# Extended examples of lexicons using LMF
(auxiliary working paper for LMF)

**Gil Francopoulo INRIA-Loria**

**29 August 2005**

## A.1 Foreword

The subject of this section is to study lexicons in order to describe the way they fit or they map to LMF/NLP. The first example is an invented one and is called “getting started lexicon”. The other examples are real existing lexicons.

When available, raw data are presented.

Concerning UML to XML conversion, the following conventions are adopted:

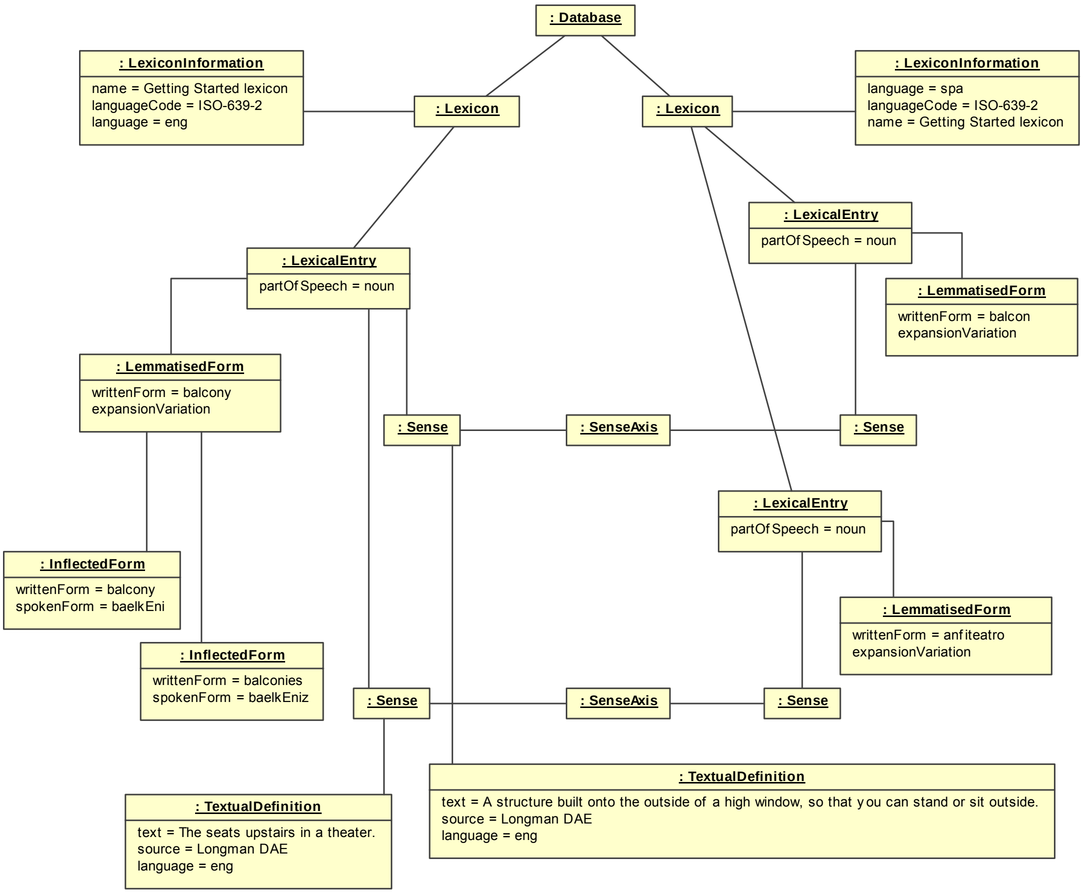
- each UML attribute is transcoded as an XML attribute;
- UML aggregations are transcoded as content inclusion;
- UML shared associations (i.e. associations that are not aggregations) are transcoded as XML attributes;
- each UML class is transcoded as an XML element.

## A.2 Getting started lexicon

This example is not a real NLP existing lexicon but an invented one taken from a publishing dictionary [26].

This lexicon is named “Getting started lexicon” and ISO-639-2 will be used for language coding, in order to specify languages with three lower-case letters. As a consequence, these values will be recorded inside the Lexicon Information instance.

We will describe the English word “balcony” and the Spanish words “balcón” and “anfiteatro”. According to [26], the word “balcony” has two senses 1) a structure built onto the outside of a high window, so that you can stand or sit outside 2) the seats upstairs in a theater. On the English side, the morphology comprises graphical and phonetic inflected forms. Nothing is said in syntax. We have two senses and two textual definitions. Each sense is connected to a Sense Axis. On the Spanish side, we have two senses connected to two different lexical entries. The UML instance diagram is as follows:



The data could be expressed by the following XML file:

```
<!—-                          DataBase level section →
<Database>
<Lexicon>
<LexiconInformation name=”Getting started lexicon” languageCode=”ISO-639-2” language=”eng”/>
<!—                           English section →
<LexicalEntry partOfSpeech=”noun”>
        <LemmatisedForm writtenForm =”balcony”/>
        <InflectedForm writtenForm=”balcony” spokenForm=”baelkEni”/>
        <InflectedForm writtenForm=”balconies” spokenForm=”baelkEniz”/>
        </Form>
        <Sense axis=”A1”/>
        <TextualDefinition text=”A structure built onto the outside of a high window, so that   you can stand
or sit outside” source=”Longman DAE” language=”eng”/>
        </Sense>
        <Sense axis=”A2”/>
        <TextualDefinition   text=”The   seats   upstairs   in   a   theater”   source=”Longman   DAE”
        language=”eng”/>
        </Sense>
</LexicalEntry>
</Lexicon>
<!—                           Multilingual section →
<SenseAxis id=”A1”/>
<SenseAxis id=”A2”/>
<!—-                          Spanish section →
<Lexicon>
<LexiconInformation name=”Getting started lexicon” languageCode=”ISO-639-2” language=”spa”/>
<LexicalEntry language=”spa” partOfSpeech=”noun”>
        <LemmatisedForm writtenForm=”balcón“/>
        <Sense axis=“A1“/>
</LexicalEntry>
<LexicalEntry language=”spa” partOfSpeech=”noun”>
        <LemmatisedForm writtenForm=”anfiteatro“/>
        <Sense axis=“A2“/>
```

```
</LexicalEntry>
</Lexicon>
<!—-                              Closing DataBase level section →
</Database>
```

## A.3 LMF and OLIF

### A.3.1 Presentation

Open Lexicon Interchange Format (OLIF) has its origin in the Open Translation Environment for localization (OTELO) project, which worked on a multi-vendor machine translation environment and was funded by the EC in the 4th Framework program. The version of OLIF (OLIF-1) was a lean and flat format for lexicon exchange. OLIF-2 was defined later and is XML compliant. The consortium contained Microsoft, SAP, Basis, Trados, Systran, Xerox (www.olif.net).

The basic idea of OLIF is to facilitate the exchange of primarily the pivotal information in entries. This information should be easily compiled into information that is needed by other formalisms. OLIF also provides the option of a deeper lexical representation. Included in the OLIF format, is general coverage of inflection paradigms, verb argument structure, semantic types and selection restrictions.

Each entry is **uniquely** defined by a set of key data: canonical form, part of speech, language code, subject area, and in the case of homonyms, a semantic reading. Entries represent independent semantic units (e.g. bank/river and bank/economy are two different entries). In addition to these obligatory key data, several groups of optional attributes can be used: a) Detailed monolingual description (e.g. grammatical gender). b) Cross-reference information, indicating related entries in the language of the entry itself (e.g. abbreviation). c) Transfer information indicating entries in languages different from the language of the entry itself, which may serve as translations if certain conditions hold. With the conventions that a "?" means optional, "*" means zero or more and "+" means one or more, such a structure is illustrated in the following figure:

```
monolingual section
        key description
                canonical form
                language
                part of speech
                subject field
                semantic reading ?
        monolingual description ?
                monolingual administration ?
                monolingual morphology ?
                monolingual syntax ?
                monolingual semantics ?
        general description ?
cross reference *
        key descriptor
        (link type
        general data data category) +
transfer *
        key descriptor
        transfer restriction
                contextual expression
                test expression
                action *
```

### A.3.2 Application

The most important characteristic in OLIF is that an unique key is mandatory in order to ease interoperability. And the structure attached to the key is largely left underspecified.
OLIF is targeted for European languages. Inflectional paradigms are explicitly listed for five languages (i.e. fra, eng, ger, por, spa) but there is no mechanism to define any paradigm for other languages like in LMF. Each Inflectional paradigm is defined as four fields: numeric code, part of speech, example and a textual definition. There is no possibility to define the character string operations like in LMF.
OLIF comprises recommendations for defining the canonical form in case of simple words and multi-word expressions on a per language basis with six covered languages [29]. OLIF has not been designed with Semitic and Asian languages in mind [30] so nothing is provided for multi-orthographic languages.
OLIF is well suited for localization process and simple translations.

### A.3.3 Data within OLIF

```
<entry EntryUserId="2312">
        <mono MonoUserId="2311">
                <keyDC>
                        <canForm>Briefkurs</canForm>
                        <language>de</language>
                        <ptOfSpeech>noun</ptOfSpeech>
                        <subjField>gac-fi</subjField>
                        <semReading>b</semReading>
                 </keyDC>
                <monoDC>
                        <monoAdmin>
                                <syllabification>brief-kurs</syllabification>
                                <entryFormation>cmp</entryFormation>
                                <originator>FISHERF</originator>
                                <adminStatus>ver</adminStatus>
                                <entrySource>sapterm</entrySource>
                                <company>sap</company>
                        </monoAdmin>
                        <monoMorph>
                                <morphStruct>brief:kurs</morphStruct>
                                <inflection>like Tisch</inflection>
                                <head>kurs</head>
                                <gender>m</gender>
                        </monoMorph>
                        <monoSyn>
                                 <synType>cnt</synType>
                        </monoSyn>
                        <monoSem>
                                <semType>meas</semType>
                        </monoSem>
                </monoDC>
                <generalDC>
                        <updater>HANSENPOU</updater>
                        <modDate>1999-28-01</modDate>
                        <usage>online</usage>
                        <note>online-A</note>
                </generalDC>
        </mono>
        <transfer>
```

```
                    <keyDC>
                              <canForm>bank selling rate</canForm>
                              <language>en</language>
                              <ptOfSpeech>noun</ptOfSpeech>
                              <subjField>gac-fi</subjField>
                              <semReading>b</semReading>
                    </keyDC>
                    <equival>full</equival>
          </transfer>
</entry>
```

### A.3.4 Migration into LMF

OLIF entries fit into LMF structure. The key description splits into three LMF classes: Form, Lexical Entry and Sense. Concerning the subject field, a set of 37 categories (taken from Eurodicautom) is pre-defined in OLIF. In LMF, the corresponding values must be taken from the ISO-12620 DCR. So, prior to import, 37 Semantic Features must be created in order to be connected to the senses. In OLIF, it is possible to use a non defined value, and in this case, such a value much be managed on the fly.

Import of an OLIF entry implies the following operations:

| within OLIF | type of operation | concerned LMF class |
|---|---|---|
| canonical form | creation of an instance | Form |
| Language | set a data category | Lexical Entry |
| part of speech | set a data category | Lexical Entry |
| subject field | creation of an instance and connect a semantic feature | Sense |
| semantic reading | set the label | Sense |

Of course, this kind of insertion tends to produce small atomic instances and LMF does not impose any particular grouping. So, in order to minimize the number of instances, and in a second phase, the user can group together Lexical Entries and Forms that share the same values for the form, language and part of speech.

The OLIF transfer descriptors imply the creation of a Sense Axis within the LMF context.

For the other values, it does not seem to be possible to process automatically in a generic manner (i.e. whatever OLIF source it is). But if a coherent strategy has been respected within the OLIF data, an automatic processing can be applied.

### A.3.5 Data within LMF

```
<Database>
<Lexicon>
<LexiconInformation           language=”ger”/>
<LexicalEntry        partOfSpeech=”noun”
          <LemmatisedForm  writtenForm=”Briefkurs“
                    inflectionalParadigm=”likeTisch”>
                    <Decor    syllabification=“brief-kurs“
                              entryFormation=“cmp“
                              originator=“FISHERF“
                              adminStatus=“ver“
                              entrySource=“sapterm“
                              company=“sap“
                              morphoStruct=“brief :kurs“
                              head=”kurs”/>
```

```
        </Form>
        <SyntacticBehavior frame=“FRcnt“/>
        <Sense  label=”b”
                semanticFeatures=“SFgacfi SFmeas“
                axis=”A1”/>
        <Décor  updater=”HASENPOU”
                modDate=”1999-28-01”
                usage=”online”
                note=”online-A”/>
</LexicalEntry>
<InflectionalParadigm       id=”likeTisch”
                            gender=”masculine”/>
<SemanticFeature            id= “SFgacfi”
                            att=”subjectField”
                            val=”gac-fi”/>
<SemanticFeature            id=”SFmeas”
                            att=”semType”
                            val=”meas”/>
</Lexicon>
<SenseAxis id=”A1”/>
<Lexicon>
<LexiconInformation         language=” eng”/>
<LexicalEntry       partOfSpeech=”noun”
        <LemmatisedForm  writtenForm=”bank selling rate”/>
        <Sense  label=”b”
                semanticFeatures=”SFgacfi”/>
</LexicalEntry>
</Lexicon>
<Database>
```

## A.4 LMF and CLIPS

### A.4.1 Presentation

“Corpora e Lessici dell’Italiano Parlato e Scritto” (CLIPS) was a three-year Italian national project headed by the “Consiglio Nazionale delle Ricerche, Istituto di Linguistica Computazionale” (CNR-ILC) that started in 2000 (www.ilc.cnr.it/clips/CLIPS_ENGLISH.htm). One of its main objectives was to create a wide-coverage and multipurpose computational semantic lexical database for Italian, by extending the PAROLE-SIMPLE lexicon which share with other eleven European lexica a common conceptual model, representation language and lexicon building methodology. The underlying theoretical model is grounded on the EAGLES project recommendations and, at semantic level, it implements and extends major aspects of Generative Lexicon (GL) theory; nevertheless, the lexicon is not strictly theory-dependent. The model enables a very fine-grained description to be performed, but allows a more shallow one too, in so far as the information provided meets the model requirements.

To date, in 2005, the lexicon is the largest Italian computational lexical resource. The lexicon consits of 53 000 lemmas encoded at morphological level and phonological level (for a total of about 390 000 word-forms), 51 000 lemmas encoded at syntactic level, and 57 000 semantically encoded word senses. Conformity of the data to the model is ensured by an XML DTD, whereas internal formal validation is performed by an XML parser.

### A.4.2 Data within CLIPS

Like in Eagles, a word is a chain of small elements starting from morphology, crossing syntax and ending in semantics. The chain begins with a morphological unit that is simple (i.e. a MuS), a graphical morphological unit (i.e. a Gmu), a syntactic unit (i.e. a SynU), an intermediate object between syntax and semantics called a CorrespSynUSemU, a semantic unit (i.e. a SemU) and a predicate. It's not very easy to present CLIPS entries because a lot of objects are concerned. The structure is not a tree but it is a graph of interconnected objects. In the following entries, only some specific characteristics are presented and the cut branches of the network are signaled with an XML comment.

Two features are presented here :

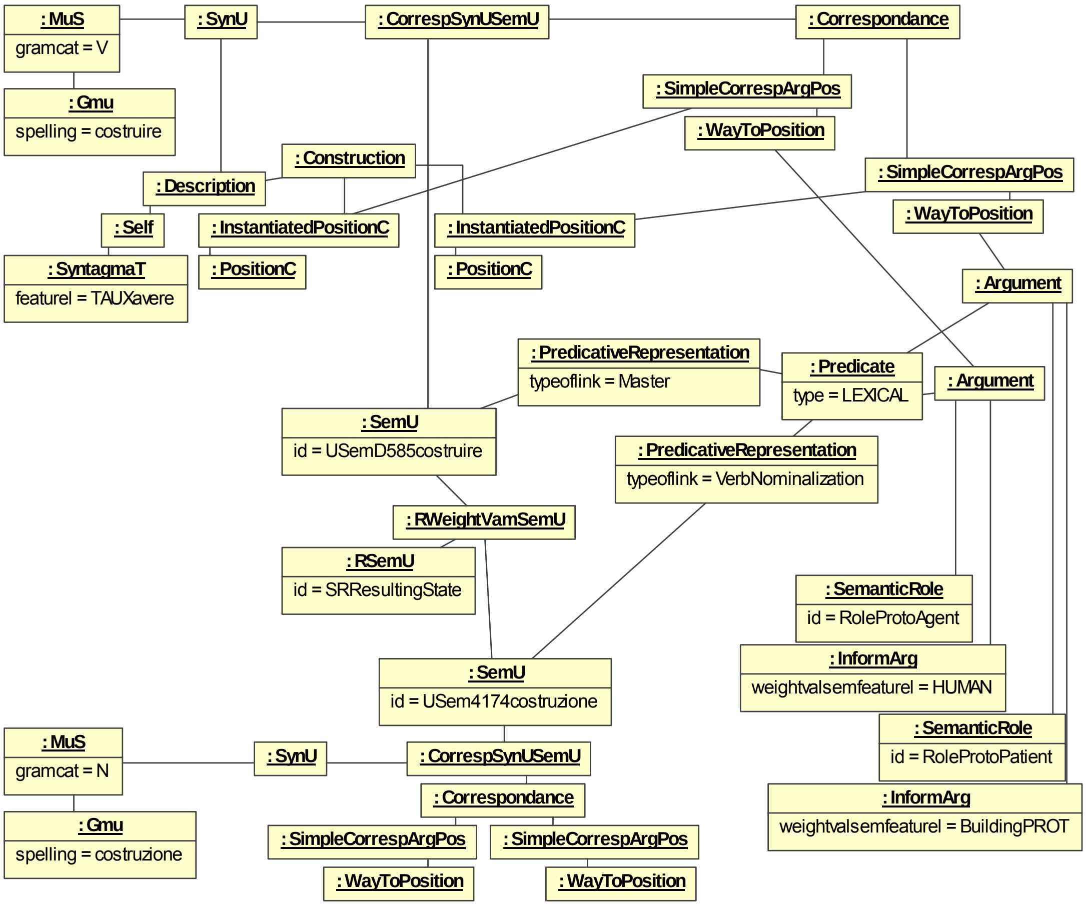
1. The linkage between syntax and semantic for the entry “costruire” (“to build” in English).

2. The semantic derivation around “costruire”. One entry is the verb “costruire” and the second entry is the noun “costruzione”. These two entries are bound at the semantic level by two sorts of links: a) they share the same predicate b) there is a relation that states that “costruzione” is the resulting state of “costruire”.

```
                        <!—Morphology of the first entry→
<MuS    gramcat=”V”
        synulist=”SYNUcostruireV”>
        <Gmu    inp=”GINP446”>    <!—refers to inflectional paradigm, not expanded here →
                <Spelling>costruire</Spelling>
        </Gmu>
</MuS>
                        <!—Syntax of the first entry→
<SynU   id=”SYNUcostruireV”
        example=”costruire un ponte ; - una storia ; - una frase”
        description=”txa”>
                        < !—The verb has three meanings but only the following is presented →
        <CorrespSynUSemU        targetsemu=”USemD585costruire”
                                correspondance=”ISObivalent”/>
</SynU>
<Description    id=”txa”
                example=”abbassare un muro ; - la testa”
                self=”SELFVxa”
                construction=”t”/>
<Construction   id=”t”
                syntlabel=”Clause”>
                        < !—Subcategorization frame→
        <InstantiatedPositionC  range=”0”       optional=”YES”  positionC=”Psubj”/>
        <InstantiatedPositionC  range=”1”       optional=”NO”   positionC=”Pobj”/>
</Construction>
                        < !—Surface syntactic realizations (not expanded)→
<PositionC      id=”Pobj”       function=”OBJECT”       syntagmacl=”SNTnp”/>
<PositionC      id=”Psubj”      function=”SUBJECT”      syntagmacl=”SNTnp”/>
                        <!—Information about auxiliary is given in the Self→
<Self           id=”SELFVxa”
                syntagmatl=”STVxa”/>
<SyntagmaT      id=”STVxa”
                syntlabel=”V”
                featurel=”TAUXavere”/>
                        <!—Type of linkage between syntax and semantics→
<Correspondance         id=”ISObivalent”
```

```
                        correspargposl=”ARG0P0 ARG1P1”
                        comment=”isomorphic mapping for bivalent predicates”/>
<SimpleCorrespArgPos    id=”ARG0P0”     accessPath=”0”>
                <WayToPosition  targetPosition=”0”/>
</SimpleCorrespArgPos>
<SimpleCorrespArgPos    id=”ARG1P1”     accessPath=”1”>
                <WayToPosition  targetPosition=”1”/>
</SimpleCorrespArgPos>
                        <!—Semantics of the first entry→
<SemU   id=”UsemD585costruire”
        example=”costruire un edificio”>
        <PredicativeRepresentation  typeoflink=”Master”             predicate=”PREDcostruire-1”/>
        <RweightVamSemU             target=”Usem4174costruzione”    semr=”SRResultingState”/>
</SemU>
<Predicate      id=”PREDcostruire-1”    type=”LEXICAL”
                argumentl=”ARG0costruire-1 ARG1costruire-1”/>
<Argument       id=”ARG0costruire-1”    semanticrolel=”RoleProtoAgent”
                informargl=”INFARGN2”/>
<Argument       id=”ARG1costruire-1”    semanticrolel=”RoleProtoPatient”
                informargl=”INFARGT103”/>
<InformArg      id=”INFARGN2”           weightvalsemfeaturel=”HUMAN”/>
<InformArg      id=”INFARGT103”         weightvalsemfeaturel=”BuildingPROT”/>
<SemanticRole   id=”RoleProtoAgent”
                example="soggetto di pensare, fare, sapere"
                comment="Usually 'translate' as SUBJECTS in surface. They can occur in the following
contexts: Maria fa, sa, ecc. compare with non-ProtoAgent subjects"/>
<SemanticRole   id=”RoleProtoPatient”
                example="uccidere"
                comment="Direct Objects plus weakly bound prepositional complements such as credere
in"/>
<RsemU          id=”SRResultingState”
                comment=”Usem1 is a transition and Usem2 is the resulting state of the transition"/>
                                <!—Morphology of the second entry →
<MuS    gramcat=”N”
        synulist=”SYNUcostruzioneN SYNUcostruzioneN2”> <!—only the first one is expanded→
        <Gmu    inp=”GINP157” <!—refers to inflectional paradigm, not expanded here →
                <Spelling>costruzione</Spelling>
        </Gmu>
</MuS>
                                <!—Syntax of the second entry →
<SynU   id=”SYNUcostruzioneN”   <!—description is not expanded for this entry→
        example=”la cosa costruita; ordinata disposizione delle parole in una frase o delle frasi in un
periodo”
        <CorrespSynUSemU        targetsemu=”Usem4174costruzione”
                                correspondance=”CROSSEDbivalent”/>
</SynU>
                        <!—Type of linkage between syntax and semantics→
<Correspondance         id=”CROSSEDbivalent”
                        correspargposl=”ARG0P1 ARG1P0”
                        comment=”crossed isomorphic mapping for bivalent predicates”/>
<SimpleCorrespArgPos    id=”ARG0P1”     accessPath=”0”>
        <WayToPosition  targetPosition=”1”/>
</SimpleCorrespArgPos>
<SimpleCorrespArgPos    id=”ARG1P0”     accessPath=”1”>
        <WayToPosition  targetPosition=”0”/>
</SimpleCorrespArgPos>
                        <!—Semantics of the second entry→
<SemU   id=”Usem4174costruzione”
        example=”una orribile costruzione deturpa il paesaggio”>
        <PredicativeRepresentation   typeoflink="VerbNominalization"
                                              predicate="PREDcostruire-1"/>
</SemU>
```

The XML data is more easily readable on the following UML diagram:



## A.4.3 Migration into LMF

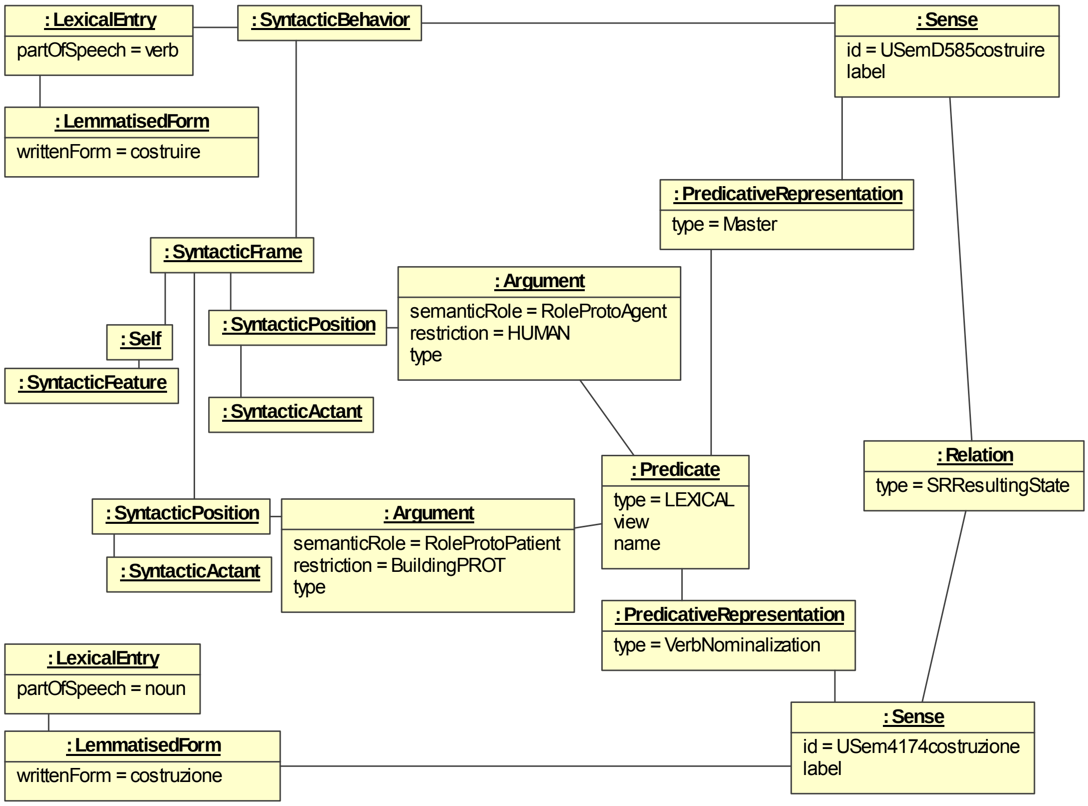
The structure of the LMF NLP extensions is very similar to Eagles. There are three main differences:

- The class names are taken from "ordinary" usage instead of being coined. For instance, in LMF the name for SemU is Sense. The motivation of this choice is to ease model understanding.

- The structure is a little bit lighter because several small intermediate classes have been removed. In Eagles, these classes had little usefulness but were mandatory and as a consequence, they make the whole model very complex. CorrespSynUSemU is one of these classes.

- A little bit more flexibility in order to lighten descriptions. For instance, it is possible to describe a sense without recording a syntactic description when nothing is to be said in Syntax.

### A.4.4 Data within LMF



## A.5 LMF and LC-Star

### A.5.1 Presentation

LC-STAR is an IST-project focusing on creating language resources for speech-to-speech translation components and thus improving human-to-human and man-machine communication in multilingual environments (www.lc-star.com). The objectives are flexible vocabulary speech recognition, high quality text-to-speech synthesis and speech centered translation into selected languages. The components are designed to be embedded in mobile appliances and network servers. Lexicons for 13 languages have been created: Catalan, Finnish, German, Greek, Hebrew, Italian, Mandarin Chinese, Russian, Slovenian, Spanish, Arabic, Turkish and US-English. The consortium contains companies like Nokia, Siemens and IBM.

### A.5.2 Data within LC-STAR

For each entry, orthography is defined as the correct way of writing a given inflected form. When more than one spelling is acceptable, the most common spelling for an inflected form is taken. Orthography is the key to all of the phonetic, morphological and POS information for that entry. This structuring is coded as entry groups. Therefore, words that can be associated to multiple POS have a unique entry. Multi-token entries are allowed with blanks replaced with underscores (e.g. New_York). Phonetic transcription is coded for each entry with SAMPA phonetic alphabet with stress marker, syllable boundary marker and tone markers. Multiple pronunciations are coded if they are common use. For each group, the lemmatised form is specified with the POS(s) it belongs to. For each POS, where applicable its attributes are described (e.g. number, person …) depending on the language. Additional rules are given for each of the 13 described languages covering orthography, syllabification, lemmatisation and pairing POS with morphological features.

Two statements can be made:

- Data is organized toward recognition: each entry goes from the inflected form to the lemmatised form.

- For highly inflected languages like German or Hungarian, using such a format produces numerous duplicated representations. The lexicon is huge and hardly manageable. In these languages, this format could be considered more as a delivery format than a management format. Inflectional paradigms are more suited for management.

An example in Arabic is as follows:

```
<ENTRYGROUP orthography="الحمراء" xml:lang="ar">
        <ENTRY>
                <NOM     class="common"     gender="feminine"   number="singular" />
                <LEMMA>ءارمح</LEMMA>
                <PHONETIC>a l – " X\ a m r a: ?</PHONETIC>
        </ENTRY>
        <ENTRY>
                <ADJ     case="genitive"      degree="positive"    gender="feminine" number="singular" />
                <LEMMA>ءارمح</LEMMA>
                <PHONETIC>a l – " X\ a m r a: ?</PHONETIC>
        </ENTRY>
</ENTRYGROUP>
```

An example in English is as follows:

```
<ENTRYGROUP orthography="read" xml:lang="en">
        <ENTRY>
                <AUX mood="finite" tense="present" person="not_3"/>
                <LEMMA>read</LEMMA>
                 <PHONETIC>" r i: d</PHONETIC>
        </ENTRY>
        <ENTRY>
                <AUX mood="finite" tense="past" person="invariant"/>
                <LEMMA>read</LEMMA>
                <PHONETIC>" r e d</PHONETIC>
        </ENTRY>
        <ENTRY>
                <AUX mood="participle" person="invariant" />
                <LEMMA>read</LEMMA>
                <PHONETIC>" r e d</PHONETIC>
        </ENTRY>
</ENTRYGROUP>
```

Within LC-Star, what is called “orthography” is the inflected form and not the lemmatised form. So this entry is valid for “read” and not for “reads”. Another entry must describe the inflected form “reads” as follows.

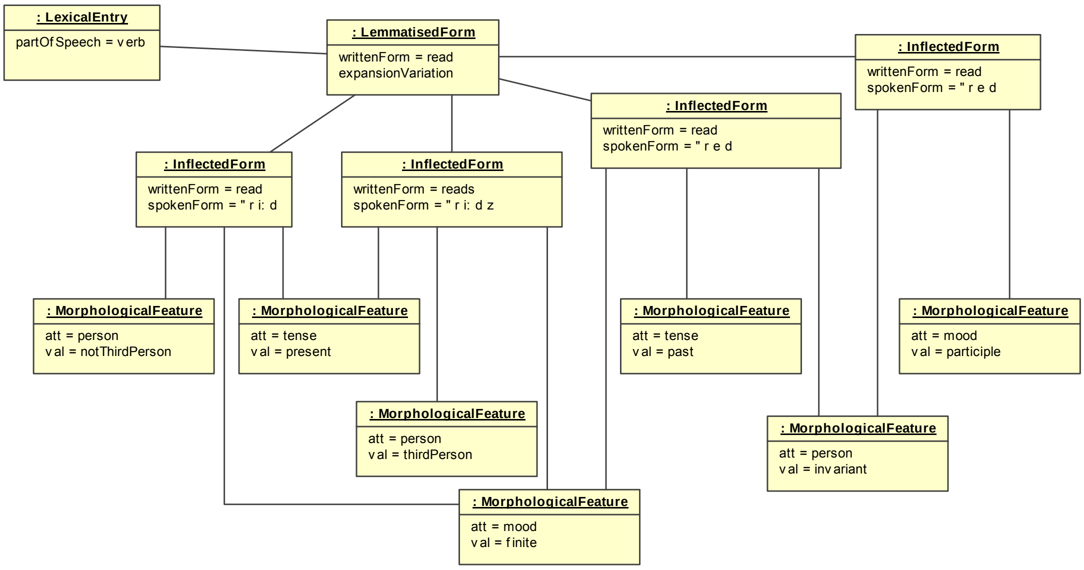
```
<ENTRYGROUP orthography="reads" xml:lang="en">
        <ENTRY>
                <AUX mood="finite" tense="present" person="3"/>
                <LEMMA>read</LEMMA>
                 <PHONETIC>" r i: d z</PHONETIC>
        </ENTRY>
</ENTRYGROUP>
```

### A.5.3 Migration into LMF

Data must be mapped. In LC-Star, the headword is the inflected form and everything is organized around that.

### A.5.4 Data within LMF

These two LC-Star entries give the following structuring:



Another option could be to automatically compute inflectional paradigms during import.

## A.6 LMF and WordNet

### A.6.1 Presentation

WordNet is an online English lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory [33]. Downloading the data being free, WordNet is one of the most popular lexical databases. And compared to WordNet-1.X, version-2.1 offers significant additions like morphology and derivation. WordNet is developed at Princeton University (http://wordnet.princeton.edu).

Information in WordNet is organized around logical grouping called synsets. English nouns, verbs, adjectives and adverbs are described. A word is considered as being either a single word or a MWE. In WordNet documentation, a MWE (e.g. “fountain pen” or “take in”) is called a “collocation” and this meaning is in contradiction with usual English definition. Each synset consists of a list of synonymous words and pointers that describe the relation between this synset and other synsets. A word may appear in more than one synset, and in more than one part of speech. The words in a synset are logically grouped such that are interchangeable in some context. Two kinds of relation are described: lexical and semantic ones. Lexical relations hold between word forms; semantic relation hold between word meanings. These semantic relations include hypernymy/hyponymy, antonymy, entailment, meronymy/holonymy. These semantic relations link the synonym sets in an ontological structure. But unlike what is often said in a too simplistic manner, WordNet is not an ontology of knowledge: WordNet contains an ontology of English meanings.

WordNet has two file formats: lexicographer (specified in WNINPUT documentation) and data file formats (specified in WNDB documentation). The first one is intended to human beings editing and is ruled by some defaulting behavior in order to avoid painful manual operations [34]. These source files are then processed by a data compiler called the “grinder” that expands into data file format. Data file format is an ASCII (not XML), readable but non-editable format. And this latter format attests more accurately than the first one of the real WordNet’s structure.

In the data file format, each line is a synset. Each line is structured as follows:

- field-1: synset identifier also known as synset offset

- field-2: not important for us
- field-3: part of speech
- field-4: number of words in the synset
- field-5: ASCII form of a word
- field-6: lex_id. One integer that when appended onto field-5 uniquely identifies a sense within a lexicographer file.

Field-5 and field-6 may be repeated. The rest of the line concerns pointers to other synsets.

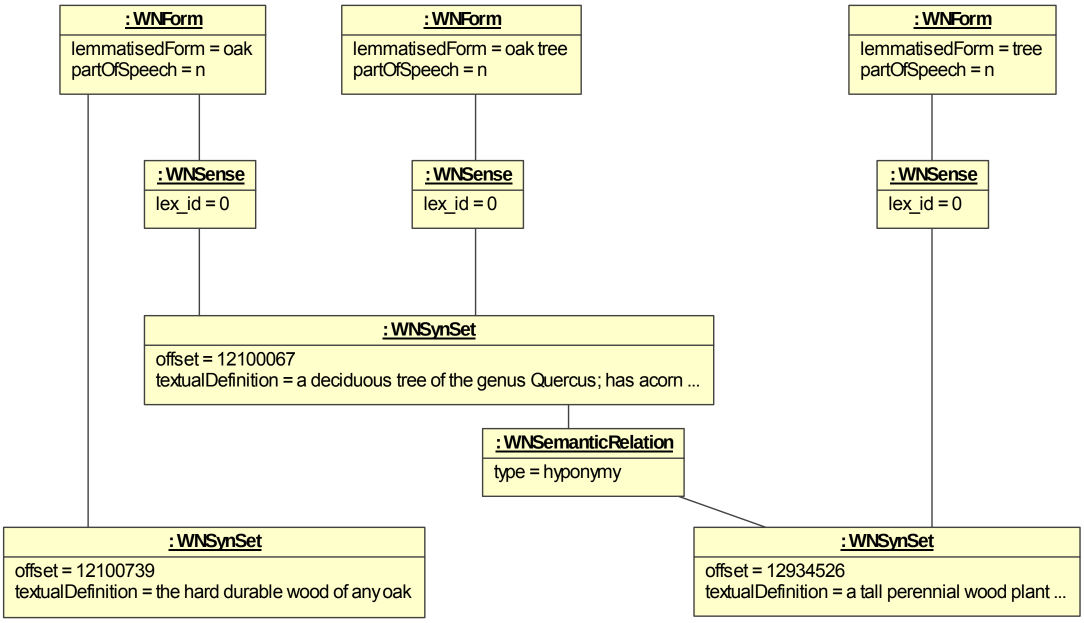
### A.6.2 Data within WordNet

Three synsets taken from WordNet-2.1 will be presented: “oak” in the sense of the tree, “oak” in the sense of the wooden material and “tree” in the sense of the plant. All forms are linked to one or several synsets. The synset oak/tree holds an hyponymy relation with the synset “tree”. In other words: “an oak tree is a tree”.

Raw data is as follows:

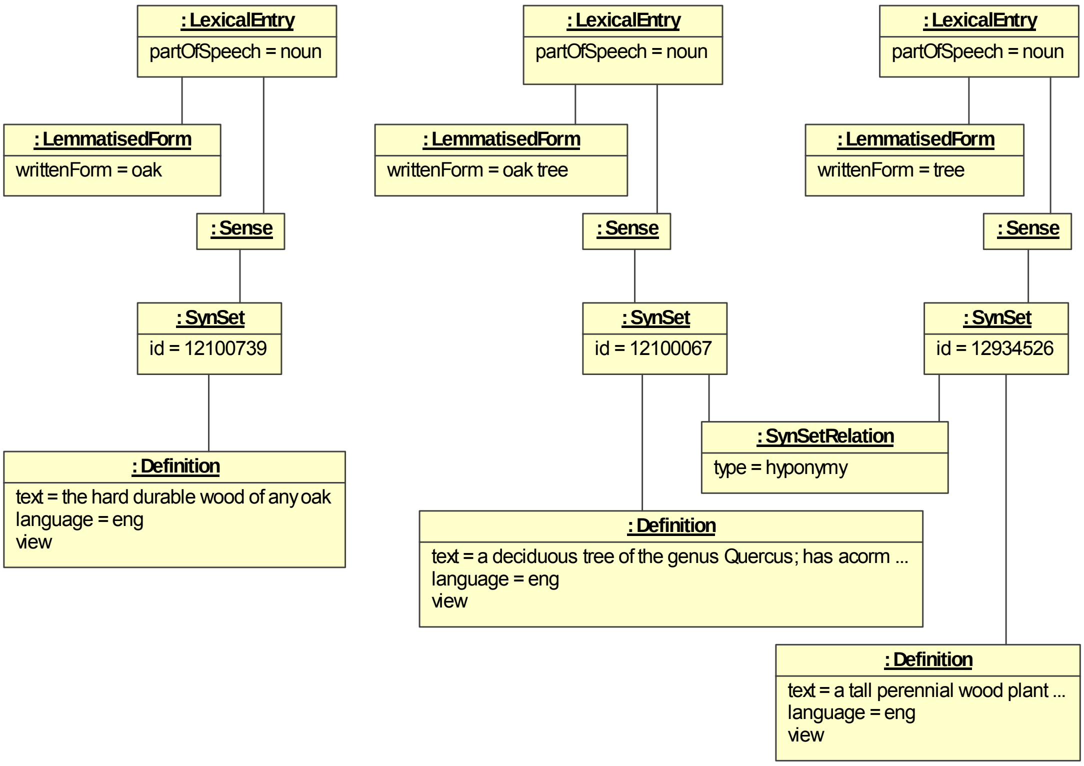
```
12100067 20 n 02 oak 0 oak_tree 0 029 @ 12934526 n 0000 #m 12099917 …

12100739 20 n 01 oak 2 004 @ 14900228 n 0000 #s 12100067 n 0000 …

12934526 20 n 01 tree 0 189 @ 12933603 n 0000 #m 08323882 n 0000 …
```

In order to ease reading, the structure is illustrated as followed:



### A.6.3 Data within LMF

The data fits within LMF as illustrated:



## A.7 LMF and FrameNet

### A.7.1 Presentation

The Berkeley FrameNet is an on-line lexical resource for English based on frame semantics and supported by corpus evidence [35]. The aim is to document the range of semantic and syntactic combinatory valences of each word in each of its senses, through manual annotation of example sentences and automatic capture and organization of the annotation results. At the time of Fall, 2003 release (version 1.1) the database included 7,500 lexical units, for which there are about 130,000 annotated sentences.

A lexical unit is a pairing of a word with a meaning. Typically, each sense of a polysemous word belongs to a different semantic frame, a script-like structure of inferences that characterize a type of situation, object or event. In the case of predicates or governors, each annotation accepts one word in the sentence which fill in information about a given instance of the frame. These phrases are identified with frame elements (FEs). These FEs are classified in term of how central they are to a particular frame distinguishing four levels: core, peripheral, extra-thematic and core-unexpressed. A core FE is one that instantiates a conceptually necessary particular or prop of a frame, while making the frame unique and different from other frames. Peripheral FE marks notions as Time or Place. Extra-thematic FE situate an event against a backdrop of another event, as in iteration: “Lee called the office [again]”. Core-unexpressed is used to avoid blind inheritance.

A frame semantic description of a predicative word derives from such annotations, identifies the frames which underlie a given meaning and specifies the ways in which FEs are realized in structure headed by the word.

A predicate is a constellation of triples that make up the FE realization for each annotated sentence, each triple consisting of a FE (say, PATIENT), a grammatical function (say, OBJECT) and a phrase type (say, NP). Valence descriptions of predicating words are generalizations over such structures.

FrameNet database consists in:

- Lexical units for individual word senses
- Descriptions of frames and FE with links from lexical units
- Annotation subcorpora.

### A.7.2 Data within FrameNet

Morphological descriptions are very simple: they are organized in a two level structure: lexical unit level with a lemmatised form and a definition and lexeme level with component descriptions when the entry is a MWE. In this case, each component is described specifying whether the component is a head and whether the MWE permits an insertion with the “breakBefore” attribute.

For instance in:
They called on the man.
*They called the man on.
compared with :
The called up the man.
They called the man up.

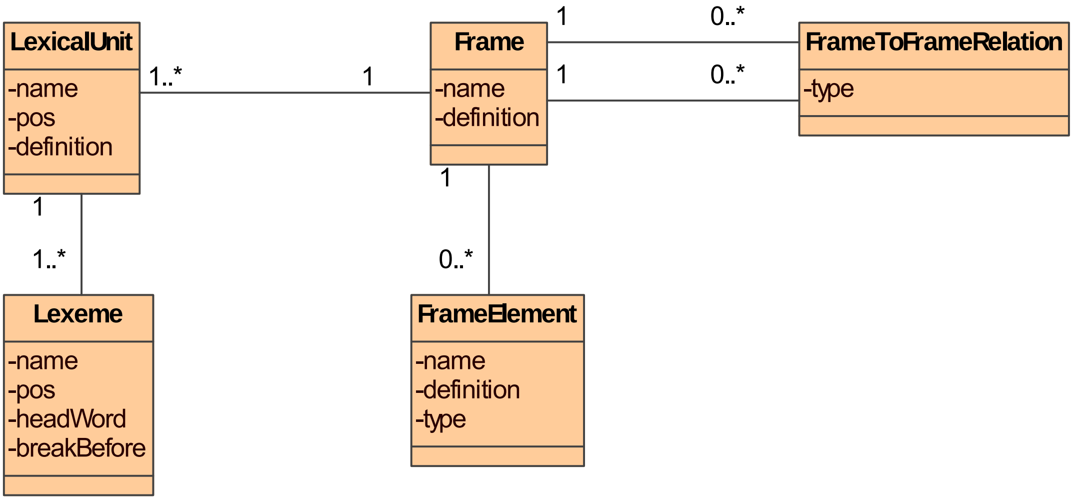
Thus, the “up” in “call up” is marked BreakBefore = 'Y', while the “on” in “call on” is marked BreakBefore = 'N'.

In order to ease reading: XML identifiers, inherited tags and book-keeping attributes (like creation date) have been removed.

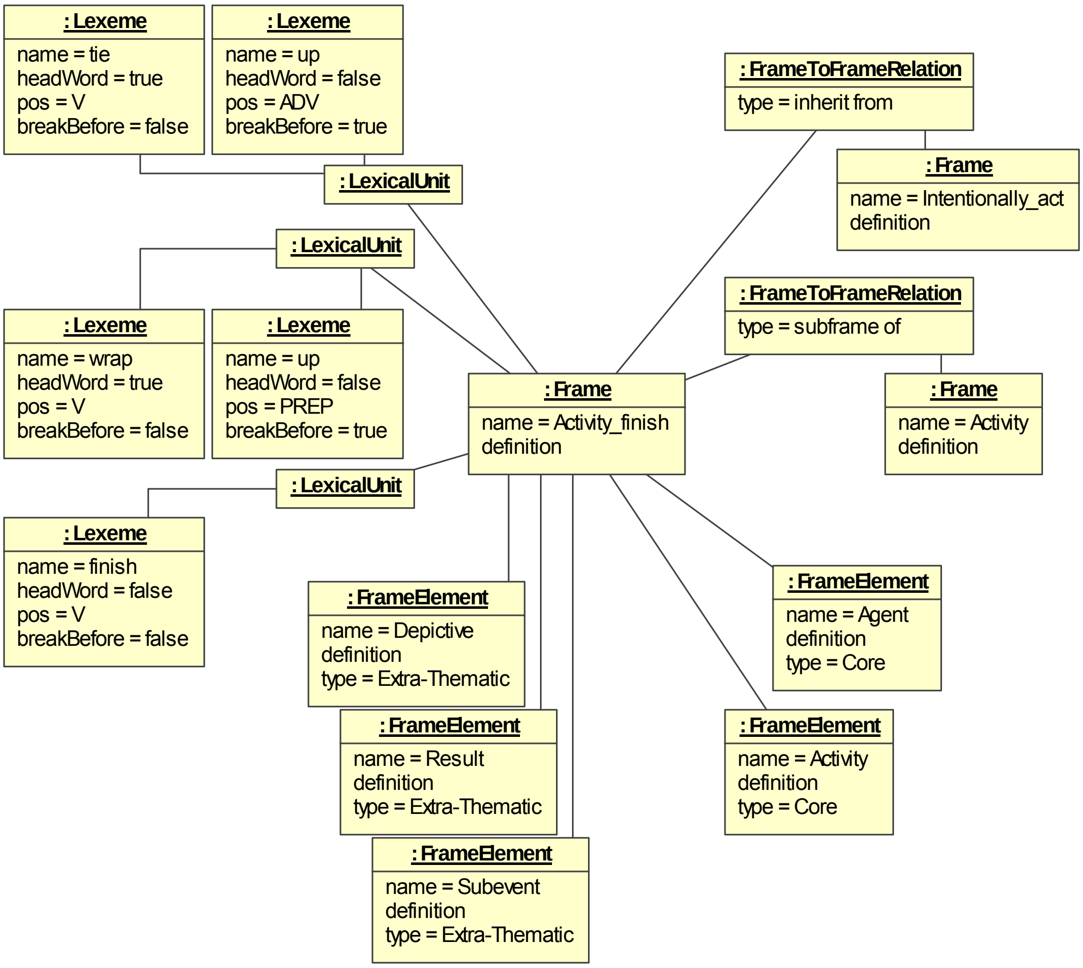
```
<frame name="Activity_finish">
 <--The frame inherits from intentionally_act and is a subframe of Activity-->
 <definition>An Agent finishes an Activity, which can no longer logically continue.</definition>
 <fes>
  <fe name="Depictive" coreType="Extra-Thematic">
   <definition>This FE identifies the Depictive phrase describing an actor or an undergoer of an action.
   </definition>
  </fe>
  <fe name="Result" coreType="Extra-Thematic">
   <definition>This FE identifies the Result of finished Activity.
   </definition>
  </fe>
  <fe name="Subevent" coreType="Extra-Thematic">
   <definition>This FE identifies the last Subevent of the finished Activity.Ex: The peace march  ENDED
["Sub" with a prayer].
   </definition>
  </fe>
  <fe name="Agent" coreType="Core">
   <definition>This FE identifies the Agent who has finished an Activity.
   </definition>
  </fe>
  <fe name="Activity" coreType="Core">
   <definition>This FE identifies the Activity that the Agent has finished.
   </definition>
```

```
    </fe>
    <!--Other FEs are inherited (like Manner, Place etc.) and not expanded here-->
  </fes>
  <lexunits>
    <lexunit name="tie up.v" pos="V">
      <definition>FN: bring something to a satisfactory conclusion
      </definition>
      <lexeme name="up" pos="ADV" breakBefore="true" headword="false" />
      <lexeme name="tie" pos="V" breakBefore="false" headword="true" />
      </lexunit>
    <lexunit name="wrap up.v" pos="V">
      <definition>FN: finish
      </definition>
      <lexeme name="up" pos="PREP" breakBefore="true" headword="false" />
      <lexeme name="wrap" pos="V" breakBefore="false" headword="true" />
    </lexunit>
    <lexunit name="finish.v" pos="V">
      <definition>FN: bringing or coming to an end
      </definition>
      <lexeme name="finish" pos="V" breakBefore="false" headword="false" />
    </lexunit>
  </lexunits>
</frame>
```

FrameNet Class model is illustrated by the following UML class diagram:



Instances showed in the example are illustrated by the following UML object diagram:



### A.7.3 Migration into LMF

Mapping of FrameNet against other models has already been studied like in [36] for MILE.

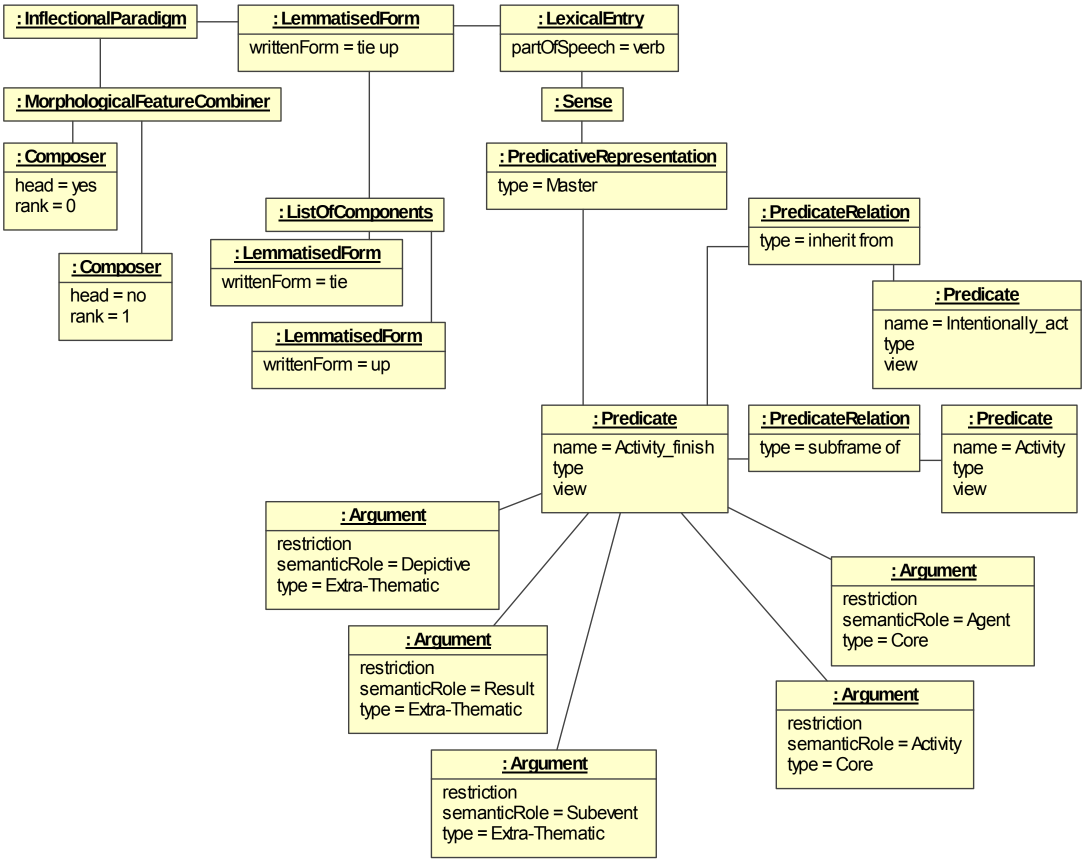
In FrameNet, the lexical unit is a complex notion that comprises the lemmatised form, the part of speech, components for MWE and semantic decomposition by the mean of a textual definition. Thus, within the LMF NLP extension, the FrameNet lexical unit is a sense and this sense is connected to a lexical entry.

The notion of Frame within FrameNet corresponds to a Predicate within LMF. The FrameToFrameRelation within FrameNet is a PredicateRelation within LMF.

The notion of FrameElement within FrameNet is an Argument within LMF. The attribute “name” corresponds to the attribute “semantic role”.

### A.7.4 Data within LMF

The example is illustrated by the following diagram in which only one of the three lexical units is drawn due to a lack of space.



## A.8 LMF and BDéf

### A.8.1 Presentation

BDéf is a formal database of lexicographic definitions fully derived from the four volumes of the *Explanatory Dictionary of Contemporary French* (ECD) [22]. The BDéf has two aims: firstly, it will make a representative subset of formal lexicographic definitions available for research in computational semantic. Secondly, building the BDéf permits to conduct a much needed research on the internal structuring of lexical meanings and how it should be modeled [18].

### A.8.2 Data within BDéf

The following French example presents the sense of “défierI.1” as described in [18]. This sense is taken from DEC-4 and corresponds to “Marcel a défié ce pédant en duel à l’épée”. Each definition is a structured set of elementary propositions. Elementary proposition is the smallest unit to manage. **These propositions are not based on smaller semantic primitives, but instead contain a formalized text.** BDéf is chosen in order to exhibit how such textual definitions could fit inside the LMF classes Sense, Definition and Proposition.

Propositional form

X défier Y en W à Z

Central component

1 : X communiquer à Y que *2

2 : X vouloir *3

3 : Y prendre_part à Y avec Z

Specific differences

/*objective*/

4 : *1 dans_le_but *5

5 : X punir Y pour *9

/*means*/

6 : façon de X de *5 être *7

7.1 : X blesser#il Y

7.2 : X tuer Y

/*situation*/

8 : X croire *9

9.1 : Y insulter X

9.2 : Y porter_atteinte à honneur de X

10 : *9 passé

Variable typing

X : individu

Y : individu

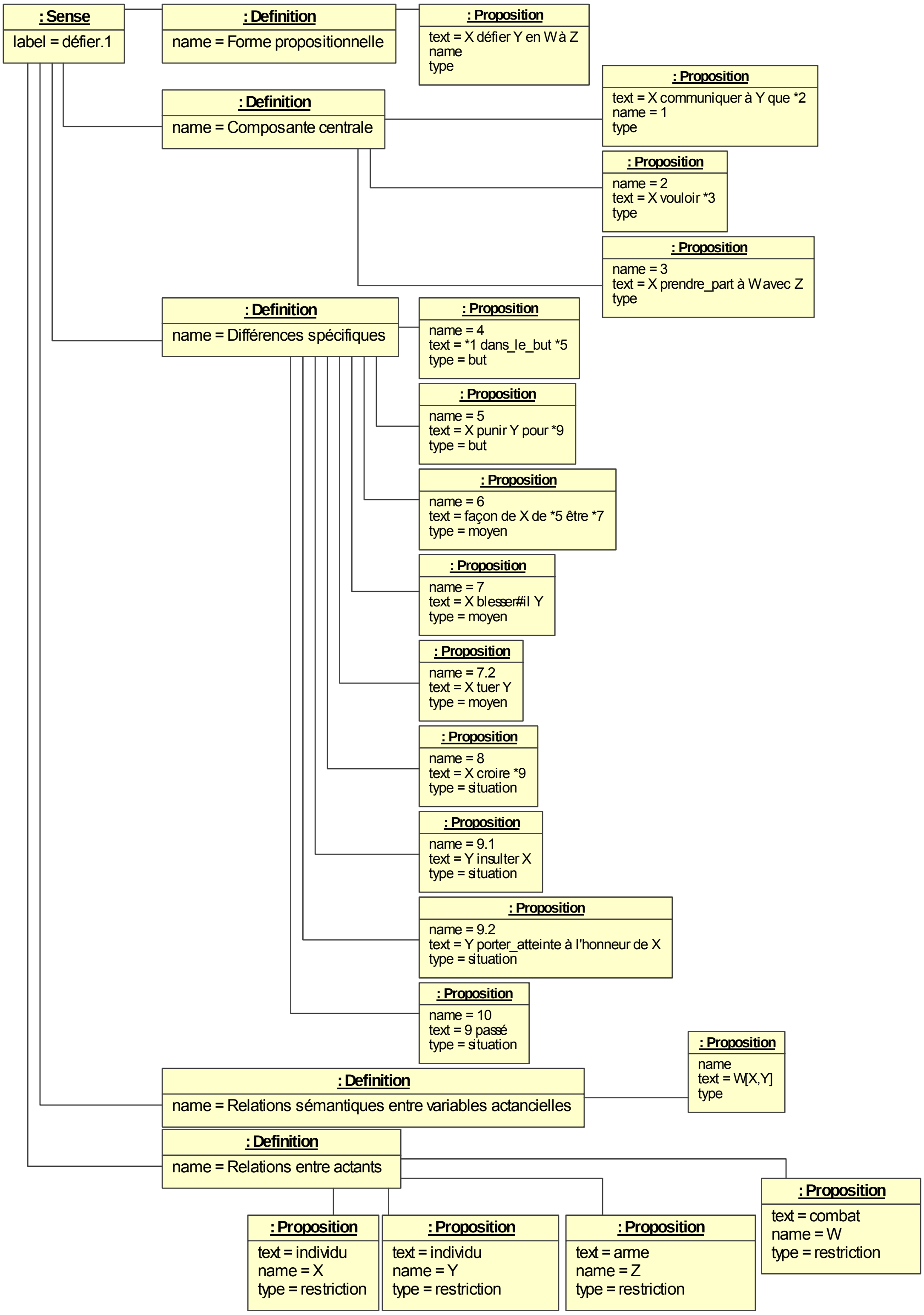
Z : arme

W : combat

Relation between actants

W[X,Y]

## A.8.3 Data within LMF


: Sense
label = défier.1
: Definition
name = Forme propositionnelle
: Proposition
text = X défier Y en W à Z
name
type
: Definition
name = Composante centrale
: Proposition
text = X communiquer à Y que *2
name = 1
type
: Proposition
name = 2
text = X vouloir *3
type
: Proposition
name = 3
text = X prendre_part à W avec Z
type
: Definition
name = Différences spécifiques
: Proposition
name = 4
text = *1 dans_le_but *5
type = but
: Proposition
name = 5
text = X punir Y pour *9
type = but
: Proposition
name = 6
text = façon de X de *5 être *7
type = moyen
: Proposition
name = 7
text = X blesser#il Y
type = moyen
: Proposition
name = 7.2
text = X tuer Y
type = moyen
: Proposition
name = 8
text = X croire *9
type = situation
: Proposition
name = 9.1
text = Y insulter X
type = situation
: Proposition
name = 9.2
text = Y porter_atteinte à l'honneur de X
type = situation
: Proposition
name = 10
text = 9 passé
type = situation
: Proposition
name
text = W[X,Y]
type
: Definition
name = Relations sémantiques entre variables actancielles
: Definition
name = Relations entre actants
: Proposition
text = combat
name = W
type = restriction
: Proposition
text = individu
name = X
type = restriction
: Proposition
text = individu
name = Y
type = restriction
: Proposition
text = arme
name = Z
type = restriction