, Greece
penny@ilsp.athena-innovation.gr

⁴ University Pompeu Fabra, Barcelona, Spain
marta.villegas@upf.edu

Abstract. Keywords: keywords

1 Introduction [JPM, JG]

Language technology applications require language resources (LRs) as core components. Aimed at advancing research and adoption of LRs, several initiatives have recently emerged that index metadata of LRs in digital repositories, thus making them easier to find and reuse by humans. One of the most remarkable of such initiatives is META-SHARE [16] (www.meta-share.eu), an open, integrated, secure and interoperable exchange infrastructure where LRs are documented, uploaded and stored in repositories, catalogued and announced, downloaded, exchanged and discussed, aiming to support a data economy. To this end, a complex and rich in information metadata schema covering the whole lifecycle of a LR from production to usage has been designed, while high quality metadata is sought after as an important step in this endeavour. The schema has been implemented in the form of an XML Schema Definition (XSD) while descriptions of individual LRs conforming to this definition are available as XML records/instances/files. PL: How do we know it's a small percentage? compared to what? Also, the description of CLARIN is not very precise; to check again However, META-SHARE still covers only a small percentage of LRs and its resource-intensive curation methodology means it is unlikely to cover all LRs. In contrast, there have been a number of initiatives that have attempted to collect much more metadata from either existing institutional repositories [4, CLARIN] or by crowd-sourcing this data from researchers [7, LRE-Map]. The former method has led to data in different incompatible formats and the latter to noisy, incomplete and duplicate records. In this paper, we propose a solution to these issues by means of the development of a single ontology for the representation of LRs based on the original META-SHARE schema, but represented

using the Web Ontology Language [15] and building on an existing standard, namely DCAT [14]. We take as starting point the work developed by Villegas et al. [19] for the UPF’s META-SHARE node, covering part of the original schema, however extending this initial effort to the whole schema and all LRs and incorporating the consensus reached in the context of the W3C Linked Data for Language Technologies (LD4LT) community group⁵. This is a necessary step to the development of a resource called LingHub⁶, which incorporates META-SHARE data as well as data from other sources and aims to make it queriable by humans and software agents. We hope that this will improve the representation of LR metadata in two fronts. Firstly, the use of Semantic Web techniques (i.e., OWL, RDF) allows to interlink different LRs metadata among themselves and with other external resources on the Web of Data, and enables standardized means of representing and accessing the data (e.g., via SPARQL) thus not relying on domain-specific data formats or proprietary APIs. Secondly, we hope that the use of this ontology will enable the representation of metadata in a manner that allows existing resources to adopt a common core vocabulary, while still being able to represent specific extensions to their existing model and we evaluate this hypothesis by reference to the CLARIN and LRE-Map data models. As an ancillary contribution of this paper, we also describe our experience in technically converting the META-SHARE schema and data to RDF. This was unusually complex as the META-SHARE schema is very complex and as such we needed to develop a new tool, we call the Lightweight Invertible XML to RDF Mapping Language (LIXR), and we demonstrate quantitatively how this ameliorated the process of conversion, and thus as a result proved to be a tool that may significantly help in future conversions from XML. The rest of this paper is structured as follows: In section 2 we will describe the related work in the fields of LR metadata and metadata harmonization. The development of the META-SHARE ontology is described in section 3 and then in section 4 we describe how the data was converted for use in the LingHub portal and how the ontology was used for other data sources in that resource. Finally, in section 5 we consider the broader impact of this ontology as a tool for computational linguists and as a method to realize an architecture of (linked) data-aware services.

2 Related Work [JPM]

The task of finding common vocabularies for linguistics is of wide interest and several general ontologies for linguistics have been proposed. The General Ontology for Linguistic Description [9, GOLD] was proposed as a common model for linguistic data, but its relatively limited scope and low coherence has not lead to wide-spread adoption. An alternative approach that has been proposed is to use ontologies to create coherence among the resources, in particular either by using ontologies to align different linguistic schemas [8] or by means of agreed identifiers [12]. For the particular case of linguistic metadata there have

⁵ <https://www.w3.org/community/ld4lt>

⁶ <http://linghub.org/>

been a number of attempts to define basic metadata for linguistic resources, most notable the Open Language Archive Community [2, OLAC] which built on the Dublin Core metadata. A similar initiative, that provided more structured metadata was the ISLE Metadata Initiative [5, IMDI]. The CLARIN project has played an important role in collecting information about metadata and in particular proposed a common system by which metadata from disparate sources could be aggregated. This system, called the Component Metadata Infrastructure [6], involved the bringing together and sharing of individual data "profiles", which are already in use for different resource types by different user communities (e.g. for literary texts, for corpora as used by social scientists, for video corpora as used by linguists etc.), by means of customized XSD schemes. Data profiles are themselves created on the basis of "components", which are description building blocks consisting of semantically close elements. As we observe in section 4.2, this has in practice merely resulted in each contributing institute using its own scheme, with very little commonality between different institutes. To improve this situation it was recently proposed that the conversion of these CMDI schemas to RDF would enable better interoperability [18], however it is not clear if this project has been realized.⁷

3 The META-SHARE Ontology

3.1 Original MS XSD schema[PL]

The design of the META-SHARE schema [11] has been based upon previous similar efforts and metadata schemas used for the description of LRs as well as user needs recorded for the META-SHARE infrastructure. It has, therefore, been designed not only as an aid for LRs' search and retrieval processes but also as a means to fostering their production, use and re-use by bringing together knowledge about LRs and related objects and processes. Thus, the schema purports to encode information about the whole lifecycle of the LR from production to usage stages: for instance, information about funding is of interest to policy makers, about creation tools and processes can serve as a model for other LR producers, about the use of LRs in various projects and research papers for specific applications shows their usefulness and can be recommended to prospective LR consumers working on the same area. The central entity of the META-SHARE schema is the LR per se, which encompasses both **data sets** (textual, audio and multimodal/multimedia corpora, lexical data, ontologies, terminologies, computational grammars, language models, etc.) and **technologies (tools/services)** used for their processing. It should also be stressed here that the term "LR" in META-SHARE is intended for whole sets of text/audio/video etc. files (corpora), sets of lexical entries (lexical/conceptual resources), integrated tools/services and so on, rather than individual items (e.g. single texts, such as journal articles, poems in corpora or noun entries in lexica). In addition to the central entity, other

⁷ JPM: I emailed Menzo Windhouwer about this and may change this statement based on his response, if any

entities are also documented in the schema; these are reference documents related to the LR (papers, reports, manuals etc.), persons/organizations involved in its creation and use (creators, distributors etc.), related projects and activities (funding projects, activities of usage etc.), accompanying licenses, etc. Thus, the schema recognizes the following distinct "satellite entities":

- the actor, further distinguished into person and organization,
- the project,
- the document, and
- the licence.

These are described only when the case arises, i.e. when they are linked to a specific LR. For their description, other schemas and guidelines that have been devised specifically for them (e.g. BibTex for bibliographical references) have been taken into account. The META-SHARE schema proposes a set of elements to encode specific descriptive features of each of these entities and relations holding between them, taking as a starting point the LR. Following the CMDI approach, these elements are grouped together into "components", which act as placeholders for well defined categories of information: for instance, the communication component of a person or organisation includes elements on email, postal address, telephone, url etc., while the identification component of a LR brings together elements required to identify it, such as the LR's full and short names, identifiers, a short description of its contents etc. One of the advantages of this mechanism is that it allows for a better structuring of the information, which is crucial for a complex schema like META-SHARE. The core of the schema is the `resourceInfo` component (Figure 1), which subsumes components that combine together to provide the full description of a resource and its lifecycle. For each LR there are:

- **PL: instead of enumerating them, maybe I should just give a couple of examples** administrative components common to all LRs: `identificationInfo`, `distributionInfo`, `contactPerson`, `metadataInfo`, `versionInfo`, `validationInfo`, `usageInfo`, `resourceDocumentationInfo`, `creationInfo` and `relationInfo`;
- **PL: introducing resourceType & mediaType without any explanations; see if it's needed** components specific to *resourceType* and *mediaType* combinations (the two classification axes of the schema), that cater for the encoding of information relevant to text, audio, video and image parts of corpora, text, audio, video and image parts of lexical/conceptual resources etc.; broadly speaking, these cover information related to contents, formatting, classification etc. which differ depending on the resource/media combination (e.g. genre takes different values for texts and videos, creation processes and tools are described differently for texts and videos etc.) .

The META-SHARE schema has been implemented as an XSD (available at [GITHUB](#)). An integrated environment supports the description of LRs, either from scratch or through uploading of XML files adhering to the META-SHARE metadata schema, as well as browsing, searching and viewing of the LRs.

3.2 Purpose of the ontology [MV,JG,JPM]

(e.g., why do RDF and OWL for an already defined vocabulary?)

3.3 Formal modelling and mapping issues [MV, JPM, PL]

When mapping an XML scheme to RDF there are naturally differences that must be accounted for, which generic mapping methodologies cannot accommodate without tending to a high degree of verbosity. PL: we say this elsewhere also; decide where to put The META-SHARE metadata model is formalised in a XSD schema that 'transcodes' a component-based model as suggested by CLARIN [6]. Essentially, the component-based approach revolves around two central concepts: *elements* and *components*. *Elements* are used to encode specific descriptive features of the resources and are linked to conceptually similar existing elements in the Dublin Core and/or the ISocat registry. *Components* are complex elements and can be seen as bundle of semantically coherent *elements*. In the META-SHARE XSD schema, *elements* are formalized as simple elements whereas *components* are formalized as complex-type elements. When mapping the XSD schema to RDF, *elements* can be naturally understood as properties (e.g. name, gender, etc.). *Components* (i.e. complex-type elements), however, deserve a careful analysis. General mapping rules from XSD to RDF establish that a local element with complex type translates into an object property and a Class. An insight analysis of the META-SHARE schema showed that the straightforward application of such a principle may derive into unnecessary verbose graphs. PL: to better represent what is in the XSD schema, I think the highlighted paragraph could be replaced by the following: META-SHARE distinguishes between two kinds of *components*, namely:

- 'special status components': these are used for the representation of three satellite entities (persons, organizations and documents), which can be re-used throughout the model with different roles: validators, annotators, resource documentation, validation reports etc. These are implemented as components(personInfo, organizationInfo, documentInfo) which can be used for elements denoting the roles: validator, annotator, contactPerson, validation-Report etc. Moreover, some of these elements are implemented as a choice between two components: annotator can be implemented as either a personInfo or an organizationInfo; documentation is used to bring together the choice between a structured documentInfo component (intended to be filled in like a bibliographic record) and a simple element 'documentUnstructured' (allowing for typing in links to or titles of simple readme files)Marta, are these choices the complex elements or linked components? what about contact-Person which is implemented directly as a personInfo?
- 'normal components', which simply group together semantically coherent information (e.g. metadataInfo, validationInfo etc.).

META-SHARE distinguishes between three kinds of *components*, namely: 'special status components', 'linked components' and 'bare components'. The

former include concepts such as person and document and they can be attached to various *components* performing different roles (i.e. creator, validator, documentation, etc.). 'Linked components' can be understood as relations between *components* and include concepts such as validationReport or validator, among many others. Finally, 'bare components' are used to group together semantically coherent information (i.e. metadataInfo, validationInfo etc.).

In the XSD schema, 'special status components' are formalised as complex types whereas 'linked components' are complex elements. Thus, when applying the conversion rules, the special status components become Classes and the linked components become object properties which correctly captures the semantics behind. For 'bare components' things are more complex as they are formalised as both complex elements and complex types. This means that the general conversion rule will produce an object property and the corresponding Class which, in most cases, may be unnecessary. For example: in the META-SHARE schema, the resourceInfo node contains a number of elements which organise information into coherent sets ⁸:

```
resourceInfo/identificationInfo(1)/...
resourceInfo/distributionInfo(1)/...
resourceInfo/contactPerson(n)/...
resourceInfo/metadataInfo(1)/...
resourceInfo/versionInfo(1)/...
resourceInfo/validationInfo(n)/...
resourceInfo/usageInfo(1)/...
resourceInfo/resourceDocumentationInfo(1)/...
resourceInfo/resourceCreationInfo(1)/...
resourceInfo/relationInfo(1)/...
resourceInfo/resourceComponentType(1)/...
```

For elements such as the identificationInfo above, the application of the rule will produce an unnecessary node. Following [19], we identified potentially removable nodes before the actual RDFication process. The criteria applied take into account the tree structure of the nodes, their cardinality and the XPath axes. Thus, embedded complex elements with cardinalityMax=1 are identified as potentially removable, provided they do not contain text nor attributes. This allows for a simplification of the model, as exemplified below.

```
resource/identificationInfo/resourceName
resource/identificationInfo/description
resource/identificationInfo/resourceShortName
resource/identificationInfo/url
```

```
becomes
resource/resourceName
resource/description
resource/resourceShortName
resource/url
```

⁸ We use XPath expressions. Number in brackets shows nodes cardinality.

Note that such a simplification rule can be applied provided this does not derive in sibling conflicts: promoted nodes may cause naming conflicts in their new axe. Thus, a careful checking is needed in order to avoid possible clashes.

Interestingly enough, the removal of the superfluous wrapping elements has also led to a change of philosophy to the schema and a need for re-structuring in order to ensure that properties are attached to the most appropriate node, as exemplified and discussed in the 3.5

Besides the 'bare elements' described so far, other potentially superfluous nodes were also identified: namely complex elements with one and only one simple element. This is, for example, what happens with the path validation-Info/validationTool/targetResourceNameURI. In such cases the terminal node can be removed. PL: I think the targetResourceNameURI is the only such case; correct?

Beyond this, we made the following extensions to our mapping strategy:

- We decided to rename some of the elements when falling into one of the following categories: (a) removed the Info suffix from the wrapping elements: e.g. validationInfo becomes simply validation PL: check tomorrow all classes and make a list as promised (b) changed the names of elements that created confusion, as already noted by the META-SHARE group and/or the ld4lt group; thus, 'resource' was renamed 'languageResource', 'restrictionsOfUse' became 'conditionsOfUse', etc. (c) PL: I lost some text and I can't remember what I had here; tomorrow... (d) shortened some names such as ConformanceToBestStandardsAndPractices **JPM: Perhaps we introduce sameAs links to handle this; PL: I think we decided against d; pls confirm**
- Developed novel classes based on existing values, e.g., Corpus $\equiv \exists \text{resourceType.corpus}$ **PL: IMPORTANT: discuss what we do with resourceComponentType, corpusMediaType, corpusTextInfo etc.; what remains and what is removed; tomorrow...**
- Removing unnecessary properties such as **versionInfo**. PL: I think this is the same as **identificationInfo**; if yes, removed
- Generalized elements such as **notAvailableThroughMetashare** to **availableThroughOtherDistributor**
- Simplified some complex structures, such as **membershipInfo** PL: come back to this tomorrow
- Grouping similar elements under novel superclasses, e.g., **DiscourseAnnotation**, **genre** PL: one of the advantages of the RDF approach; say a bit more
- Extending existing classes with new values and including new properties (see section 3.5)

3.4 Interface with DCAT and other vocabularies [JPM]

The META-SHARE model can be considered broadly similar to DCAT in that there are classes that are nearly an exact match to ones in DCAT for three out of four cases. DCAT's *dataset* corresponds nearly exactly to the *resource info* tag and similarly, *distributions* are similar to *distribution info* classes and *catalog record* is similar to *metadata info*. The fourth main class, *catalog* covers a level not modelled by META-SHARE.

DCAT uses Dublin Core properties for many parts of the metadata, and often these properties are in fact deeply nested into the description. For example, language is found in several places deeply nested under six tags⁹ This is in accordance to the META-SHARE view that a language resource may consist of modules with different media types, which have different properties and need to be described in different terms: for instance, a multimedia corpus may have a video module (the moving image part per se), a video module for the dialogues which can be separated from the video, and three text modules for the subtitles, the transcription of the dialogues and the scripts. These modules can have different properties, e.g. the dialogues and the scripts may be in English, but the subtitles can be in French and German (two translations). Thus, language as a property is attached not to the languageResource but to each module. Even after removing the superfluous nodes, language will still be embedded at a deeper level, although not as deep as in the XSD schema.

Similarly, it also the case that some Dublin Core properties are not directly specified in the META-SHARE model, but can be inferred from related properties, e.g., Dublin Core’s ‘contributor’ follows from people indicated as ‘annotators’, ‘evaluators’, ‘recorders’ or ‘validators’. Similarly, several DCAT specific-properties, such as ‘download URL’, are nearly exactly equivalent to those in Metashare but occur in places that do not fit the domain and range of the properties. In this particular case, it was a simple fix to move the property to the enclosing **DistributionInfo** class. Inevitably, several properties from DCAT did not have equivalences in META-SHARE, notably ‘keyword’ and ‘byte size’. We **did something about this... I am not sure what though** PL: isn’t this in sizeInfo?

3.5 Licensing module [VRD, PL]

Skeleton We describe first the Metashare schema, whose licensing information is described in an independent XML Schema file, available on git¹⁰. We discuss on the needs that motivated the evolution from the previous model. We describe (if not done before) also the procedure and methodology. Short introduction on the ODRL vocabulary. We describe the most important changes that we have introduced. And going beyond ODRL: License Templates as an easy entry points for Semantic Web - laymans. Example of license template, example of license. Directly in TTL. Maybe introducing a figure depicting what is metadata for resource/distribution/license?

⁹ `resourceInfo > resourceComponentType > corpus* > corpusMediaType > corpusVideoInfo >`

¹⁰ <https://github.com/metashare/META-SHARE/blob/master/misc/schema/v3.0/META-SHARE-LicenseMetadata.xsd>

4 META-SHARE in LingHub

LingHub ¹¹ is a large resource containing information about a wide range of language resources, but unlike META-SHARE it does not directly collect this information, but instead harmonizes the metadata from a wide range of sources. In this section, we will first describe how the original META-SHARE data was translated into RDF and the alignment with DCAT [14], previously described, was achieved. Furthermore, we will then consider how we have used the META-SHARE vocabulary as a base vocabulary to align terms from other resources included in LingHub.

4.1 Mapping META-SHARE to RDF [JPM]

When translating XML documents into RDF, one of the most common approaches is based on Extensible Stylesheet Language Transformations (XSLT) [21, 17, 3], which has been extended by some authors into a significant framework [13]. However, XSLT has a number of disadvantages for this task:

- The set of functions and operators supported by most processors is limited.
- Limited ability to declare new functions.
- Does not allow stream (SAX) processing of large files.
- XSLT is a one-way transformation language and it is not possible to ‘round-trip’ the conversion, i.e., convert RDF to XML.
- XSLT syntax is expressed in XML and thus is very verbose and aesthetically unpleasing. For this reason, many people use alternative more compact syntaxes¹²¹³

Furthermore, the META-SHARE syntax is very complex consisting of 111 complex types and 207 simple types. As such we deemed that the development of a new language for transformation and writing our converter in that language would take less development effort than writing a conversion entirely in XSLT. The mapping methodology we developed is a domain-specific language [10] called Lightweight Invertible XML to RDF Conversion (LIXR) and aims to improve on the situation by fixing the concerns above. To begin with we selected the Scala programming language as the basis for LIXR as it has a proven syntactic flexibility that makes it easy to write domain-specific languages [20]. A simple example of a LIXR mapping is given below:

```
object Metashare extends eu.liderproject.lixr.Model {
  val dc = Namespace("http://purl.org/dc/elements/1.1/")
  val ms = Namespace("http://purl.org/ms-lod/MetaShare.ttl#")
  val msxml = Namespace("http://www.ilsp.gr/META-XMLSchema")
  msxml.resourceInfo --> (
    a > ms.ResourceInfo,
    handle(msxml.identificationInfo)
```

¹¹ <http://linghub.org>

¹² Compact XML: <https://pythonhosted.org/compactxml/>

¹³ Jade: <http://jade-lang.com/>

```

)
msxml.identificationInfo --> (
forall(msxml.resourceName)(
dc.title > (content @@ att("lang"))
)
)
}

```

In this example, we first create our model extending the basic LIXR model and define namespaces as dynamic Scala objects¹⁴. We then make two mapping declarations for the tags `resourceInfo` and `identificationInfo`. LIXR (as XSLT) simply searches for a matching declaration at the root of the XML document to begin the transformation. Having matched the `resourceInfo` tag, the system first generates the triple that states that the base element has type `ms:resourceInfo`, and then ‘handles’ any children `identificationInfo` tags by searching for an appropriate rule for each one. For `identificationInfo` the system generates a triple using the `dc:title` property whose value is the content of the `resourceName` tag tagged with the language given by the attribute `lang`. To evaluate the effectiveness of our approach we compared directly with

Name	Tags	Implementation	LoC	LoC/Tag
ine TBX	48	Java	2,752	57.33
CLARIN (OLAC-DMCI)	79	XSLT	404	5.11
CLARIN (OLAC-DMCI)	79	XSLT (Compact Syntax)	255	3.22
TBX	48	LIXR	197	4.10
CLARIN (OLAC-DMCI)	79	LIXR	176	2.23
MetaShare	730	LIXR	2,487	3.41

Table 1. Comparison of XML to RDF mapping implementations, by number of tags in XML schema, and non-trivial lines of code (LoC)

two other XML to RDF transformations, we had carried out in this project, and reimplemented them using the LIXR language. In particular these were the TBX model [1] as well as the OLCA-DMCI profile of the CLARIN metadata¹⁵. In table 1, we see the effort to implement these using LIXR is approximately half of using XSLT and about ten times less than writing a converter from scratch. In addition to the reduction in effort using this approach, we also note several other advantages of the LIXR approach, due to its declarative declaration

- We can easily switch to using a stream-based parse for XML (e.g., SAX) so we can process large files without having to use much memory
- A reverse mapping can be extracted that re-generates the XML from the outputted RDF

¹⁴ This is a newer feature of Scala only supported since 2.10 (Jan 2013)

¹⁵ http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/profiles/clarin.eu:cr1:p_1288172614026/xsd

- We can extract the type, range and domain of RDF entities generated during this procedure. This export formed the initial version of the ontology described in this paper

4.2 Harmonizing other resources with META-SHARE [JPM]

LingHub brings resources from a wide-range of sources and while we can use standards such as DCAT and Dublin Core to guarantee a common representation of the basic Metadata of a resource, there does not a standard for the representation of metadata specific to linguistics. For that reason, we use the META-SHARE model as a standard for other resources to use in LingHub. In particular, we will focus on the application of META-SHARE as a model to harmonize CLARIN data. The CLARIN repository describes its resources using a small common set of metadata and a larger description defined by the Component Metadata Infrastructure [6, CMDI]. These metadata schemes are extremely diverse as shown in table 2.

Component Root Tag	Institutes	Frequency
Song	1 (MI)	155,403
Session	1 (MPI)	128,673
OLAC-DcmiTerms	39	95,370
mods	1 (Utrecht)	64,632
DcmiTerms	2 (BeG,HI)	46,160
SongScan	1 (MI)	28,448
media-session-profile	1 (Munich)	22,405
SourceScan	1 (MI)	21,256
Source	1 (MI)	16,519
teiHeader	2 (BBAW, Copenhagen)	15,998

Table 2. The top 10 most frequent component types in CLARIN and the institutes that use them. Abbreviations: MI=Meertens Institute (KNAW), MPI=Max Planck Insitute (Nijmegen), BeG=Netherlands Institute for Sound and Vision, HI=Huygens Institute (KNAW), BBAW=Berlin-Brandenburg Academy of Sciences

5 Discussion

5.1 Applications of the MetaShare model (beyond LingHub) [MV]

The IULA-UPF CLARIN Competence Centre¹⁶ aims to promote and support the use of technology and text analysis tools in the Humanities and Social Sciences research. The centre includes a Catalogue¹⁷ with information on language

¹⁶ <http://www.clarin-es-lab.org/index-en.html>

¹⁷ <http://lod.iula.upf.edu/>

resources and technology. The Catalogue is based on the initial LOD version of the META-SHARE model as described in [19] and includes full descriptions for 100 NLP Services and 150 language resources. The original data come from the UPF META-SHARE node¹⁸ as XML files compliant with the META-SHARE schema. XML records were converted into RDF and augmented with service descriptions (not included in the UPF META-SHARE node) and relevant documentation (appropriate articles, documentation, sample data and results, illustrative experiments, examples from outstanding projects, illustrative use cases, etc) to encourage potential users to embrace digital tools. Finally, the data was enriched with links, including internal as well as external links. The LOD approach, specially the linking, allowed to maximize the information contained in the original repository and to enrich this by using external repositories and datasets. The original data missed many relevant internal links. For example: in the source model related concepts such as 'Named Entity Recognition' and 'named entity' are unconnected. Similarly, there is no connection between 'semantic annotation' and the relevant standard (SemAF), between 'semantics' and 'semantic roles', 'derivation' and 'morphology', and so on. In the eventual dataset, such relations are explicitly encoded and this allows the Catalogue to provide better browsing functionalities which result in a better understanding of the whole data. For example, when the user gets the 'derivation' page¹⁹ he is advised to see 'morphology' and 'morphological tagging'. External links include sameAs relations and 'reference' relations such as 'creator/contributor', 'subject' and 'references' relations (all from Dublin Core). Catalogue uses the sameAs relations for data mashup. Two procedures were defined to retrieve and display additional data. In the first case, the system gets data from the DBpedia. Thus, for any individual in the dataset having a sameAs property linking to some DBpedia resource, the Catalogue retrieves and displays the 'subjects' for that DBpedia resource. For example, when browsing the Apertium project page²⁰, the Catalogue adds the links to the DBpedia/Wikipedia subjects found there, in this example: "natural language processing tools", "free software programmed in c++" and "machine translation". In the second case, for any person with a sameAs property linking to the DBLP dataset, the system generates a link to the DBLP SPARQL end point with the query to get all publications for that person²¹. 'Reference' relations are used in a simpler way: they do not imply retrieving information from an external SPARQL endpoint but simply provide a link to some external relevant resource. Linking to external resources not only fulfils the principles of LOD but provides the user with the possibility to explore beyond the Catalogue itself. Finally, the Catalogue makes extensive use of the so called backward relations. For any resource page, the system retrieves all triples in which the resource occurs as object of the relation. The subjects are grouped into Classes and in this way the user gets all resources that have some-

¹⁸ <http://metashare.upf.edu>

¹⁹ <http://lod.iula.upf.edu/resources/morpho-Derivation>

²⁰ http://lod.iula.upf.edu/resources/project_Apertium

²¹ http://lod.iula.upf.edu/resources/person_Jorge_Vivaldi

thing to do with the current resource organised into classes. For example, in the IULA-UPF page²² the backwards relations include instances of person, project, services among many other. The Catalogue demonstrates the benefits of the LOD framework and how LOD can be easily used as the basis for a web browser application that maximizes information and helps users to navigate throughout the dataset in a comprehensive way.

5.2 Challenges and future outlooks [PC]

6 Conclusion [JG]

Acknowledgments. We are very grateful to the members of the W3C Linked Data for Language Technologies (LD4LT) for all the useful feedback received and for allowing this initiative to be developed as an activity of the group. This work is supported by the FP7 European project LIDER (610782) and by the Spanish Ministry of Economy and Competitiveness (project TIN2013-46238-C4-2-R). ANYTHING ELSE? PL: ADD CLARIN-EL here for funding!

References

1. Systems to manage terminology, knowledge and content – TermBase eXchange (TBX). Tech. Rep. 30042, ISO (2008)
2. Bird, S., Simons, G.: The olac metadata set and controlled vocabularies. In: Proceedings of the ACL 2001 Workshop on Sharing Tools and Resources-Volume 15. pp. 7–18. Association for Computational Linguistics (2001)
3. Borin, L., Dannells, D., Forsberg, M., McCrae, J.P.: Representing Swedish lexical resources in RDF with lemon. In: Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference (2014)
4. Broeder, D., Kemps-Snijders, M., Van Uytvanck, D., Windhouwer, M., Withers, P., Wittenburg, P., Zinn, C.: A data category registry-and component-based metadata framework. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation. pp. 43–47 (2010)
5. Broeder, D., Offenga, F., Willems, D., Wittenburg, P.: The imdi metadata set, its tools and accessible linguistic databases. In: Proceedings of the IRCS Workshop on Linguistic Databases, Philadelphia. pp. 11–13 (2001)
6. Broeder, D., Windhouwer, M., Van Uytvanck, D., Goosen, T., Trippel, T.: CMDI: a component metadata infrastructure. In: Describing LR with metadata: towards flexibility and interoperability in the documentation of LR workshop programme. p. 1 (2012)
7. Calzolari, N., Del Gratta, R., Francopoulo, G., Mariani, J., Rubino, F., Russo, I., Soria, C.: The LRE Map. Harmonising community descriptions of resources. In: Proceedings of the Eighth Conference on International Language Resources and Evaluation. pp. 1084–1089 (2012)
8. Chiarcos, C.: Ontologies of linguistic annotation: Survey and perspectives. In: LREC. pp. 303–310 (2012)

²² http://lod.iula.upf.edu/resources/organization_UPF-IULA

9. Farrar, S., Lewis, W., Langendoen, T.: A common ontology for linguistic concepts. In: *Proceedings of the Knowledge Technologies Conference*. pp. 10–13 (2002)
10. Fowler, M., Parsons, R.: *Domain-specific languages*. Addison-Wesley Professional (2010)
11. Gavrilidou, M., Labropoulou, P., Desipri, E., Piperidis, S., Papageorgiou, H., Monachini, M., Frontini, F., Declerck, T., Francopoulo, G., Arranz, V., Mapelli, V.: The meta-share metadata schema for the description of language resources. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. pp. 1090–1097 (2012)
12. Kemps-Snijders, M., Windhouwer, M., Wittenburg, P., Wright, S.E.: Isocat: Corraling data categories in the wild. In: *LREC (2008)*
13. Lange, C.: KREXtor – an extensible XML \rightarrow RDF extraction framework. *Scripting and Development for the Semantic Web (SFSW)* (449), 38 (2009)
14. Maali, F., Erickson, J., Archer, P.: Data catalog vocabulary (DCAT). W3C recommendation, The World Wide Web Consortium (2014)
15. Motik, B., Patel-Schneider, P.F., Parsia, B., Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Rutteneberg, A., Sattler, U., Smith, M.: OWL 2 web ontology language structural specification and functional-style syntax. W3C recommendation, The World Wide Web Consortium (2012)
16. Piperidis, S.: The META-SHARE language resources sharing infrastructure: Principles, challenges, solutions. In: *Proceedings of the Eighth Conference on International Language Resources and Evaluation*. pp. 36–42 (2012)
17. Van Deursen, D., Poppe, C., Martens, G., Mannens, E., Walle, R.: XML to RDF conversion: a generic approach. In: *Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on*. pp. 138–144. IEEE (2008)
18. Ďurčo, M., Windhouwer, M.: From CLARIN component metadata to linked open data. In: *Proceedings of the 3rd Workshop on Linked Data in Linguistics*. pp. 13–17 (2014)
19. Villegas, M., Melero, M., Bel, N.: Metadata as linked open data: mapping disparate xml metadata registries into one rdf/owl registry. In: Chair, N.C.C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland (may 2014)
20. Wampler, D., Payne, A.: *Programming Scala*, chap. 11. O'Reilly (2008)
21. Wüstner, E., Hotzel, T., Buxmann, P.: Converting business documents: A clarification of problems and solutions using XML/XSLT. In: *Advanced Issues of E-Commerce and Web-Based Information Systems, International Workshop on*. pp. 61–61. IEEE Computer Society (2002)