The LRE Map Disclosed

Riccardo Del Gratta, Gabriella Pardelli, Sara Goggi
ILC-CNR
Consiglio Nazionale delle Ricerche
Via Moruzzi, 1 - Pisa, Italy
firstname.lastname@ilc.cnr.it

Abstract

This paper describes a serialization of the LRE Map database according to the RDF model. Due to the peculiar nature of the LRE Map, many ontologies are necessary to model the map in RDF, including newly created and reused ontologies. The importance of having the LRE Map in RDF and its connections to other open resources is also addressed.

Keywords: Language Resource, LOD, Metadata

1. Introduction

The LRE Map is an initiative started in conjunction with LREC2010.[1] It was conceived as a campaign for collecting information about the language resources and technologies (Calzolari et al., 2010) underlying the scientific work presented at that conference. The initiative continued with LREC2012, where the role of a coherent and rich documentation of Language Resources (LRs) was clearly identified and pushed to the community, (Calzolari et al., 2012). The rationale behind the LRE Map is the indisputable need of accurate and reliable documentation of LRs to make them really "existing" and available.

In this paper we present a new "vision" for the LRE Map, moving the data of the map from the database[2] towards the world of the (Linguistic) Linked Open Data ((L)LOD).[3]

(L)LOD are still quantitatively a minority within the linked data cloud, (Chiarcos et al., 2011; Chiarcos et al., 2012) but they are growing (Lezcano et al., 2013) and becoming a central modality for linguistic data publication. The LRE Map, though not big in number of triples, has a significant specific weight since it contains a manually developed/checked normalization of all data contained in the database.

The advantage of rendering the LRE Map in RDF/XML (in the following RDF) is the immediate connection to big resources, (Wikipedia, Dbpedia etc.) that are *in nuce* in some metadata of the map, namely the URL and the documentation.

In addition, as a component of the linguistic cloud, the LRE Map will be visible and accessible to a wider community.

2. The LRE Map: from limited To quasi-open semantics

The LRE Map database collects the set of metadata (type, name, use, status and other information) the authors assign to the Language Resources they use and/or describe during the submission procedure of conferences.

The rationale behind the LRE Map is to let authors to ride the tide of their judgments about the LRs. The power and the novelty of the map is the fact that it collects, for example, as many WordNets as authors decide to describe, in different conferences, in different years and for different purposes.

As a database, the semantics in the map is limited: no complex assertions can be explicitly formalized and no reasoning can be performed, excluded the one that the SQL languages provides.

The intrinsic importance of the metadata is left to the list of values a specific metadata can assume. For examples, aggregates such as how many *freely available* Language Resources, of which resource *type* and how these numbers change in time and over conferences are essential for people interested in studying such specific trends.

2.1. The LRE Map is submission-centric

Because of the collecting methods we have used to populate the LRE Map, the resulting map is *submission-centric*, the submission being a link among authors, papers and Language Resources: a given submission presents a paper (along with its authors) and describes one or more LRs. From a logical point of view, we can assert that *is* the paper which is linked to the described Language Resources, but it was necessary to distinguish between submission and paper since some conferences only provide data containing anonymous submissions -together with the resources they are linked to- while in other cases (notably LREC) a full description of the papers is also available. Hence the necessity of a simple submission object, which is only identified by a code and by the reference to its related conference, and which may or may not be enriched by further information on the actual paper. As a consequence, the records in the database reflect this organization and are arranged as in Figures 1 and 2, in which the objects have been logically grouped.

2.2. The LRE Map as a collection of "instances"

Before physically transforming the LRE Map database into RDF, we need to slightly change our point of view on the

[1] This initiative has been partially funded by the FLaReNet Thematic Network (http://www.flarenet.eu).

[2] The LRE Map database is currently accessible through a basic interface, available at http://www.resourcebook.eu.

[3] http://linguistics.okfn.org/resources/llod/.

Submission:	S1
Name:	WordNet
Type:	Lexicon
Availability:	Freely Available
Use:	Summarization
Paper:	P1
Author(s):	A1,A2
....:	....

Submission:	S2
Name:	WordNet
Type:	Lexicon
Availability:	freely-available
Use:	Knowledge Discovery
Paper:	P2
Author(s):	A3,A4
....:	....

Figure 1: WordNet example from LREC 2010.

S1:
- Conference: $LREC$
- Year: 2010
- Resource(R1):
  - Name: WordNet
  - Type: Lexicon
  - Availability: Freely Available
  - Use: Summarization
  - ....
- paper(P1):
  - Title: $T1$
  - Author(s):
    - A1
    - A2
    - ....
  - ....

Figure 2: WordNet example: first record with object logically grouped

map itself. Since we will adapt a set of "ontologies" (cf. section 3.) it is necessary to refer to the map using a proper terminology: we should look at the LRE Map as a collection of resource instances rather than as a collection of database records.

As an ontology, standard ontologic operations can be made on the LRE Map. The term ontology, however, should not suggest, here, the fact that the LRE Map will be a resource where a full reasoning can be made: in fact, it is impossible to derive information on facts such as "Resource A is used for creating Resource B" or similar. It will rather be a schema for defining basic ontologic aspects of LRs such as inheritance and properties definition.

2.3. The LRE Map changes language

We can look at Figure 2 from a different point of view. The fact that the values in the record(s) are interconnected allow us to assign *semantics* to their logical grouping. For example, the connection between the submission *S1* and the paper *P1* can be transformed in a triple:

$$S1 \text{ hasDocument } P1$$

likewise the connection between the paper and its author(s):

$$P1 \text{ authorList } [A1, A2]$$

Figure 3 shows how the connections between the submissions and other objects are interpreted as triples.[4]

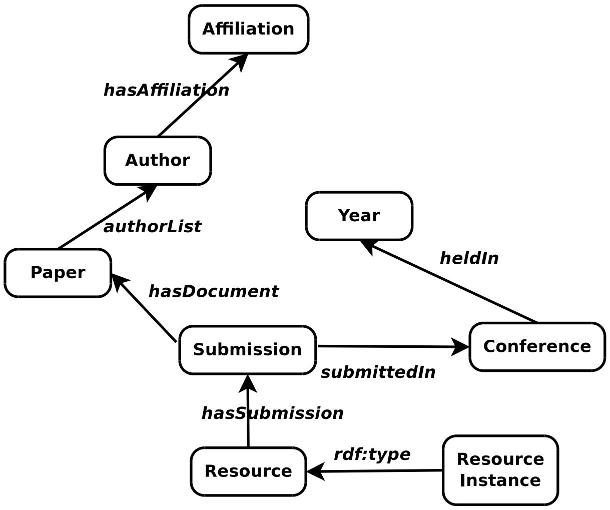


Figure 3: The submission and related entities

3. The LRE Map set of ontologies

Figure 3 contains specific ontologies needed for covering the different aspects contained in the LRE Map structure: namely years, conferences, authors, affiliations and papers. Some of these ontologies have been reused and/or slightly customized, while other have been created from scratch. The LRE Map ontology needs to import and use such ontologies and define properties among their instances if they have not been already defined. In fact, in Figure 3, the property *authorList* comes from `bibo` while *heldIn*, which links instances of the *Conference Ontology* to the *Year Ontology* has been defined. The submission-centric aspect of the LRE Map is managed by introducing the *Submissions Ontology*. The latter collects the identifiers of submissions as its instances and it is connected to *Papers* and *Language Resource* through the *hasDocument* and *hasSubmission* properties respectively.

In what follows the notation:

[Model: ontology_model; Instance Collection: ontology_instances]

specifies that we keep separate, when possible, the model, the ontology schema, from the collection of individuals (instances) of such schema.

On the contrary,

[Model and Instance Collection ontology_instances]

specifies that the modelling schema also contains the instances.

[4] We have added the *Affiliation*, which is connected to the *Author* entity.

3.1. Reused and/or customized ontologies

This section briefly describes the available ontologies that have been reused within the LRE Map frame.

Author Ontology [Model: obj_auth; Instance Collection: lremap_auth] We have modelled the *Author* object using `foaf`[5] for addressing aspects such as email, first and last name and affiliations. The latter (*Affiliations*) have been enriched with geographical features such as their countries according to `geonames`[6] for managing properties in terms of geographical figures.
The records of the database of the LRE Map which contain the *Author* data (the *Author* instances) are inserted into a different documents lremap_auth which, on the hand imports obj_auth, and on the other defines the property *hasAffiliation* used to connect authors' instances to affiliations. In principle authors can have more than one affiliation.

Paper Ontology [Model and Instance Collection: lremap_paper] We have modelled the *Paper* object using `bibo`[7] for addressing aspects such as title, list of authors, status, topics, etc. The lremap_paper ontology imports lremap_auth to manage the connections between authors and papers. Formally we have used the property *authorList*, which is a `bibo` property to connect a paper instance to the list of its authors.

3.2. Newly defined ontologies

Year Ontology [Model and Instance Collection: lremap_year] The temporal dimension of the ontology is pretty much straightforward: it is the year when the Language Resource has been described. The lremap_year ontology is the simplest year ontology which is coherent with the LRE Map entries: a list of *years*.

Conference Ontology [Model and Instance Collection: lremap_conf] The conference ontology (lremap_conf) has a list of *conferences* as individuals. They represent the conferences which used the LRE Map for their submission phase. If the year ontology addresses the question "when the resource X has been described", this ontology handles the spatial dimension, that is "in which conference(s) the resource X has been described".
The object property *heldIn* has been defined to connect *Conference* and *Year* ontologies;

Submission Ontology The submission ontology, lremap_subs, is a simple list of identifiers. Submissions are the bridge between *Papers*, *Conferences* and *Resources* to which they are connected through the properties, *hasDocument* and *submittedIn* respectively. *Resources* and *Submissions* are connected through the *"hasSubmission"* relation;

Resource Ontology [Model: lremap_resource; Instance Collection: lremap_ri] *Language Resources instances* (lremap_ri) describe the LRs (lremap_resource) in terms of descriptive metadata in the same way that blueprints describe the items that form buildings. Just as a blueprint can be used to create multiple buildings, a single *Resource* can be used to model as many *instances* as authors provided during the conference submission phase, cf. section 4.

4. The Language Resource model and instances

We decided to base the lremap_resource schema on the set of metadata provided to the authors for the submission phase of their articles. This means that we have created a *static* schema which formalizes those values. In addition we took apart the *type* metadata from the others, this because we are aware that the specific *type* of a LR is more significant than the other set of metadata. In other words we decided to identify the *type* with the *hierarchical* axis:

$$\text{resource\_i is-a } type \text{ and } hasSomeMetadata \text{ Md\_i} \qquad (1)$$

Assertion 1 establishes that the resource_i inherits some specific features from its *type* and it is enriched with the collection of its metadata.
The example reported in Figure 4 is the same of Figure 1 but we have added properties to metadata to make assertion 1 clearer:

Submission:	S1 *"hasSubmission"*
Name:	WordNet *"hasName"*
Type:	Lexicon **is-A**
Availability:	Freely Available *"hasAvailability"*
Use:	Summarization *"hasUse"*
Modality:	Written *"hasModality"*
....:	....

Figure 4: WordNet example along with *hasMetadata_i* properties

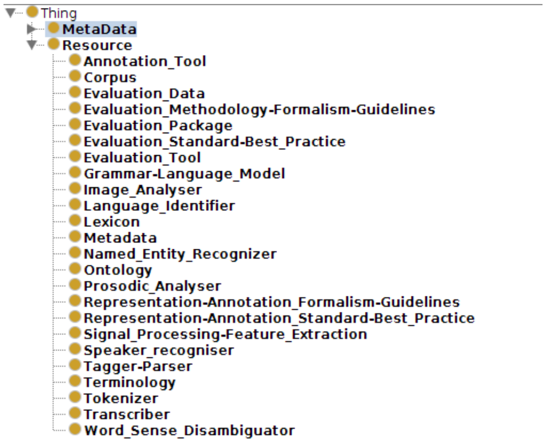
In building the lremap_resource schema from the records of the database we agreed on the following strategy:

Keep the *type* apart from other metadata The classes of the proposed model have been structured according to Figure 5. The class *Resource* has as many sibling subclasses as the different *types* which have been provided to the authors at the submission stage: *Corpus*, *Lexicon*, *Tokenizer* and so on. These subclasses do not contain individuals;

[5] http://xmlns.com/foaf/0.1/.

[6] http://www.geonames.org/ontology/ontology_v3.1.rdf.

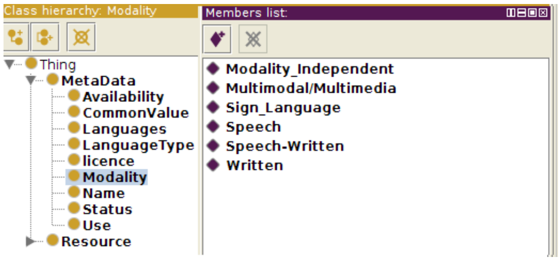
[7] http://purl.org/ontology/bibo/.



- Thing
  - MetaData
  - Resource
    - Annotation_Tool
    - Corpus
    - Evaluation_Data
    - Evaluation_Methodology-Formalism-Guidelines
    - Evaluation_Package
    - Evaluation_Standard-Best_Practice
    - Evaluation_Tool
    - Grammar-Language_Model
    - Image_Analyser
    - Language_Identifier
    - Lexicon
    - Metadata
    - Named_Entity_Recognizer
    - Ontology
    - Prosodic_Analyser
    - Representation-Annotation_Formalism-Guidelines
    - Representation-Annotation_Standard-Best_Practice
    - Signal_Processing-Feature_Extraction
    - Speaker_recogniser
    - Tagger-Parser
    - Terminology
    - Tokenizer
    - Transcriber
    - Word_Sense_Disambiguator

Figure 5: Resources types and other Metadata

Populate the remaining metadata with standard values Each metadata is a specific *Class* which contains the static set of values provided to the authors during the submission stage, see Figure 6;



Class hierarchy: Modality

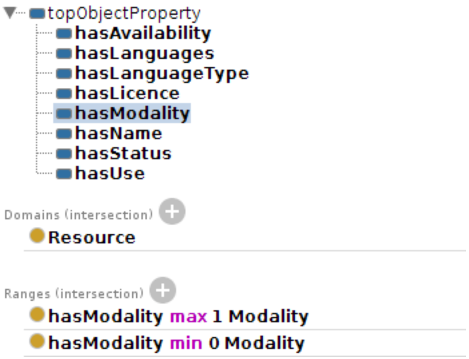
- Thing
  - MetaData
    - Availability
    - CommonValue
    - Languages
    - LanguageType
    - licence
    - Modality
    - Name
    - Status
    - Use
  - Resource

Members list:

- Modality_Independent
- Multimodal/Multimedia
- Sign_Language
- Speech
- Speech-Written
- Written

Figure 6: Metadata: list of individuals

Define properties to link resources to their metadata Each metadata Md_i is connected to the *Resource* through the corresponding $hasMd_i$ property.
Figure 7 reports the definition of the *hasModality* property



- topObjectProperty
  - hasAvailability
  - hasLanguages
  - hasLanguageType
  - hasLicence
  - hasModality
  - hasName
  - hasStatus
  - hasUse

Domains (intersection)

- Resource

Ranges (intersection)

- hasModality max 1 Modality
- hasModality min 0 Modality

Figure 7: hasModality: domain and range

4.1. Gathering data from authors

The process of gathering Language Resources data through the conference submission phase is quite complex since authors have been left free to ride their judgments about the resources they describe. In other words they are neither constrained to pick metadata values from the list we provide, nor to select one of the provided *Types*. They can add *Types* as well as metadata values.
We call the Language Resources they describe *Resource Instances* since they instantiate the abstract *Resource* adding concrete information. In the blueprint-paradigm, these instances are the different buildings extracted from the same blueprint. For the sake of clarity, the comparison is not fully exact, since two different situations can occur:

Authors use provided type and metadata In this case the *Resource Instance* individual is a true instance of the *Resource*, since the author have simply used the data provided and added a new record to the database which is then rendered as an instance, see Figure 8.

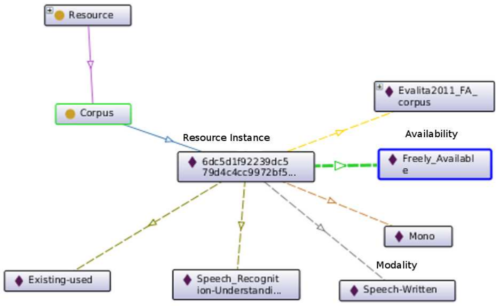


Figure 8: True Instance

Only one or none out of type and metadata is used In this case the *Resource Instance* is both an instance and an extension of the *Resource*. As an instance it instantiates the provided values; as an extension it adds new *Type* and/or metadata, as in Figure 9.

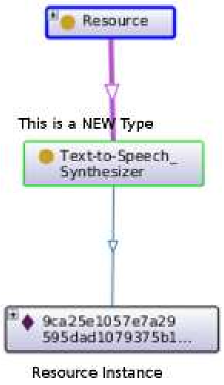


Figure 9: Instance and extension

4.2. The complete picture

The lremap_ri schema collects the resources as they come out from the authors' submissions and imports the submission ontology (lremap_subs) to manage connections to *Papers* and *Authors*.
Figure 10 is the Protégé[8] rendered graph of Figure 3.

[8] http://protege.stanford.edu/.

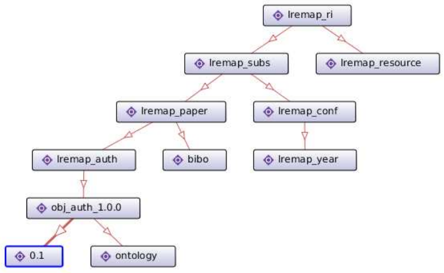


Figure 10: lremap_ri and the set of ontologies

5. Before modelling the LRE Map Ontology: what *is* in the data

In this section we briefly explain what we actually have in the records of the database. If we focus on *name* and on description of the resource authors provide, they contain terms that identify and describe the Language Resources in natural language. Often they contain (or are only) acronyms: for example the "British National Corpus" can be found as "BNC", as "British National Corpus" or as "British National Corpus (BNC)" and as any other possible combination. In (luckily) few case, authors provide shortened forms or abbreviations that are hardily resolved to the correct name.

The noisy data we have represent an issue that we have to solve before providing the data to the community, but (at the same time) they represent also a challenge, an opportunity. The data as they have been natively submitted must be stored somewhere and taken into account for further analysis.

The strategy we have followed to address the previous issue is to carry on a *normalization process* of the values contained in the metadata provided by the authors[9] and to link original values to normalized ones.[10]

ResourceName Normalization of resource names consists in acronym addition and capitalization of the first letter;

ResourceType As explained before, this metadata is essential in classifying the LRs. The normalization process must be accurately carried out: a typical normalization is substituting a provided *Corpora* with the official *Corpus*;

ResourceUse The use of the resource is very important within the LRE Map, since it describes what authors think the resource should be used for. In this case, very often, authors decide to insert a free description instead of using provided values. The normalization is quite heavy since sentences like "disambiguate the correct sense" must be mapped onto the value *Word Sense Disambiguation* and so on;

ResourceLanguage This is most normalized metadata. To uniform the languages we decided to add the three-letter ISO code[11] to the language: "English" → "English (eng)". Data are very noisy since they contain typos [*Chinse* instead of *Chinese*] and errors such as "C++" which clearly is *not* a language in the sense we have in mind.

The normalization process is time-consuming and it must be carefully carried out. In synthesis it complies to the following steps:

Work on a Conference We decide to normalize conference by conference, in order to have small data to work on;

Duplicate the database entries Each LR is copied onto a corresponding resource whose metadata contain only normalized values. In terms of ontologies we create a *lremap_nri* collection of instances;

Creation of a sort-of-template Once normalized, the (normalized) resources are analyzed to extract common features in order to define a sort-of-template for LRs. This template collects common metadata values from the normalized values and creates a grid that can be used to generate new Language Resources.

6. modelling the LRE Map Ontology

In this section we present the strategy we have defined for converting the LRE Map database into RDF.

Language Resources are collected during the paper submission procedures and then gathered into the LRE Map database.

Given these peculiarities, we need a more complicated strategy which allows us to model ontologies for the three types of LRs below:

The Standard LR This is the LR with its original set of metadata, M_r;

The Normalized LR This is the LR whose original set of metadata has been normalized, M_{nr};

The Template LR This is an abstract LR which contains a subset $M_t \subseteq M_r$ of metadata and represents a sort of grid that can be used to fill the metadata of LRs. The subset M_t contains metadata whose value is unique over the whole database.

with additional constraints:

- Not all resources have their template;[12]

[9] Remember that authors can either select one provided value or type the one they like in natural language. The process of normalization we are going to describe is related to the latter data.

[10] The list below is not exhaustive, it simply reports the main metadata that need an accurate investigation during the normalization process. Metadata such as modality contain only the provided values.

[11] The ISO codes for languages are extracted from http://www.iso.org/iso/language_codes taking into account the tables 639-2 and 639-3.

[12] From LRs which appear only once in the database, the template has not been extracted.

- Each standard resource instance is connected to the corresponding normalized instance. From this association is possible to derive the association between M_r and M_{nr}, for example *freely-available* → *Freely Available*.

From Standard, Normalized and Template LRs, we have extracted three distinct ontologies. Such ontologies have the same basic schema and only differ on the number of instances. Figure 11 shows the cooperation among these three LR ontologies.

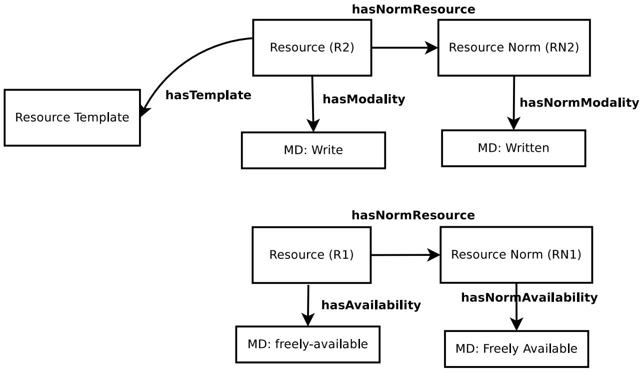


Figure 11: Standard, Normalized and Template LRs

6.1. Modelling the resource ontology

The definition of *Classes* and *Properties* is based on the analysis of the LRE Map database. The main issue to address is to understand whether one metadata can be promoted to be the characterizing axis for the resources, that is which dimension can play the role of the *is_a* relation. Analyzing the database, we recognized that two metadata, *ResourceType* and *ResourceName* can be used as characterizing dimensions. We are aware that, from the "human" perspective, the *name* of the resource is more expressive, but we also understand that the resource can inherit more information from its *type*.

A second crucial aspect in modelling the resource ontology is the definition of the unique identifier. In the database, this identifier depends on the provided *type* and *name* of the specific Language Resource, combined with the submission identifier:

$$Id = \quad F(S1, Name, Type) \tag{2}$$

We have also used this strategy to define identifiers for LR templates and normalized LRs:

$$\begin{aligned} Id_n = &\quad F(S1, Name_n, Type_n) \\ Id_t = &\quad F(S1, Name_t, Type_t) \end{aligned} \tag{3}$$

Types and *names* in assertion 3 are the values of the normalized resource and of its template.

Figure 12 arranges the resources in Figure 2 showing how the three LR ontologies emerge from the data. The same figure presents the relations among the identifiers.

$$\text{S1:} \begin{cases} \text{Resource(R1):} \begin{cases} \text{Name:} & WordNet \\ \text{Type:} & Lexicon \\ \text{Availability:} & Freely \\ & Available \\ \text{Use:} & Summarization \\ \text{Id:} & Id_1 = \\ & F(S1, Name, \\ & Type) \end{cases} \\ \text{Norm(R1\_n):} \begin{cases} \text{Name:} & WordNet \\ \text{Type:} & Lexicon \\ \text{Availability:} & Freely \\ & Available \\ \text{Use:} & Summarization \\ \text{Id:} & Id_{n1} = \\ & F(S1, Name, \\ & Type) \end{cases} \end{cases}$$

$$\text{S2:} \begin{cases} \text{Resource(R2):} \begin{cases} \text{Name:} & WordNet \\ \text{Type:} & Lexicon \\ \text{Availability:} & freely-available \\ \text{Use:} & Knowledge \\ & Discovery \\ \text{Id:} & Id_2 = \\ & F(S2, Name, \\ & Type) \end{cases} \\ \text{Norm(R2\_n):} \begin{cases} \text{Name:} & WordNet \\ \text{Type:} & Lexicon \\ \text{Availability:} & Freely \\ & Available \\ \text{Use:} & Knowledge \\ & Discovery \\ \text{Id:} & Id_{n2} = \\ & F(S2, Name, \\ & Type) \end{cases} \end{cases}$$

$$\text{Resource Template (RT):} \begin{cases} \text{Name:} & WordNet \\ \text{Type:} & Lexicon \\ \text{Availability:} & Freely\ Available \\ \text{Id:} & Id_t = F(Name, \\ & Type) \end{cases}$$

$$Id_1 \text{ hasNormId } Id_{n1} \; Id_1 \text{ hasTempId } Id_t$$
$$Id_2 \text{ hasTempId } Id_t$$

Figure 12: Standard, Normalized and Template LRs and relations among Ids

Each LR instance is identified with its Id:

$$R1 \equiv Id_1 \; R2 \equiv Id_2 \; R1_n \equiv Id_{n1} \; RT \equiv Id_t$$

and can be formalized as follows:[13]

$$is\_a(Id_1, Lexicon) \wedge hasName(Id_1, WordNet)$$
$$\wedge\, hasNormId(Id_1, Id_{n1})$$
$$\wedge\, hasTempId(Id_1, Id_t) \; \ldots$$

We have used Protégé to manually create the ontologies and a set of Python scripts (cf. section 8.) or an ad-hoc conversion of the LRE Map database into RDF; these scripts

[13] This example is for $R1$.

are responsible for extracting the data from the database and create the individuals according to the rules described above.

7. Serialization of LRE Map and (L)LOD

7.1. Data Availability

Table 1 summarizes the URLs where schemas and individuals can be found. In what follows by *mapbase* we mean the URL *http://www.resourcebook.eu/lremap/owl/*.

Object	***URI***
Year	mapbase/lremap_year
Conference	mapbase/lremap_conf
Submission	mapbase/lremap_subs
Author	mapbase/lremap_auth
Paper	mapbase/lremap_paper
Resource	mapbase/lremap_resource
Resource Instance	mapbase/lremap_ri
Normalized Resource Instance	mapbase/lremap_nri
Template Resource Instance	mapbase/lremap_tri

Table 1: LRE Map set of ontologies and URLs

7.2. Serializing the *Resource Instances*

There are a lot of tools designed to help transforming structured data into RDF[14] and providing additional outcomes. For example, *csv2rdf4lod* makes the conversion simple and straightforward, while *sparqlify*[15] adds one SPARQL endpoint which is surely a positive aspect. We know that using tools will save a lot of work but writing conversion from scratch will help us to get more confident with the data, at least in the early stages of the conversion process. So we decided to proceed from scratch using ad-hoc scripts for two main reasons:

- The LRE Map is a complex database: to be more precise, it is at least 3 databases: Authors, Papers and Resources. They are interconnected and need manipulation before a tool such as *csv2rdf4lod* can manage the data;

- We have a lot of codes that communicate with the database and produce triples as reported (Del Grattaet al., 2013; Bartolini et al., 2013). We decided to reuse what we had so that the codes could be improved and made more general.

In this section we focus on a fictional example which clarifies the two different situations in section 4.

As described above, two different situations can arise during the submission phases of the conferences: either the authors select one of the value the submission interface provides or they insert new values. And this is valid for any type of metadata, including the type of resource. The *Resource* schema contains the proposed values as instances,[16] while *Resource Instance* extends *Resource* with the capability of adding new values for metadata as well as new *types* of resources.
The serialization process needs to manage these issues and correctly switch the output.

```
@prefix subs: <mapbase/lremap_subs#> .
@prefix paper: <mapbase/lremap_paper#>
    .
@prefix conf: <mapbase/lremap_conf#> .
@prefix auth: <mapbase/lremap_auth#> .
@prefix ri: <mapbase/lremap_ri#> .
@prefix res: <mapbase/lremap_resource#>
    .

res:Corpus
 a owl:Class ;
 rdfs:subClassOf :Resource .

<ri#6dc5d1f92239dc579d4c4cc9972bf5ec>
 res:hasAvailability res:
    Freely_Available ;
 res:hasLanguage <http://www.lexvo.org/
    page/iso639-3/eng> ;
 res:hasLanguageType res:Mono ;
 res:hasModality res:Speech-Written ;
 res:hasName :resName ;
 res:hasStatus res:Existing-used ;
 res:hasUse res:Speech_Recognition-
    Understanding ;
 ri:hasInnerID ”6
    dc5d1f92239dc579d4c4cc9972bf5ec” ;
 ri:hasSubmission <subs#1000X–xxxx> ;
    a res:Corpus, owl:NamedIndividual .

:Evalita2011_FA_corpus
    a res:Name, owl:NamedIndividual .

.....

.....
```

Figure 13: Resource Instance with type “Corpus”, one of the values proposed.

[14] An exhaustive list of such tools can be found at https://en.wikipedia.org/wiki/Knowledge_extraction #Survey_of_Methods_.2F_Tools.

[15] http://sparqlify.org/

[16] Essentially the provided types of the resource are classes that identify the resources, while other metadata are individuals: “Corpus” is-a *Resource*, while “freely-available” is an individual of the class *Availability*.

```
@prefix subs: <mapbase/lremap_subs#> .
@prefix paper: <mapbase/lremap_paper#>
    .
@prefix conf: <mapbase/lremap_conf#> .
@prefix auth: <mapbase/lremap_auth#> .
@prefix ri: <mapbase/lremap_ri#> .
@prefix res: <mapbase/lremap_resource#>
     .

:Text-to-Speech_Synthesizer
    a owl:Class ;
    rdfs:subClassOf res:Resource .

<ri#9ca25e1057e7a29595dad1079375b100>
  hasSubmission <subs#1000X–xxxx> ;
  a :Text-to-Speech_Synthesizer , owl:
      NamedIndividual .
.....

.....
```

Figure 14: Resource Instance with type "Text-to-Speech_Synthesizer", a new resource type.

The type of *Resource Instance* individual identified by `6dc5d1f...` in Figure 13 is a "Corpus", which is a value provided by the submission interface. The serialization tools set this instance as `a res:Corpus`. On the contrary, the instance identified by `9ca25e1...` in figure 14 has "Text-to-Speech_Synthesizer" as type. This value is a new one, thus it needs to be firstly defined as a *Resource* and then assigned to the instance (individual). The serialization tools define this new value as a `rdfs:subClassOf res:Resource`, then assert that this specific *Resource Instance* individual is `a :Text-to-Speech_Synthesizer`.

8. Conclusion and Future Work

In this paper we have presented the strategy to convert the LRE Map database into RDF. We used Protégé to create the schemas and a set of ad-hoc scripts to populate the schemas with proper instances. We preferred this solution instead of using available serializing tools[17] because of the peculiar nature of the data in the LRE Map database and in order to maintain a full control over the serializing process, in terms of data coherence. However, the csv2rdf4lod tool[18] has been used to create a benchmark for the future serialization.

The work will be completed by the identification of the Id_t with the ISLRN (Choukri et al., 2012). In such a way, instances of the ontology for LR templates will be of the following form:

`mapbase/lremap_ri#islrn_1`

9. References

Bartolini, R., Del Gratta, R., and Frontini, F. (2013). Towards the establishment of a linguistic linked data network for Italian. In *Proceedings of the 2nd Workshop On Linked Data in Linguistic, Collocated with the 6th International Conference on Generative Approaches to the Lexicon*, Pisa, Italy, September.

Del Gratta, R., Frontini, F., Khan, F., and Monachini, M. (2013). Converting the PAROLE SIMPLE Clips Lexicon into RDF with lemon. *Semantic Web Journal (Submitted)*.

Calzolari, N., Soria, C., Gratta, R. D., Goggi, S., Quochi, V., Russo, I., Choukri, K., Mariani, J., and Piperidis, S. (2010). The LREC Map of Language Resources and Technologies. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 949–956, Valletta, Malta, May. European Language Resources Association (ELRA).

Calzolari, N., Gratta, R. D., Francopoulo, G., Mariani, J., Rubino, F., Russo, I., and Soria, C. (2012). The LRE Map. Harmonising Community Descriptions of Resources. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 1084–1089, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Chiarcos, C., Hellmann, S., and Nordhoff, S. (2011). Towards a Linguistic Linked Open Data cloud: The Open Linguistics Working Group. *TAL*, 52(3):245–275.

Chiarcos, C., Hellmann, S., and Nordhoff, S. (2012). *Linked Data in Linguistics*. Springer.

Choukri, K., Arranz, V., Hamon, O., and Park, J. (2012). Using the International Standard Language Resource Number: Practical and Technical Aspects. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 50–54, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Lezcano, L., Sanchez, S., and Roa-Valverde, A. J. (2013). A survey on the exchange of linguistic resources: Publishing linguistic linked open data on the Web. *Program: electronic library and information systems*, 47(3):3–3.

[17] https://en.wikipedia.org/wiki/Knowledge_extraction.

[18] http://logd.tw.rpi.edu/technology/csv2rdf4lod