RELISH LMF:
Unlocking the Full Power of the Lexical Markup Framework

Menzo Windhouwer[1], Justin Petro[2], Shakila Shayan[3]
[1] The Language Archive, DANS
Anna van Saksenlaan 51, 2593 HW The Hague, The Netherlands
[2] LINGUIST List - Eastern Michigan University
2000 Huron River Drive, Suite 104 Ypsilanti, MI 48197 United States
[3] The Language Archive, MPI for Psycholinguistics
Wundtlaan 1, 6525 XD Nijmegen, The Netherlands
Menzo.Windhouwer@dans.knaw.nl, justin@linguistlist.org, Shakila.Shayan@mpi.nl

Abstract

The Lexical Markup Framework (ISO 24613:2008) provides a core class diagram and various extensions as the basis for constructing lexical resources. Unfortunately the informative Document Type Definition provided by the standard and other available LMF serializations lack support for many of the powerful features of the model. This paper describes RELISH LMF, which unlocks the full power of the LMF model by providing a set of extensible modern schema modules. As use cases RELISH LL LMF and support by LEXUS, an online lexicon tool, are described.

Keywords: lexical resources, Lexical Markup Framework, interoperability

1. Introduction

In 2008 ISO[1] Technical Committee 37 *Terminology and other language and content resources* released ISO 24613 Lexical Markup Framework, abbreviated as LMF (ISO 24613, 2008)[2]. To create an LMF compliant lexicon various parts have to be combined. The basis is formed by the LMF core class diagram, which is specified in UML (Rumbaugh, Jacobson, & Booch, 2004). The standard then provides 8 extensions to the core model, and also states that "additional extensions may be developed over time". From these at least one extension specifying a non-abstract subclass of the mandatory, but abstract, class Form has to be selected. The last step is to adorn the class model with data categories selected in the ISOcat Data Category Registry (DCR).[3]

A lexicon tool which supports LMF can instantiate this model internally. But one of the goals of LMF is to foster exchange of lexical resources. This requires a serialization of instances of the class model, which preferably can also be inspected by humans. In general exchange formats like this use XML (W3C, 2008). This is also true for LMF. The standard also provides a Document Type Definition (DTD), a specific kind of schema for XML documents, in the informative Annex R. The introduction of that annex states "A user can decide to define another DTD or schema to implement LMF in it. It is also possible to use the XML structures that are defined in the Feature Structure Representation standard (i.e. ISO 24610-1)". This paper describes such an alternative schema, i.e., the RELISH LMF schema[4]. But before describing how this schema supports and strengthens the power of the LMF class model other existing LMF schema's, including the DTD from Annex R, are discussed.

2. Existing LMF Serializations

The informative DTD of Annex R captures the whole LMF class model, i.e., including its 8 extensions. It also supports a basic representation of a feature structure. However, there are a number of drawbacks to this DTD.

It is one monolithic schema and thus does not allow one to select only the extensions required by a specific lexicon or to define new extensions.

The DTD also inherits a synchronization problem between the LMF standard and the one implemented by the ISO DCR (ISO 12620, 2009). When the LMF standard was in its final stages it was not yet clear how to refer to data categories stored in the DCR. LMF basically assumes that the name of a data category is unique. However, this is not the case in the DCR. As this registry has to cater for different domains names can be ambiguous and so another identifier was introduced: a unique and stable URI. An LMF schema should embed these URIs to clearly identify which data category is meant. The DTD lacks facilities for this.

Last, but not least, here is an impedance mismatch between XML DTDs and UML class models, which means that certain constraints expressed in the class diagram cannot be expressed in the DTD and hence the DTD is more lax than the standardized model, e.g., cardinality constraints like a List of Components should refer to at least two Components are not validated by the DTD.

In the KYOTO project a Wordnet LMF DTD was developed (Vossen, Soria, & Monachini, 2013). This schema is based on the informative DTD, so it inherits the

[1] There is a lookup table for abbreviations at the end of the paper.
[2] See also http://www.lexicalmarkupframework.org/
[3] http://www.isocat.org/
[4] http://tla.mpi.nl/relish/lmf/

problems of that schema, but it also adds additional problems. While the informative DTD allowed one to use any data category in its simple feature structure representation, the KYOTO Wordnet LMF DTD has a fixed set of them serialized as XML attributes. Also implements only a specific selection of LMF extensions is supported. These problems clearly indicate that this schema is project specific and not meant to cover LMF as a whole.

The Dutch Cornetto LMF RDF project (Cornetto-LMF-RDF project, 2014) follows basically the same approach as KYOTO, i.e., supports a project specific set of features and extensions. But next to a DTD also a W3C XML Schema (W3C, 2014) is available.

The Text Encoding Initiative (TEI) provides a very flexible XML environment for marking up digital text. This includes a dictionaries module to be used by lexical resources (TEI Consortium, 2014). It has been proposed by (Romary, 2013) to converge the TEI and LMF initiatives, but at the time of writing this has not happened yet.

Next to these XML serializations other serialization formats are possible, e.g., into the Resource Description Framework (RDF) (W3C, 2004). Lemon is such an RDF representation. It claims to be "highly LMF compliant", which already indicates it deviates from the LMF standard (McCrae, et al.). Deviations include the use of different terminology, dropping of classes and different modelling of senses and semantics. The predecessor of Lemon LexInfo's LMF ontology (Buitelaar, Cimiano, Haase, & Sintek, 2009) was directly based on the LMF UML model.

In the next section the RELISH LMF serialization is described, which does provide full support for the complete LMF model.

3. The RELISH LMF Serialization

The RELISH project [5] promoted language-oriented research by addressing a two-pronged problem: (1) the lack of harmonization between digital standards for lexical information in Europe and America, and (2) the lack of interoperability among existing lexicons of endangered languages (Aristar-Dry, Drude, Gippert, Nevskaya, & Windhouwer, 2012). The RELISH LMF serialization was one of the results of this project (Windhouwer, Petro, Nevskaya, Drude, Aristar-Dry, & Gippert, 2013). Instead of a DTD RELISH LMF uses two more modern XML validation languages[6]. RELAX NG (ISO/IEC 19757-2, 2008) is a relatively simple XML schema language, which allows building modular schemas. And Schematron (ISO/IEC 19757-3, 2006) is a rule based validation language that allows checking the (non) occurrence of certain patterns in an XML document. These two languages can interact nicely especially given that Schematron rules can be embedded in RELAX NG modules.

RELISH LMF consists of 12 RELAX NG modules. Of these two are mandatory:

1. A generic module declares some common structures, e.g., mandatory or optional ID attributes, that are reused by other modules;
2. The RELISH LMF core module, which specifies a serialization of the core LMF UML module.

Next there are 8 modules corresponding to the 8 extensions specified in the LMF standard:

3. The morphology extension;
4. The machine readable dictionary extension;
5. The NLP syntax extension;
6. The NLP semantics extension;
7. The NLP multilingual notations extension;
8. The NLP morphological patterns extension;
9. The NLP multiword expression patterns extension;
10. The constraint expression extension.

A coherent selection of these modules can be made. The RELAX NG and Schematron validation of the instances will indicate the coherency of the selection.

The introduction of Annex R showed that, next to the one used in the informative DTD, there are other feature structure representations possible, i.e., the more powerful and standardized TEI/ISO representation. None of the discussed LMF serializations support this advanced representation. RELISH LMF does by allowing the user to select one of the following modules:

11. The simple feature structure representation of the informative DTD, but extended with support for references into the DCR (Windhouwer & Wright, Referencing ISOcat data categories, 2010);
12. The full power of the TEI/ISO feature structure representation (FSR, (ISO 24610-1, 2006)) including feature system declarations (FSD, (ISO 24610-2, 2011)) which declare the allowed structure of a FSR.

Although RELISH LMF supports both the simple and the TEI/ISO FSR, usage of the latter is advised due to its potential for validation using FSDs (see also the end of this section).

Next to these modules based on the LMF standard it is also possible to create one's own extensions and restrictions. To make a proper distinction between XML elements' coming from the standard or lexicon specific extensions, RELISH LMF uses a namespace (W3C, 2009). User specific extensions should be created in their own namespace.

The following is an example of a simple LMF compliant lexicon schema using RELISH LMF (lexicon.rng):

```
<grammar
  xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:lmf="http://www.lexicalmarkupframework.org/"
  xmlns:dcr="http://www.isocat.org/ns/dcr"
>
```

[5] http://tla.mpi.nl/relish/

[6] In the RELISH project also TEI was considered as a pivot format, but was considered too unconstrained (Aristar-Dry, Petro, Miller, Wicks, & Aristar, 2011). Section 4 describes the RELISH LL LMF pivot format used in RELISH to exchange lexica. RELISH LMF is a super set of this encompassing the full LMF class model.

```
<!--
  1. always
-->
<include href="RELISH-LMF-common.rng"/>
<include href="RELISH-LMF-core.rng"/>

<!--
  2. select LMF extensions
-->
<include href="RELISH-LMF-morphology.rng"/>

<!--
  3. choose a feature structure representation
-->
<include href="RELISH-LMF-fs-lmf.rng"/>

<!--
  4. optionally: add your own extensions
-->

</grammar>
```

A valid instance of this schema can look as follows:

```
<?xml-model
  href="lexicon.rng" type="application/xml"
  schematypens="http://relaxng.org/ns/structure/1.0"
?>
<?xml-model
  href="lexicon.rng" type="application/xml"
  schematypens="http://purl.oclc.org/dsdl/schematron"
?>
<LexicalResource lmfVersion="ISO 24613:2008"
  xmlns="http://www.lexicalmarkupframework.org/"
  xmlns:dcr="http://www.isocat.org/ns/dcr"
>
  <GlobalInformation>
    <feat
      att="languageCoding"
      val="ISO 639-3"
    />
  </GlobalInformation>
  <Lexicon xml:lang="en">
    <feat
      att="language"
      val="eng"/>
    <LexicalEntry xml:id="le0001">
      <feat
        att="partOfSpeech"
        val="commonNoun"
      />
      <Lemma type="Form">
        <feat
          att="writtenForm"
          val="clergyman"
        />
      </Lemma>
    </LexicalEntry>
  </Lexicon>
</LexicalResource>
```

The bold sections indicate differences in the XML representation with regard to the informative DTD. The two xml-model processing instructions (W3C, 2011) at the top trigger the validation using RELAX NG and Schematron. On the root node the RELISH LMF namespace is declared as the default. Also the Data Category Reference namespace is declared (see below). The lmfVersion attribute will allow the RELISH LMF serialization to deal with future new releases of LMF. The remainder of the document showcase the use of common attributes in the xml namespace, i.e., xml:lang and xml:id. The type attribute on Lemma allows a Schematron rule to validate that each Lexical Entry indeed contains a sub class of the abstract Form class.
This example did not contain any links to data categories in ISOcat. These can be added using the dcr:datcat and dcr:valueDatcat attributes. For example:

```
<feat
  att="partOfSpeech"
  dcr:datcat="http://www.isocat.org/datcat/DC-1345"
  val="commonNoun"
dcr:valueDatcat="http://www.isocat.org/datcat/DC-1256"
/>
```

Notice here each instance needs to be annotated, which would be highly redundant in a real world lexical resource. The ISO/TEI FSR combined with FSDs supports a much leaner way. Only the FSD needs to be annotated and one FSD functions as a schema for many FSRs. The FSD and example FSR for the LexicalEntry could look as follows:

```
<LexicalResource lmfVersion="ISO 24613:2008"
 xmlns="http://www.lexicalmarkupframework.org/"
 xmlns:tei="http://www.tei-c.org/ns/1.0"
 xmlns:dcr="http://www.isocat.org/ns/dcr"
>
 <tei:fsdDecl>
  …
  <tei:fsDecl type="LexicalEntry">
   <tei:fDecl
    name="partOfSpeech"
    dcr:datcat="http://www.isocat.org/datcat/DC-1345"
   >
    <tei:vRange>
     <tei:vAlt>
      …
      <tei:symbol
       value="commonNoun"
dcr:valueDatcat="http://www.isocat.org/datcat/DC-1256"/>
     </tei:vAlt>
    </tei:vRange>
   </tei:fDecl>
  </tei:fsDecl>
  …
 </tei:fsdDecl>
```

```
 …
 <Lexicon>
  …
  <LexicalEntry>
   <tei:f
    name="partOfSpeech"
   >
    <tei:symbol
     value="commonNoun"
    />
   </tei:f>
   …
  </LexicalEntry>
  …
 </Lexicon>
</LexicalResource>
```

The FSD is bound to the XML serialization of a LMF class using its QName. It implies that the current class XML serialization as an XML element is equivalent to a feature structure, i.e., <lmf:LexicalEntry> is equivalent to <tei:fs type=”lmf:LexicalEntry”>. The relationships between class instances are then represented by complex feature values, i.e., nesting of feature structures.

Although RELAX NG and Schematron are not as widely used as W3C XML Schema, validation is well supported. The oXygen XML editor[7], for example, will validate an RELISH LMF instance based on the xml-model processing instructions. Various options are available for command line tools and libraries (Makoto, 2014). However, unfortunately not all options will support also the validation of the embedded Schematron rules. But (Jelliffe, 2010) provides a four-stage XSLT pipeline to extract the rules and convert them into an XSLT 2.0 (W3C, 2007) stylesheet. This stylesheet can then be used to generate a report of the compliance of a RELISH LMF instance with regard to the Schematron rules.

When a RELISH LMF schema uses the TEI/ISO FSR it is possible to provide also FSDs to describe the desired structure of the feature structures. However, to the authors knowledge there is currently no FSD-based TEI/ISO FSR validator.

4. The RELISH LL LMF Use Case

One of the aims of the RELISH project was to make lexicons of endangered languages interoperable. A pivot format, RELISH LL LMF, based on LMF and TEI/ISO FSR was designed. The project only needed a small part of LMF, but the potential of the approach was realised and resulted in RELISH LMF as described in the previous section. RELISH LL LMF is now a specific use case of RELISH LMF, i.e., it is a RELAX NG schema that includes existing RELISH LMF modules and tweaks them to its specific needs. This section describes these needs.

RELISH LL LMF was developed in part to make a serialization of LMF that is interoperable with the Lexicon Interchange Format (LIFT, (Hosken, 2006)), an XML standard developed by the Summer Institute of Linguistics (SIL) and employed in the Lexicon Enhancement via the Gold Ontology (LEGO) Project[8], developed by the Institute for Language Information and Technology at Eastern Michigan University. Due to the structural differences between LIFT and LMF, three main considerations were made during the development of RELISH LL LMF. Firstly, the TEI/ISO FSR was used to encode information normally encoded by the general element <note> in LIFT.

Secondly, the TEI/ISO FSR was also employed within the <Sense> element to encode grammatical information as opposed to using part of speech, grammatical gender, and grammatical tense features of LMF. This was done to mirror the encoding of grammatical information in LIFT, where all such data is encoded as a feature of the <grammatical-info> element, itself a child of <sense> (Hosken, 2006).

Finally, example sentences were encoded using <TextRepresentation> as the head of TEI/ISO FSR. This encoding was used to mirror the LIFT encoding of example sentences and their translations within <example> and its daughter element <translation>, respectively.

A snippet of the schema shows that in this use case the power of RELAX NG is used to restrict the LMF classes provided by the RELISH LMF modules to align them with LIFT as supported by RELISH LL LMF:

```
<grammar
  xmlns="http://relaxng.org/ns/structure/1.0"
>
 …
 <include href="RELISH-LMF-morphology.rng">
  <!-- override all LMF classes not
       supported by RELISH-LL-LMF -->
  <define name="relish.lmf.WordForm">
   <empty/>
  </define>
  <define name="relish.lmf.Stem">
   <empty/>
  </define>
  <define name="relish.lmf.ListOfComponents">
   <empty/>
  </define>
  <define name="relish.lmf.Component">
   <empty/>
  </define>
 </include>
 …
</grammar>
```

Additionally there are some Schematron rules to check for required features. These rules can potentially be replaced by FSDs when a TEI/ISO FSR validator becomes available.

[7] http://www.oxygenxml.com/

[8] http://lego.linguistlist.org/

5. RELISH LMF Support in LEXUS 3.1

LEXUS[9] is a powerful online lexicon tool with a strong focus on visualization and multimedia (Shayan, Moreira, Windhouwer, König, & Drude, 2013). LMF support has been a long time goal of LEXUS. RELISH LMF and RELISH LL LMF are now supported by LEXUS 3.1[10] using a flexible import and export facility. As LEXUS allows arbitrary changes to the structure of a lexical entry users might unintentionally break LMF conformance. The internal data model management of LEXUS has been extended to detect this and warn users when this might happen.

6. Conclusions and Future Work

This paper described the RELISH LMF serialization, which due to the use of modern, extensible schema modules unlocks the full power of the Lexical Markup Framework. Future work includes transformations from and, when possible, to the other existing LMF serializations. Also validation of feature structures when feature system declarations are available will be valuable, and could be a contribution to an even wider community.

Abbreviations

DCR	Data Category Registry
DTD	Document Type Definition
FSD	Feature System Declaration
FSR	Feature Structure Representation
GOLD	General Ontology for Linguistic Description
ISO	International Organisation for Standardization
KYOTO	Knowledge Yielding Ontologies for Transition-based Organization
LEGO	Lexicon Enhancement via the GOLD Ontology
LIFT	Lexicon Interchange Format
LL	Linguist List
LMF	Lexical Markup Framework
MPI	Max Planck Institute
NLP	Natural Language Processing
QName	Qualifier Name
RDF	Resource Description Framework
RELAX NG	Regular Language for XML Next Generation
RELISH	Rendering Endangered Languages Lexicons Interoperable Through Standards Harmonization
SIL	Summer Institute of Linguistics
TEI	Text Encoding Initiative
TLA	The Language Archive
UML	Unified Modelling Language
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

[9] http://tla.mpi.nl/tools/tla-tools/lexus/

[10] At time of writing LEXUS 3.1 was not yet released into production.

References

Aristar-Dry, H., Drude, S., Gippert, J., Nevskaya, I., & Windhouwer, M. (2012). Rendering Endangered Lexicons Interoperable through Standards Harmonization: the RELISH project. *Eight International Conference on Language Resources and Evaluation.* Istanbul: European Language Resources Association.

Aristar-Dry, H., Petro, J., Miller, B., Wicks, E., & Aristar, A. (2011, October 12). TEI and the LEGO lexicons. *Tightening the representation of lexical data, a TEI perspective*. Wurzburg, Germany.

Buitelaar, P., Cimiano, P., Haase, P., & Sintek, M. (2009). Towards Linguistically Grounded Ontologies. *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications* (pp. 111-125). Heraklion: Springer.

Cornetto-LMF-RDF project. (2014, February 10). *Cornetto LMF RDF*. Retrieved March 20, 2014 from http://cornetto.inl.nl/

Hosken, M. (2006). *Lexicon Interchange Format — A Description.* Dallas: Summer Institute for Linguistics.

ISO 12620. (2009). *Specification of data categories and management of a Data Category Registry for language resources.* Geneve: International Organization for Standardization.

ISO 24610-1. (2006). *Language resource management -- Feature structures -- Part 1: Feature structure representation.* Geneve: International Organization for Standardization.

ISO 24610-2. (2011). *Language resource management -- Feature structures -- Part 2: Feature system declaration.* Geneve: International Organization for Standardization.

ISO 24613. (2008). *Lexical markup framework (LMF).* Geneve: International Organization for Standardization.

ISO/IEC 19757-2. (2008). *Information technology -- Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG.* Geneve: International Organization for Standardization.

ISO/IEC 19757-3. (2006). *Information technology -- Document Schema Definition Languages (DSDL) -- Part 3: Rule-based validation -- Schematron.* Geneve: International Organization for Standardization.

Jelliffe, R. (2010). *schematron*. Retrieved March 12, 2014 from http://www.schematron.com/

Makoto, M. (2014). *RELAX NG home page*. Retrieved March 12, 2014 from http://relaxng.org/

McCrae, J., Aguado-de-Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Pérez, A. G., et al. (n.d.). *The lemon cookbook.*

Romary, L. (2013). *TEI and LMF crosswalks.* arXiv.org.

Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The Unified Modelling Language Reference Manual.* Addison-Wesley.

Shayan, S., Moreira, A., Windhouwer, M., König, A., & Drude, S. (2013). LEXUS 3 - a collaborative environment for multimedia lexica. *Digital Humanities.* Lincoln.

TEI Consortium. (2014). 9. Dictionaries. In *TEI P5: Guidelines for Electronic Text Encoding and Interchange. 2.6.0. January 20, 2014.* (pp. 156-294). TEI Consortium.

Vossen, P., Soria, C., & Monachini, M. (2013). Wordnet-LMF: A Standard Representation for Multilingual Wordnets. In G. Francopoulo, *LMF - Lexical Markup Framework* (pp. 51 - 66). ISTE Ltd: London.

W3C. (2004, February 10). *RDF Primer*. Retrieved October 7, 2013 from http://www.w3.org/TR/rdf-primer/

W3C. (2007). *XSL Transformations (XSLT) Version 2.0.* W3C.

W3C. (2008, November 26). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. Retrieved October 7, 2013 from http://www.w3.org/TR/2008/REC-xml-20081126/

W3C. (2009, December 8). *Namespaces in XML 1.0 (Third Edition)*. Retrieved October 7, 2013 from http://www.w3.org/TR/REC-xml-names/

W3C. (2011). 3 The xml-model processing instruction. In *Associating Schemas with XML documents 1.0 (Second Edition).* W3C.

W3C. (2014). *XML Schema*. Retrieved March 12, 2014 from http://www.w3.org/standards/xml/schema

Windhouwer, M., & Wright, S. E. (2010). Referencing ISOcat data categories. *LREC 2010 LRT standards workshop.* Malta: European Language Resources Association.

Windhouwer, M., Petro, J., Nevskaya, I., Drude, S., Aristar-Dry, H., & Gippert, J. (2013). Creating a serialization of LMF: the experience of the RELISH project. In G. Francopoulo, *LMF - Lexical Markup Framework* (pp. 215 - 225). Londen: ISTE Ltd.