Lexical Markup Framework:
ISO Standard for Semantic Information in NLP Lexicons

Gil Francopoulo[1], Nuria Bel[2], Monte George[3], Nicoletta Calzolari[4], Monica Monachini[5], Mandy Pet[6], Claudia Soria[7]

[1]INRIA-Loria: gil.francopoulo@wanadoo.fr

[2] UPF: nuria.bel@upf.edu

[3] ANSI: dracalpha@earthlink.net

[4]CNR-ILC: glottolo@ilc.cnr.it

[5] CNR-ILC: monica.monachini@ilc.cnr.it

[6] MITRE: mpet@mitre.org

[7]CNR-ILC: claudia.soria@ilc.cnr.it

Abstract

Lexical Markup Framework (LMF) is a model that provides a common standardized framework for Natural Language Processing (NLP) lexicons. The goals of LMF are to provide a common model for the creation and use of such lexical resources, to manage the exchange of data between and among these resources, and to enable the merging of a large number of individual resources to form extensive global electronic resources.

Introduction

In the framework of the ISO Technical Committee 37 and Subcommittee 4 (TC37/SC4) dedicated to resources for NLP, a set of standards for linguistic resources are currently elaborated.
A two level organization has been devised to form a coherent family of standards with the following simple rules:

- high level specifications provide structural classes that are adorned by the standardized attribute names and constants
- low level specifications provide standardized attribute names and constants

High level specifications are those that deal with word segmentation (ISO 24614), annotations (ISO 24611, 24612[1] and 24615), feature structures (ISO 24610), and lexicons (ISO 24613) [Francopoulo], this latest one being the focus of the current paper.
Low level specifications dedicated to constants are for data categories (revision of ISO 12620), language codes[2] (ISO 639 or IETF BCP-47), script codes (ISO 15924), country codes (ISO 3166), dates (ISO 8601) and Unicode (ISO 10646). It should be noted that most low level specifications are existing stable standards that are taken from outside of ISO-TC37.

1 Key standards used in the normative document

Other key standards used or referenced in LMF are Unified Modeling Language (UML) [Rumbaugh] and XML.
UML is a general-purpose visual modeling language that is used to specify, visualize, construct and document data structures. The modeling language is intended to unify past experience about modeling techniques and to incorporate current software best practices into a coherent approach.

---

[1] See Nancy Ide presentation in this conference

[2] See also, the two presentations from Lee Gillam and Felix Sasaki in this conference

UML has been chosen for the following reasons:

- UML is the 'de facto' standard for modeling in the Industry. That means that a lot of professionals are able to understand the specifications.
- UML is well defined and documented;
- the use of diagrams is very efficient when a model needs to be presented and negociated;
- UML allows designers (and readers) to partition large models into workable pieces by means of UML packages;
- Various powerful UML tools are available now in order to ease the design process.

UML captures information about the static structure and dynamic behavior of a system, but in LMF, we restrict ourselves to the static aspect.
We also provide informative examples of content markup using another key standard, XML, although XML is just one way of expressing a LMF model and an XML DTD is given in an annex of the LMF document. But XML is not used during the modeling process because it is not suited for that.
In other terms, we use UML to design the specifications and produce XML from UML, manually afterwards.

3 Structure of the LMF model

As said in section 2, LMF is a high level specification for lexical resources. In order to allow a good modularity, the model is comprised of two types of components:

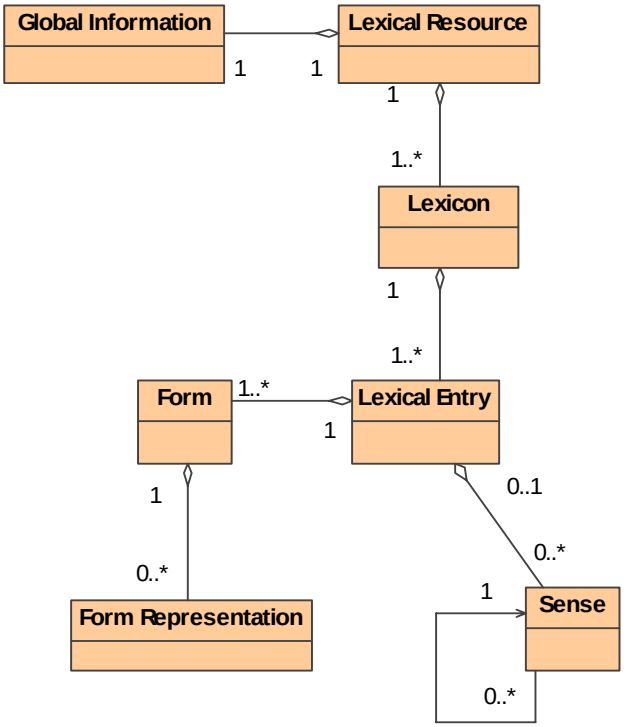
- The core package that is a structural skeleton to represent the basic hierarchy of information in a lexicon.
- Extensions to the core package under the form of UML packages. Each package reuses the core classes in conjunction with additional classes required for the description of the contents of a specific type of lexical resource. There are packages for the description in extension of the morphology of a language, for Machine Readable Dictionary (MRD), for syntactic structures, for semantic descriptions, for multilingual notations, for paradigm classes, for multi-word expression patterns and for constraint expression.

It is important to understand that LMF defines the structure of the lexicon. More precisely, LMF defines class names, class usage, class relations, and package membership by means of English text and UML diagrams. This specification goes with some guidelines and a series of examples, but it is important to highlight that attribute-value pairs like /grammatical gender/ and /feminine/ are not defined within LMF. They are to be taken from the ISO Data Category Registry as specified by ISO-12620.

2 Core package

The Core package is a structural skeleton whose root is *Lexical Resource* class. There is one and only one *Lexical Resource* instance: a singleton in Design Pattern terminology [Gamma].
A *Lexicon* is a container for the words of a given language. *Lexical Entry* is a node that allows the connection between a form and a sense. As a first approximation, a *Lexical Entry* instance is a word. *Form* is a class representing the way a word is spoken and/or written. A *Form* instance may be adorned by a set of attribute-value pairs but for more complex situations, a *Form* instance may be associated with different *Form Representation* instances, for example, when the language has various ways to express written forms, like in Chineese.
In order to express the situations where a word may have different meanings, a *Lexical Entry* instance may be linked to one or several *Senses* instances.

The Core package is defined with the following UML class diagram:



4 Semantic package

One of these extensions is for describing semantic information linked to the core class, *Sense*. The purpose is to describe one sense and its relations with other senses belonging to the same language.

The *Sense* class is associated with the *Lexical Entry* element and cannot be shared by two different entries. *SynSet* links synonymous *Sense* instances. *Semantic Predicate* describes an abstract meaning together with its association with the *Semantic Argument* class.

Semantic descriptions may be mapped to syntactic representations. More precisely, every *Semantic Argument* instance may be mapped to a syntactic argument of a subcategorization frame as defined in the LMF package for Syntax.

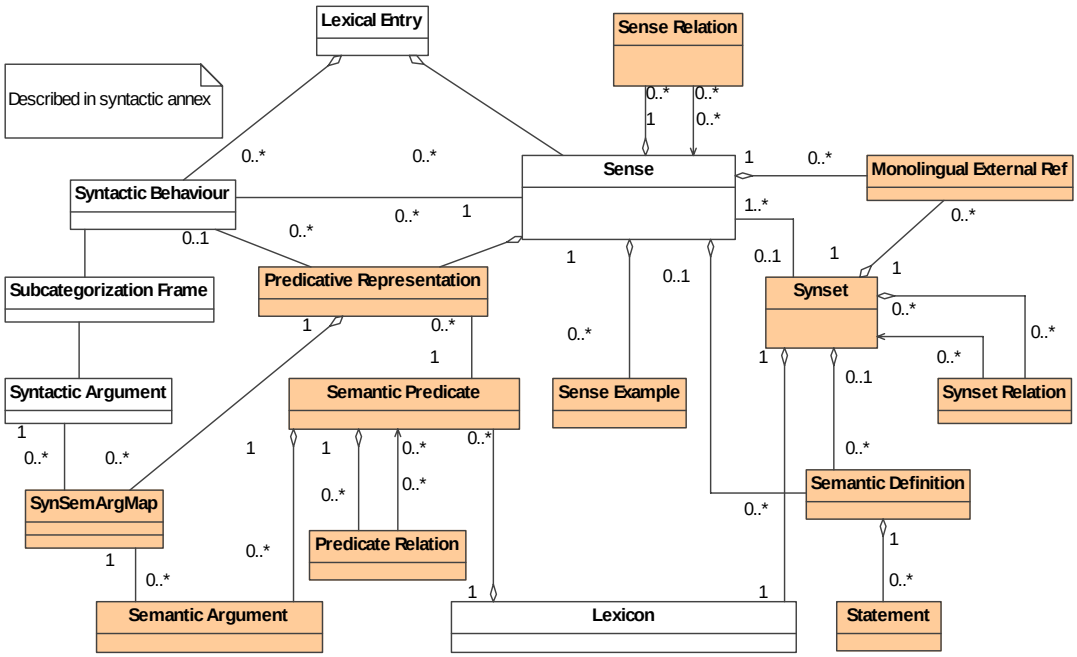
***Sense* node is the key element.** It is not possible to describe *Synset* or *Predicate* instances without any *Sense* instance.

But there is no exclusive usage of these mechanisms. For instance, a lexicon manager may decide to use *Predicate* instances for verbs and predicative nouns and *Synset* instances for other meanings. But the LMF specification does not impose such strict guidelines. The document proposes a formal model and such decisions are left to the lexicon manager.

For a complete description, please refer to LMF document[3].

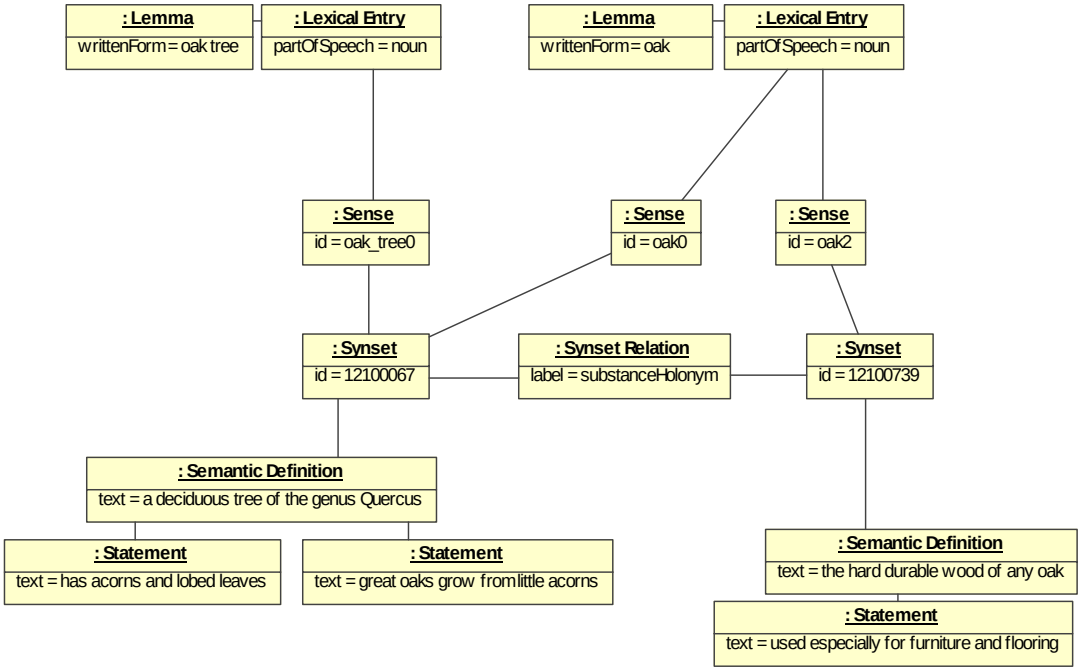
Semantic package is defined with the following UML class diagram:

[3] LMF rev-13 http://lirics.loria.fr/documents.html



5 Example of semantic representation

LMF can be used as a model for new or existing lexicon designs. The following instance diagram shows an example taken from WordNet version 2.1. This example presents two Synset instances: one for oak, the tree and one for oak, the wood. Each WordNet's lex_id is used to identify a *Sense* instance. Each gloss is split into a *SemanticDefinition* instance and possibly several *Statement* instances. The two *Synset* instances are linked by a *SynsetRelation* instance.



The same data can be expressed by the following XML fragment:

```
<LexicalEntry>
   <DC att="partOfSpeech" val="noun"/>
   <Lemma>
      <DC att="writtenForm" val="oak tree"/>
   </Lemma>
   <Sense id="oak_tree0" synset="12100067"/>
</LexicalEntry>
<LexicalEntry>
   <DC att="partOfSpeech" val="noun"/>
   <Lemma>
      <DC att=writtenForm" val="oak"/>
   </Lemma>
   <Sense id="oak0" synset="12100067"/>
   <Sense id="oak2" synset="12100739"/>
</LexicalEntry>
<Synset id="12100067">
   <SemanticDefinition>
      <DC att="text" val="a deciduous tree of the genus Quercus"/>
      <Statement>
         <DC att="text" val="has acorns and lobed leaves"/>
      </Statement>
      <Statement>
         <DC att="text" val="great oaks grow from little acorns"/>
      </Statement>
   </SemanticDefinition>
   <SynsetRelation targets="12100739"
      <DC att="label" val="substanceHolonym"/>
   </SynsetRelation>
</Synset>
<Synset id="12100739">
   <SemanticDefinition>
      <DC att="text" val="the hard durable wood of any oak"/>
      <Statement>
         <DC att="text" val="used especially for furniture and flooring"/>
      </Statement>
   </SemanticDefinition>
</Synset>
```

6 Multilingual notations package

A separate package is used for multilingual notations.

The simplest configuration is the bilingual lexicon where a single link is used to represent the equivalent of a given sense from one language into another, but actual practice reveals at least five more complex configurations:

Diversification and neutralization: in certain circumstances, simple one-to-one mapping between apparent equivalents in two or more languages does not work very well because the conceptual scope represented by words and expressions in the different languages is frequently not the same.

Number of links: although the strategy of one-to-one equivalence is viable for two languages, it becomes untenable for a more extensive number of languages because the number of links explodes to unmanageable proportions.

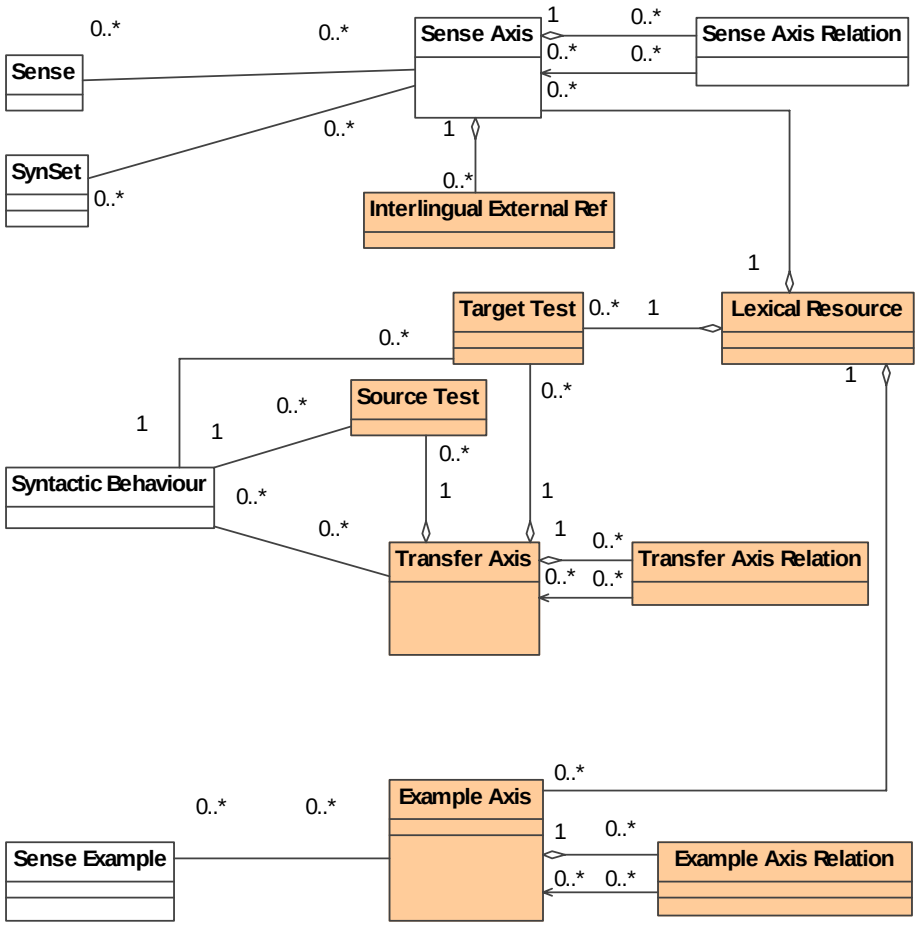
Transfer or interlingual pivot: NLP-oriented translation is based on two approaches: the use of an *interlingual pivot*, which operates on the basis of semantic analysis and *transfer*, which is

based on machine parsing of source text syntax. The pivot approach is implemented via the *Sense Axis* class, and the transfer approach via the *Transfer Axis* class.

Representation of similar languages: very closely related languages that share significant patterns can be efficiently represented using shared *Sense Axis* instances (resp. *Transfer Axis* instances), together with a limited number of specific *Sense Axis* instances (resp. *Transfer Axis* instances) for representing variations between the languages.

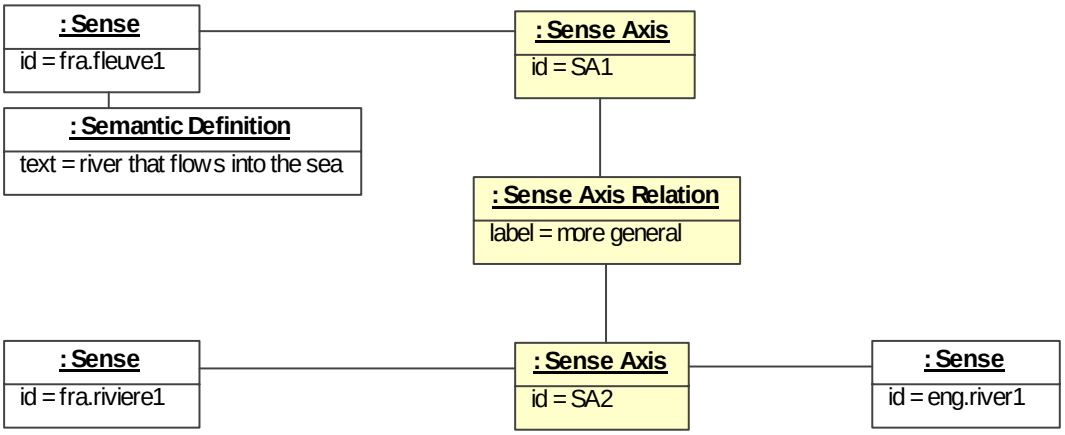
Direction and tests: some multilingual lexicons are very declarative in that every translation is represented by an interlingual object. Others are very procedural in that they restrict the translation by logical tests applied at the source or target language levels.

The multilingual notation package is defined as follows:



7 Example of multilingual notations

The example shown below illustrates how to use two intermediate *Sense Axis* instances in order represent a near match between *fleuve* in French and *river* in English, phenomenon that is called diversification and neutralization. The *Sense Axis* instance on the top is not linked directly to any English sense because this notion does not exist in English.



8 Connection with external systems like ontologies

8.1 Purpose

It is not the purpose of the semantic and the multilingual notation packages to provide a complex knowledge organization system.
LMF focus is NLP lexicons as required by user needs expressed through the channels of the National Delegations (DIN for Germany, AFNOR for France, etc).
But we must provide to our users a clear linking with these external systems.

8.2 Differences

A semantic node is a data structure representing the **meaning of a word in a particular language**.
A node in a knowledge representation system is a data structure representing **an elementary piece of what 'exists'.**
From a broader perspective, what 'exists' can be examined by separating issues of concept definition (ontology) and facts (concrete or imaginary facts), but from an LMF perspective, we stop where the meaning of word stops. In other terms, ontologies and fact data bases are considered as external systems.

8.3 Two criteria with regards to the linking with external systems

The situation can be viewed according to the following independant criteria:

- mono vs multilingual situations
- linkage with one or several external systems

8.4 mono vs multilingual situations

The context and requirement are rather different for a user working in a monolingual organization, compared to a multilingual situation. A monolingual user will tend to ignore other languages and take the shortest path. The nodes belonging to the semantic package will be considered as the pivot structure to be associated with external nodes.
To be more precise, within a monolingual context, two sub-strategies are possible depending on the presence vs absence of *Synset* instances. The rationale is based on the fact that *Sense* instances are mandatory and *Synset* instances are optional. When *Synset* instances are used; it is preferable to use them as connectors to external nodes since they group together synonyms. It should be noted that LMF does impose guidelines for defining what is a synonym compared to what is not. This is left to the lexicon manager. For instance, a lexicon manager may consider that a slang meaning for a particular word is a synonym of non-slang meaning of this given word. And another

lexicon manager may adopt the opposite decision. When Synset instances are not used, it is preferable to use *Sense* instances as connectors to external nodes.
In a multilingual environment, the situation is completly different because interlingual nodes are present. The lexical resource is comprised of *Sense Axis* instances and *Transfer Axis* instances. Because the latter are dedicated to the connection of subcategorization frames, mainly for verbs and predicative nouns at the syntactic level, they are not a great help.
In a multilingual lexical resource, *Sense Axis* instances are the perfect connectors for linking external nodes.

8.5 Linkage with one or several external systems

Let's recall that a lexical resource is a resource that is shared by a great number of people, at different levels.
The definition of a shared ontology does not seem to be practical. Under normal social conditions, such as a free society that allows a wide range of political and social thought, many ontologies will simultaneously exist and compete for adherents. Permanently adopting any single rigid system is unlikely, and probably undesirable.
Because any ontology is, among other things, a social / cultural artifact, there is no purely objective perspective from which to observe the whole terrain of concepts.
That being said, at a single lexical resource level, the only possible sharable data structure seems to be an upper ontology (aka fundation ontology) like OpenCyc, SUMO, Basic Formal Ontology, DOLCE, DnS or General Formal Ontology.
For specialized ontologies, pragmatic issues being so important, a common shared data structure does not seem to be conceivable.
From the perspective of LMF, we don't have any other choice as to provide the means to connect several external systems and to leave this decision to the lexicon manager.

8.6 Provided mechanism

From a modeling point of view, the mechanism cannot be a naive attribute adornment because the cardinality is one to many: intermediate classes must be designed for this purpose.
Therefore, the connection with external systems is provided by two classes *Monolingual External Ref*[4] class and *Multilingual External Ref*[5] class. These classes are adorned by /externalSystem/ and /externalReference/ attributes that refer respectively to the name(s) of the external system and to the specific relevant node in this given external system.

9 Concrete lexicons

Various prototype efforts are currently underway in different countries to create lexicons from scratch or to transform existing lexicons into LMF compliant models.

These data mainly deal with morphology and syntax in several dozen of European, Semitic [Khemakhem] and Asian languages.

Concerning semantic representations, the semantic model has been recently applied and tested in LeXFlow, a prototype tool designed at CNR as a platform for interoperability and integration of monolingual semantic lexicons with differently conceived architectures and diverging formats, such as two Italian lexicons from the SIMPLE and WordNet families [Soria 2006]. The system, as a general, versatile framework enabling automatic lexical resource integration, is particularly suited for the management of distributed lexical resources and for proving new cooperation methods among lexicon experts.

LMF is being currently adopted in the NEDO project (Japan grant) concerning the creation of a common standard for Asian language resources. The project aims at (i) building a description framework of lexical entries and instanciating sample lexicons in OWL and (ii) developing an upperlayer ontology. The proposed framework will be evaluated through an application in CLIR[6].

---

[4] See Semantic package

[5] See Multilingual notation package

[6] Cross Lingual Information Retrieval

Within the BootStrep project[7], LMF is currently the starting point for the definition of BioLexicon, a domain-specific lexicon related to the biological domain. The entries are being tuned to the representation of language-specific information about terms (entities and events) pertaining to the field of gene-regulation and is going to be linked to the Bio-Ontology which will provide language independent knowledge for the same domain. Together these resources will constitute the terminological backbone for supporting Text Mining and Information Extraction applications.

These two last projects are in the same key research directions, i.e. the linking between semantic lexical representation and the conceptual representation ; a challenging task which requires much further investigation.

10 Related tasks within ISO

Two other important tasks are currently being conducted in parallel and in relation with LMF within ISO-TC37/SC4.
The first one is the work done in the Data Category Registry in order to describe all the constants for all languages. Three sub-groups work in parallel (we call them 'profiles'): one for morpho-syntax, one for syntax and one for semantics.
The second import body of work deals with annotation: two standards are in preparation: one for morpho-syntactic annotation (tagger results) and one for syntactic annotation (parser results).
A great deal of energy is spent assuring that all these specifications can be used in a coherent manner.

11 Conclusion

The main focus of LMF is to provide a common, standardized framework for NLP lexicons.

It is certainly not the purpose of the semantic and the multilingual notation packages to provide a complex knowledge organization system. Ideally, LMF should rely on one or several several external systems designed specifically for that purpose.

LMF provides two classes *Monolingual External Ref* class and *Multilingual External Ref* class, for the connection with external knowledge systems, depending on the mono vs multilingual situation.

Acknowledgements

The work presented here is partially funded by the EU eContent-22236 LIRICS project[8].

References

Francopoulo G., George M., Calzolari N., Monachini M., Bel N., Pet M., Soria C. 2006 Lexical Markup Framework, LREC Genoa

Gamma E., Helm R., Johnson R., Vlissides J 1995 Design Patterns - Elements of reusable Object-Oriented software, Addison-Wesley

Khemakhem A. 2006 ArabicLDB, une base lexicale normalisée pour la langue arabe, Mémoire de Master, Université de Sfax, Tunisia

Rumbaugh J., Jacobson I., Booch G. 2004 The unified modeling language reference manual, second edition, Addison-Wesley

Soria C., Tesconi M., Bertagna F., Calzolari N., Marchetti A., Monachini M. 2006 Moving to dynamic computational lexicons with LeXFlow, LREC Genoa

[7] www.bootstrep.org/bin/view/Extern/WebHome

[8] http://lirics.loria.fr