Lexical Markup Framework (LMF) tutorial timeframe

22nd May 2006

LREC Genoa

Laurent Romary, Gil Francopoulo, Susanne Salmon-Alt, Monica Monachini

09:00-09:20	**Introduction** Laurent Romary: INRIA (FR) romary@loria.fr
09:20-10:20	**LMF data model** Gil Francopoulo: INRIA (FR) gil.francopoulo@wanadoo.fr
10:20-10:30	**questions**
10:30-10:45	**break**
10:45-11:15	**Applicability of LMF to syntactic lexicons** Susanne Salmon-Alt: ATILF-CNRS (FR) susanne.alt@loria.fr
11:15-11:25	**questions**
11:25-11:55	**LMF semantic package and mapping of existing semantic lexicons** Monica Monachini: CNR-ILC (IT) monica.monachini@ilc.cnr.it
11:55-12:05	**questions**
12:05-13:00	**extended debate based on one (or#) sample(s) provided by attendees**

LMF Introduction

Laurent Romary (INRIA)

In the first part of this tutorial we will present the rationale behind the work that has been conducted for a little bit more than two years in ISO on the LMF (Lexical Markup Framework) standard proposal. The definition of such a standard of the representation of lexical data has progressively become mandatory in the linguistic and computational linguistic community to cope with the ever increasing number of digital lexical data that are gathered and disseminated worldwide. In particular, it has proved very difficult to compare existing lexical funds, to share methodologies from one language to another and even more any kind of software[1] not to mention APIs. The worse aspect in the picture is probably the potential risk that data are lost as time goes on, because the lexical formats used at various places are not maintained, nor documented.

Following the seminal work that had been carried out in ISO committee TC 37/SC 4 on the representation of terminological data, it was clear that the variety of lexical application could not boil down to an international standard that would one propose one representation format, for instance in XML. As a matter of fact, ISO 16642 (TMF – Terminological Markup Framework) contains the idea that a given family of linguistic structures can be modelled by combining two elementary components:

- A meta-model that reflect the general organisation of data shared across several application;
- Data categories, which can be used to further instantiate to meta-model to make it fit the needs of a specific applicative context.

Such a modelling strategy resolves the difficulty of:

a) maintaining, through the comparison of specifications, a high level interoperability between applications

b) at the same time providing the final-end user with the necessary flexibility to represent his own data.

When it was contemplated to move the TMF methodology to lexical data, several difficult points were immediately identified that would have to be dealt with in the future LMF proposal:

- Computerized lexica may have a quite variable level of complexity, ranging from simple word list to the complex syntactic-semantic structures needed for machine translation: it was felt necessary to think of a highly modular framework so that one implementer would not have to master the whole standard even before he starts implementing the most basic application;
- Even for a given representation level (especially syntax), there are quite a few discrepancies from one theoretical background to another concerning the way the same information should be encoded: being able to identify precisely the corresponding differences is likely to improve our scientific knowledge in those domains;

[1] With the possible exception of Shoobox in the linguistic domain.

- Multilingualism introduces another dimension of complexity as the kind of representations needed for a given phenomena may differ between languages (e.g. root based vs. lemma based organisation of lexica in Semitic vs. Romance languages).

As an introduction to those issues, we will show how LMF can be seen as an abstraction from existing representations (e.g. the Print Dictionary chapter of the TEI) or can be used to represent in a very simple way basic lexical data such as those needed for inflectional lexica (as in the Morphalou project; the very first implementation of LMF).

As a whole, LMF should be seen by the community as a tool for modelling one's own lexical data, with the possible result that people will provide useful feedback on the usability and needed evolution of the standard project.

LMF data model

Gil Francopoulo
gil.francopoulo@wanadoo.fr
INRIA-Loria (France)
22nd May 2006

Summary

- 1 LMF project
- 2 History and roadmap
- 3 Scope
- 4 Definitions in LMF-rev-9
- 5 General principles
- 6 UML
- 7 LMF structural data model

1 LMF project

- Work started in Summer 2003 by a new work item proposal issued by the US delegation
- Fall 2003: the French delegation issued a technical proposition for a data model dedicated to NLP lexicons
- Beginning of 2004: ISO-TC37/SC4 decided to form an ISO project with:
  - Nicoletta Calzolari (IT) as convenior
  - two editors:
    Gil Francopoulo (FR)
    Monte George (US)

2 History and roadmap

- **In ISO process, everything is centered around the normative document: ISO-24613**
- **Status evolution: NWIP => WD => CD => DIS => FDIS => IS**

- **In 2 years and a half, nine versions has been written, dispatched (to the National delegations nominated experts), commented and discussed in various ISO technical meetings**
- **+ one paper in COLING-2004**

- **Situation today:**
  - LMF document is a « committee draft » document (60 pages)
  - 29th March 2006: call for a CD ballot based on LMF-rev-9
  http://lirics.loria.fr/doc_pub/LMF%20rev9%2015March2006.pdf

- **If votes are ok, LMF document will be a « draft for an international standard » document**
- **Target: IS (= published standard) in 2008**

3 Scope

- Range of topics, LMF is intended for.
  => MRD + NLP lexicons
  => all MRD and all NLP applications
  => all languages
  => small and large scale lexicons
  => monolingual, bilingual, multilingual

4 Definitions in LMF-rev-9

- In order to avoid ambiguities, technical terms are clearly defined (6 pages).
- E.g.
  - **word** = linguistic unit composed of at least a part of speech and a lemma
  - **lemma** = lemmatised form = conventional form chosen to represent words or MWE
  - **form** = sequence of morphemes and affixe forms
  - etc.

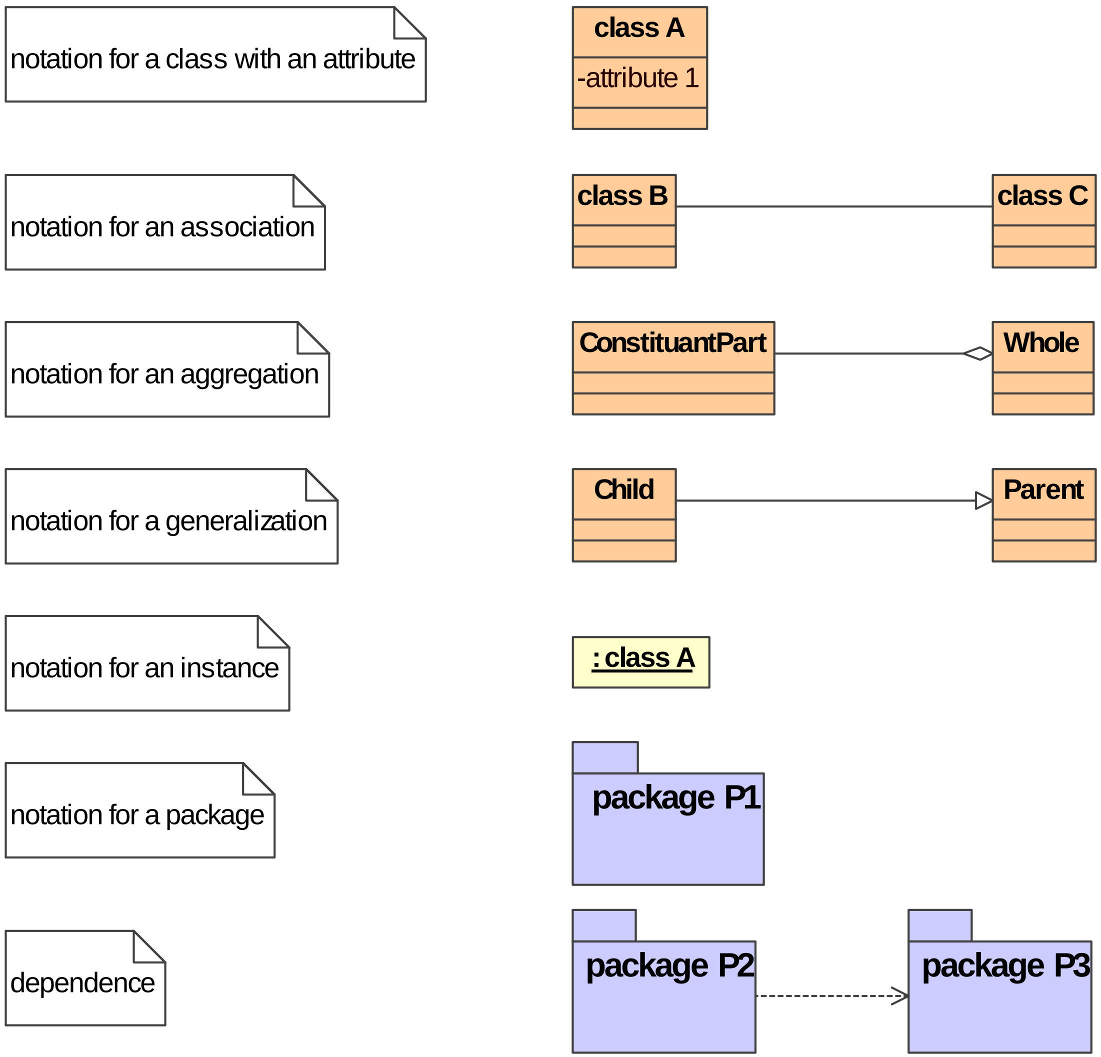
5 General principles

- LMF is a structural data model expressed by a set of Unified Modeling Language (UML) packages. Sometimes the term « meta-model » is used instead of « structural ».
- LMF is a high level specification based on constants that are defined in other standards
- Each package contains classes
- Each class is specified by:
  - a name
  - an English text describing its usage
  - an UML specification for linking with other classes
- But the class attributes are not defined. Only a list of examples is given. The attributes are to be taken from the data category registry (see next slide).
- The values are either constants or free strings.

General principles (cont.)

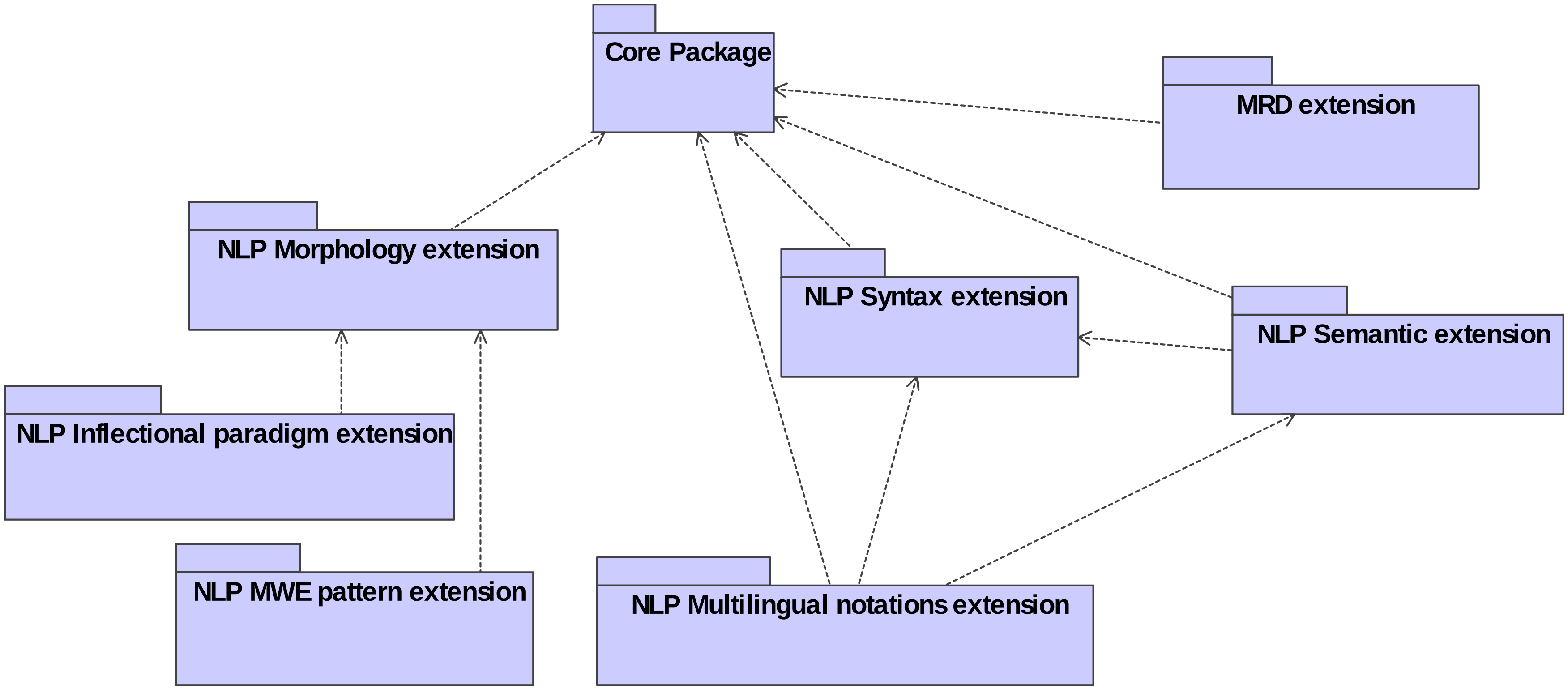
- The constants are:
  - character codes **(ISO/IEC 10646 i.e. Unicode)**
  - language codes **(ISO-639)**
  - script codes **(ISO-15924)**
  - country codes **(ISO-3166)**
  - dates **(ISO-8601)**
  - data category registry **(rev ISO-12620)** work in progress to define linguistic constants like /part of speech/, /feminine/ or /transitive/

6 UML (subset of UML we use)

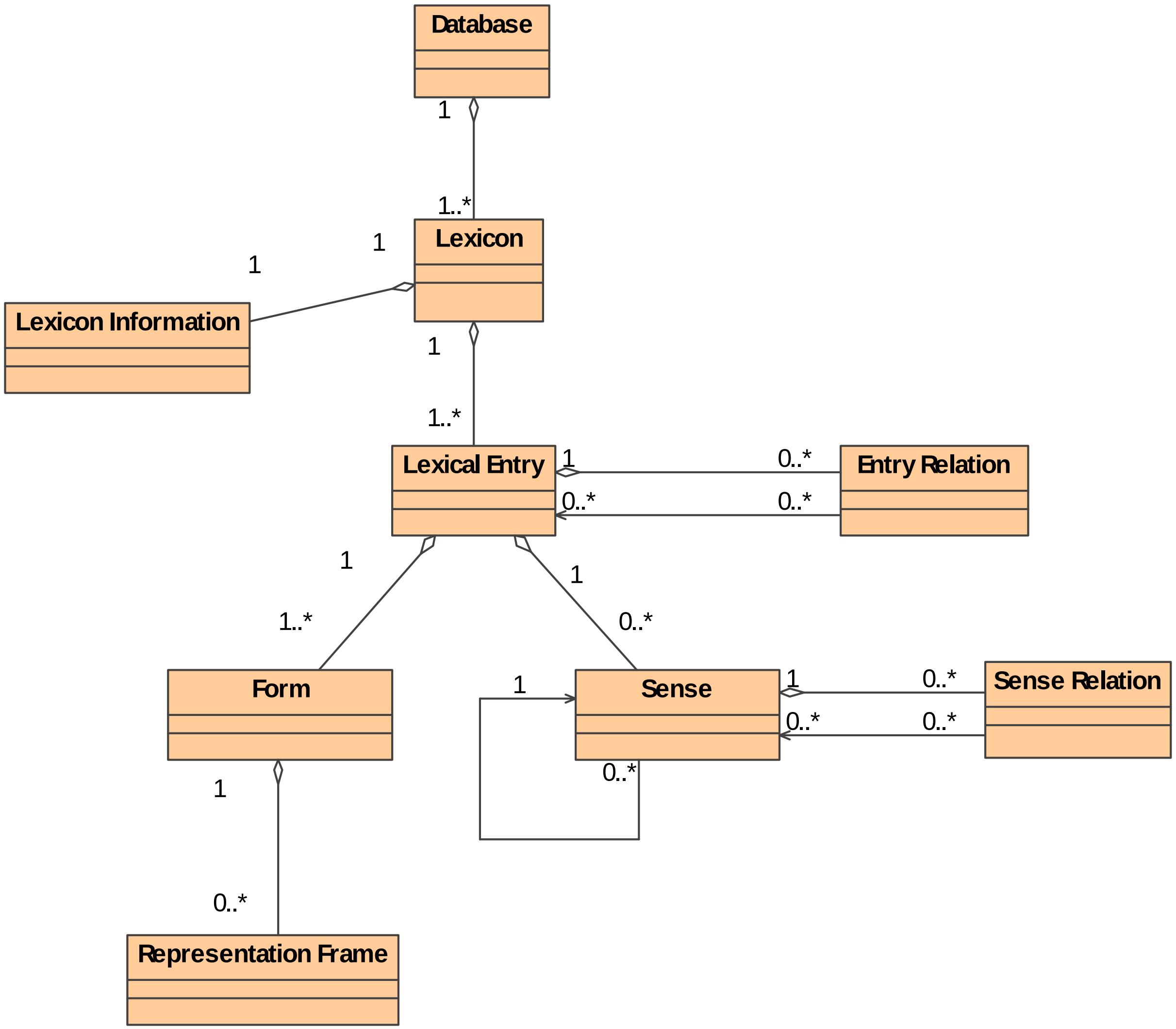


7 LMF structural data model

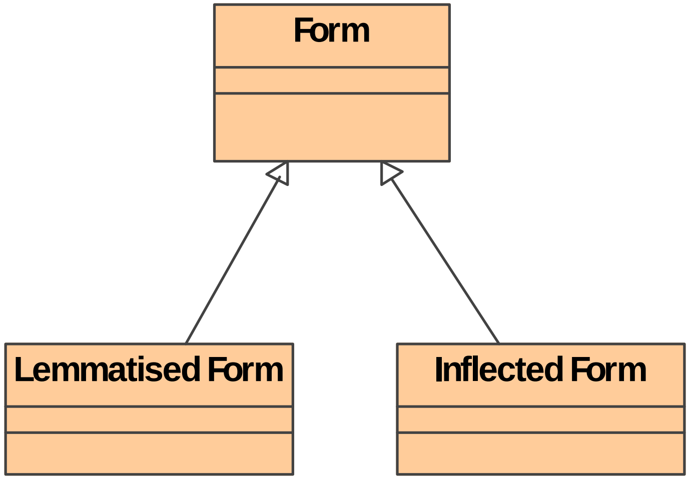
- One core package and 7 extensions



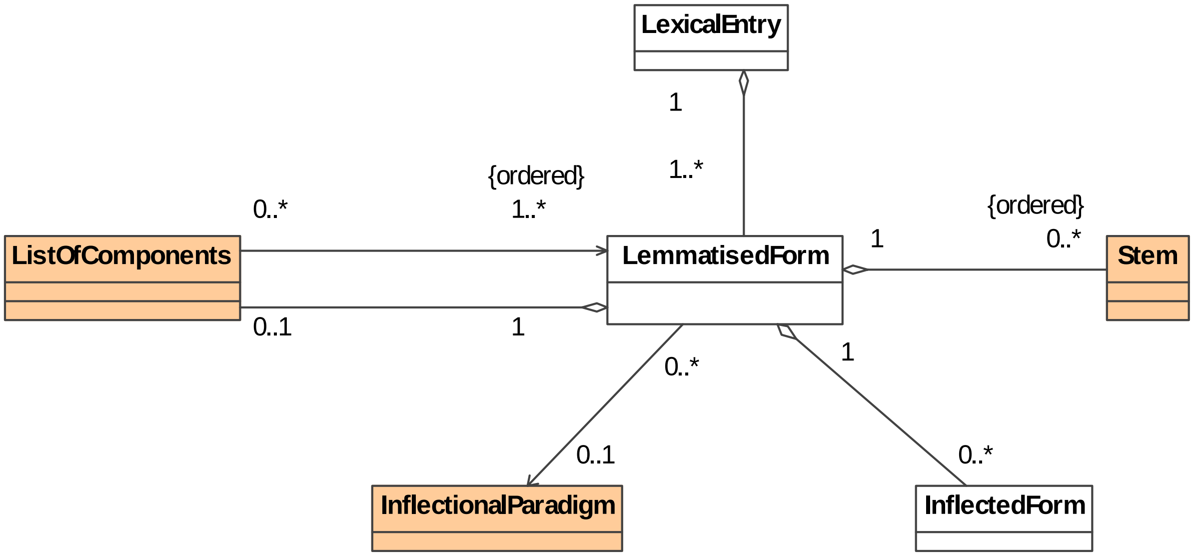
Core package



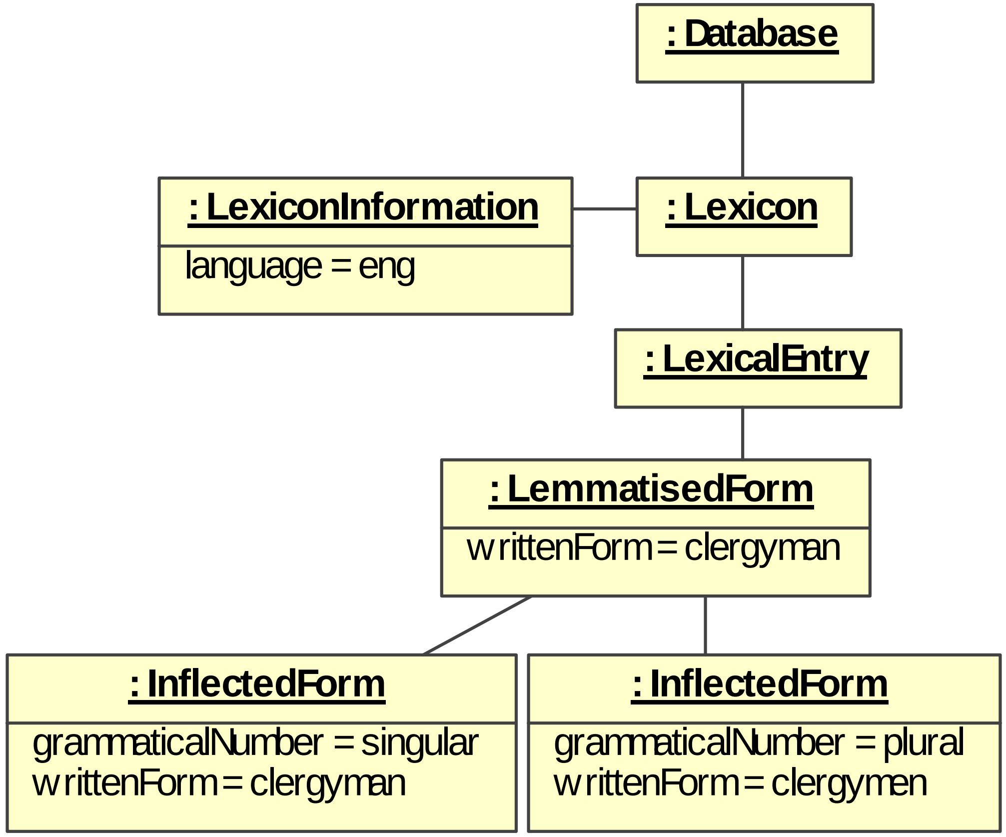
Hierarchy of forms



Package for extensional morphology



Example of clergyman



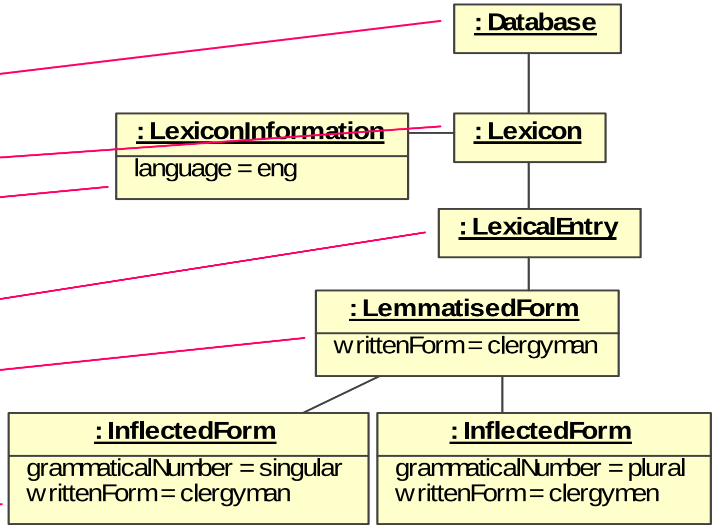
Transcoding UML class diagram into XML => very easy

- Every UML class is transcoded as an XML element
- Every aggregation is transcoded as an XML content inclusion with the right cardinality
- Every association that is not an aggregation is an XML reference: that is an IDREF, or IDREFS depending on the cardinality. As a consequence, the class that is referenced must hold an ID.
- Class adornement is implemented by a DC element.

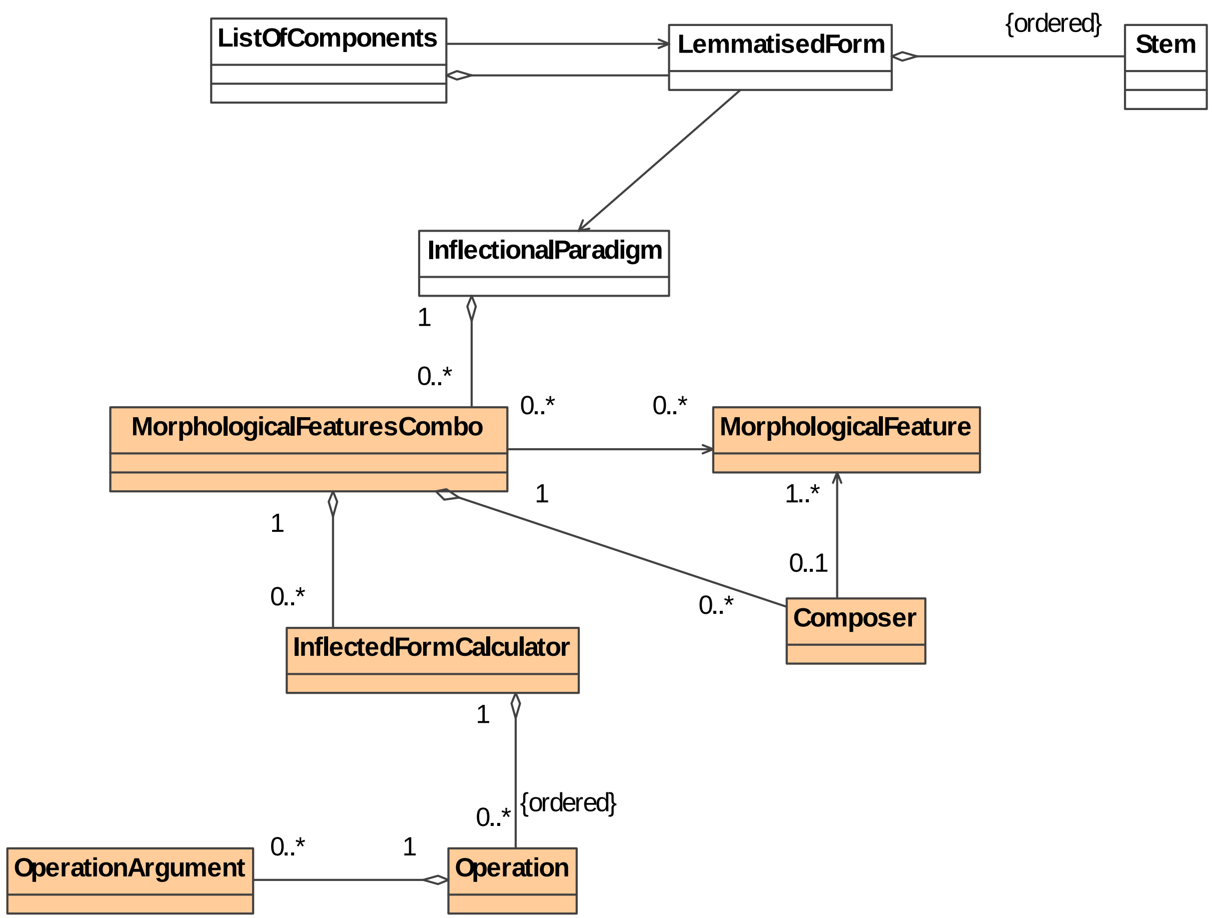
Equivalent representation in XML

- Following the LMF-rev-9 DTD:

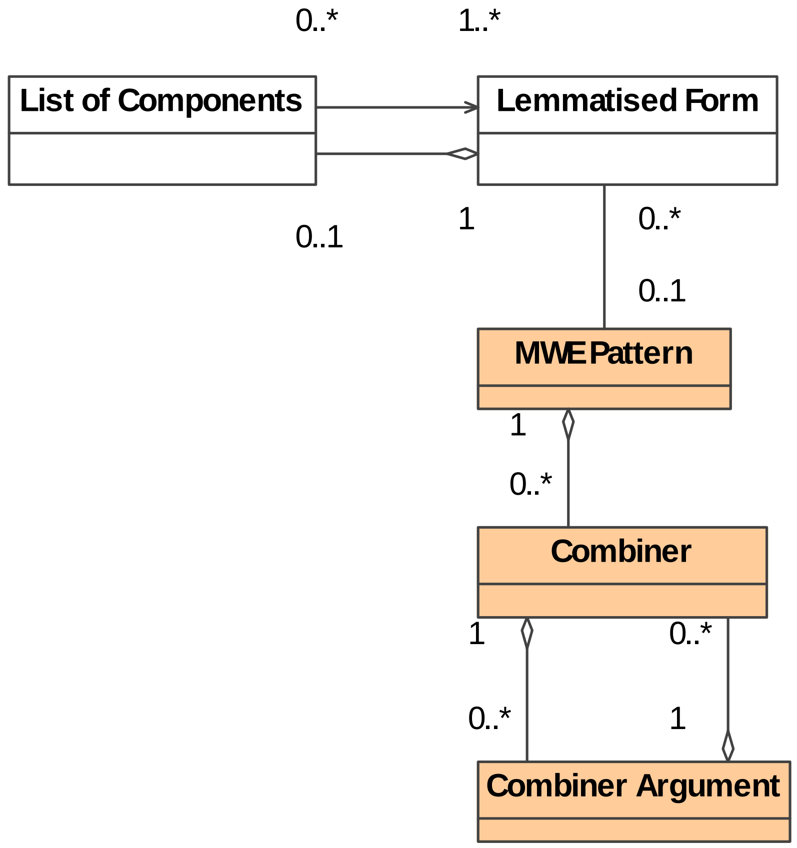
```
<DataBase>
<Lexicon>
        <LexiconInformation>
                <DC att="language" val="eng"/>
        </LexiconInformation>
        <LexicalEntry>
        <LemmatisedForm>
                <DC att="writtenForm" val="clergyman"/>
                <InflectedForm>
                        <DC att="grammaticalNumber" val="singular"/>
                        <DC att="writtenForm" val="clergyman"/>
                </InflectedForm>
                <InflectedForm>
                        <DC att="grammaticalNumber" val="plural"/>
                        <DC att="writtenForm" val="clergymen"/>
                </InflectedForm>
        </LemmatisedForm>
        </LexicalEntry>
</Lexicon>
</DataBase>
```



Package for inflectional paradigm



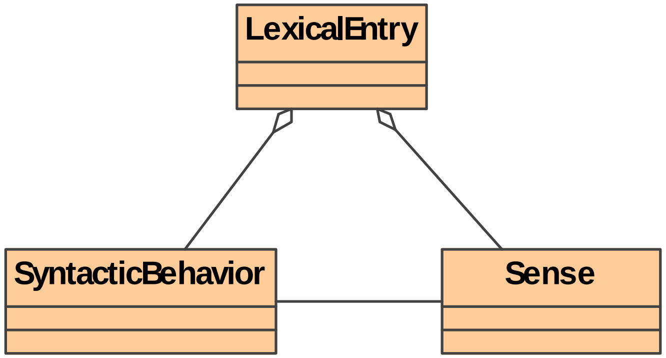
Package for Multiword expressions



Syntax/semantic connexion

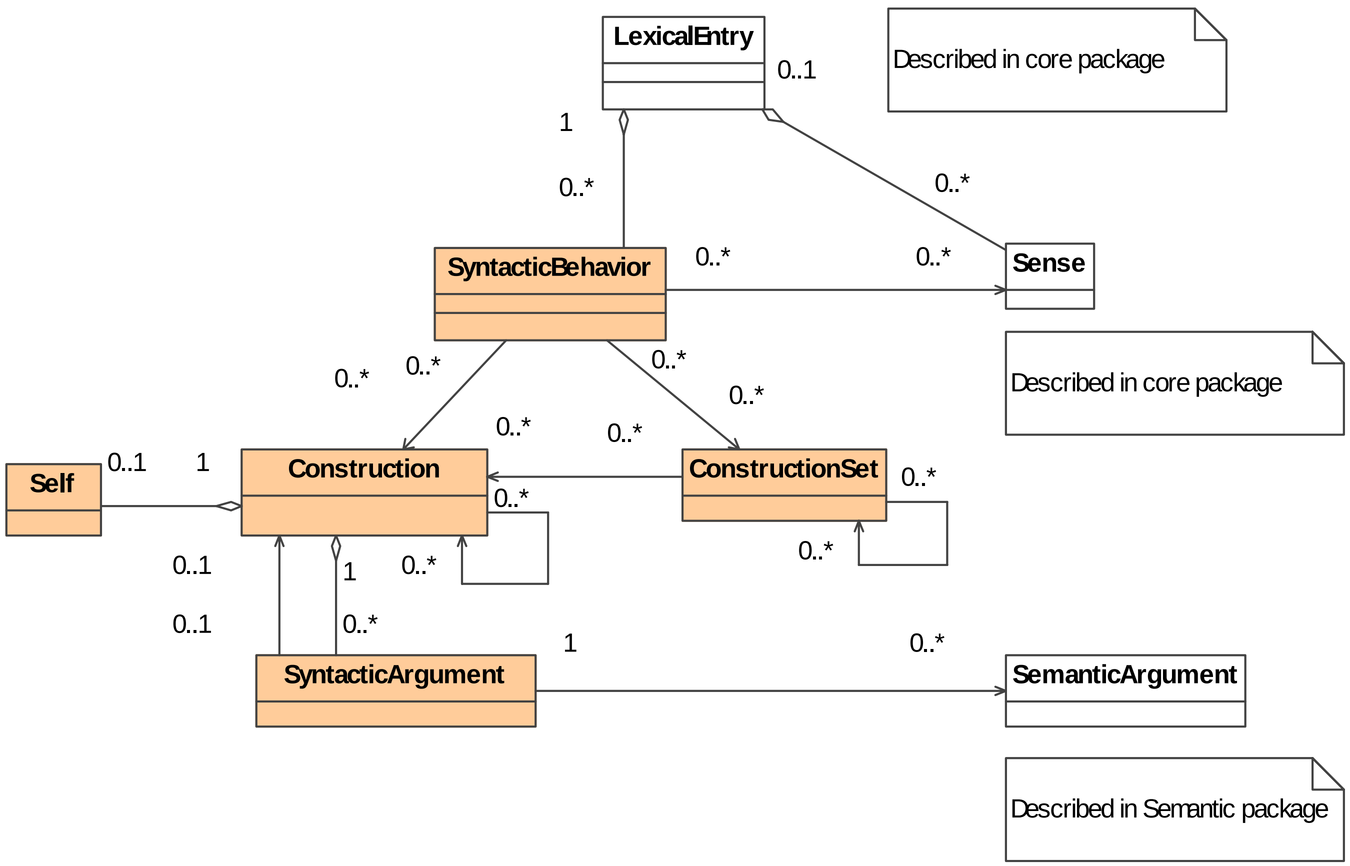
- Connexion between:
  - a lexical entry (= the presence of a word)
  - the fact that this word has one or many syntactic behaviors
  - its senses (one sense for a monosemous and several for a polysemous word)

  => a triangle



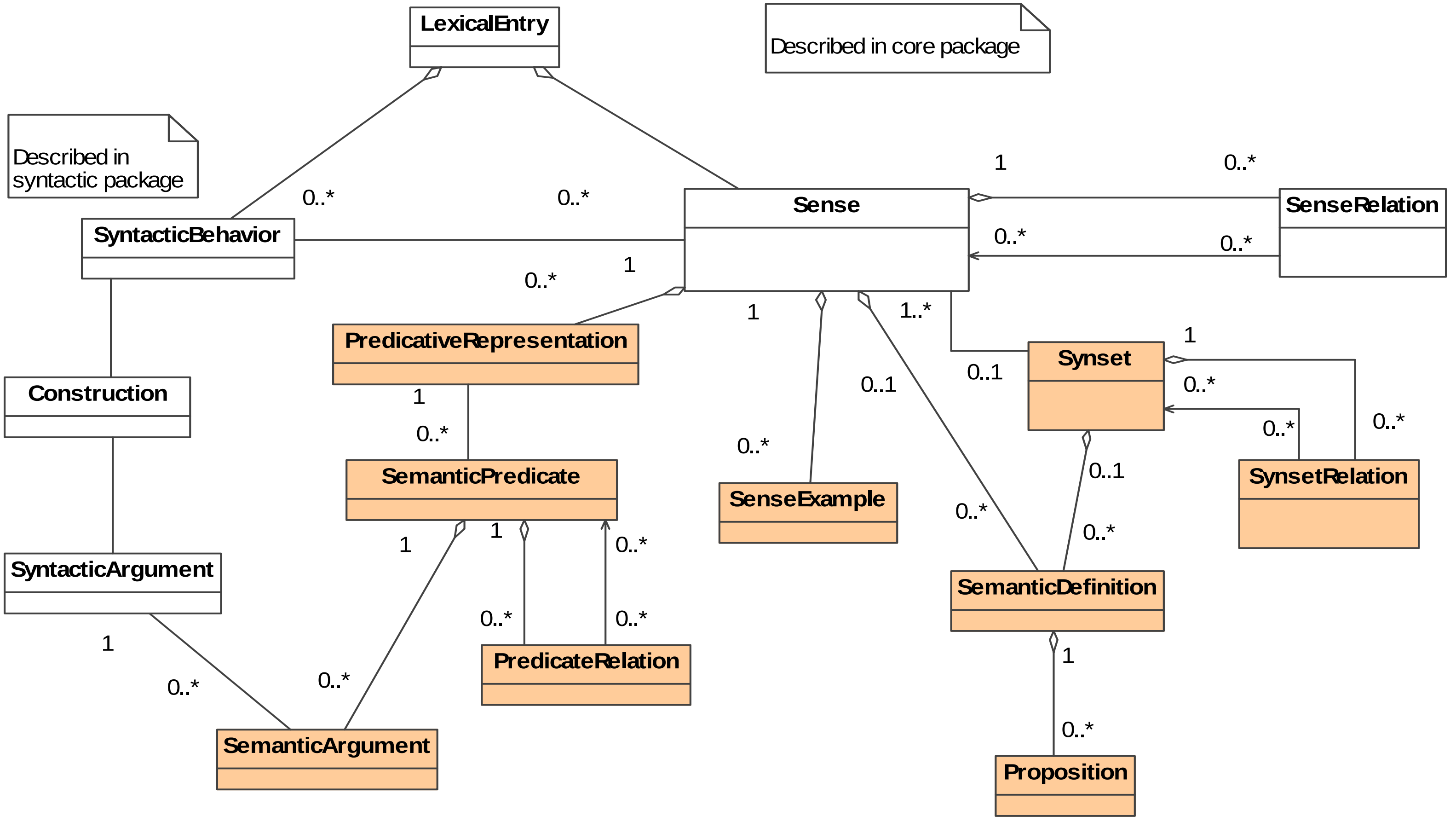
Data model: syntax

More details and examples will be presented by Susanne Salmon-Alt



Data model: semantics

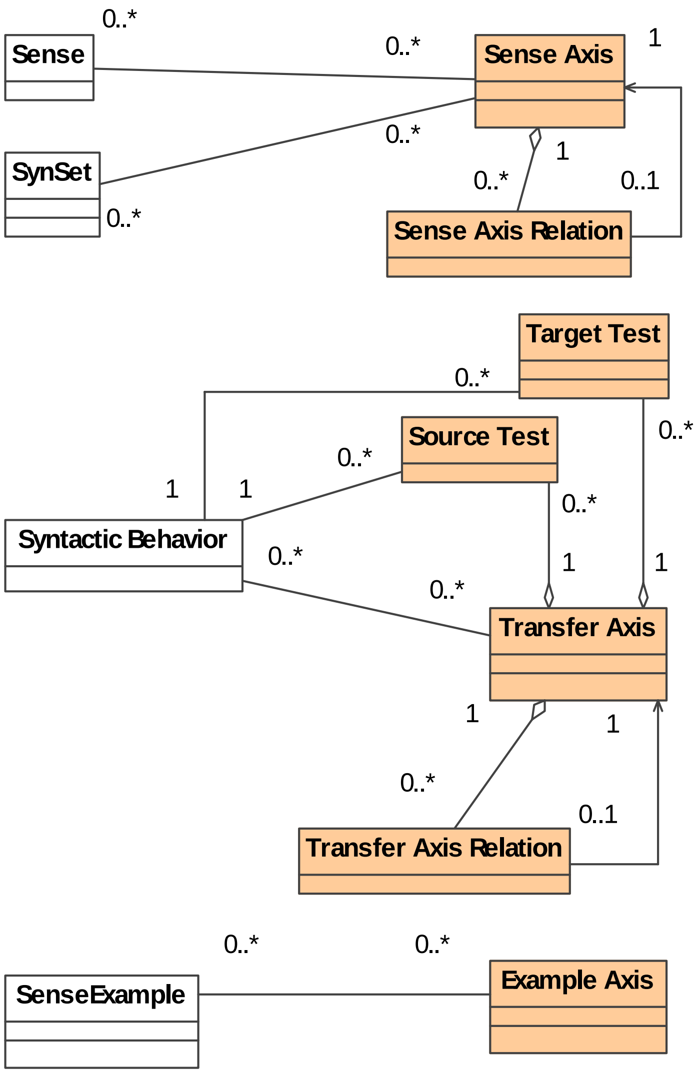
More details and examples will be presented by Monica Monachini



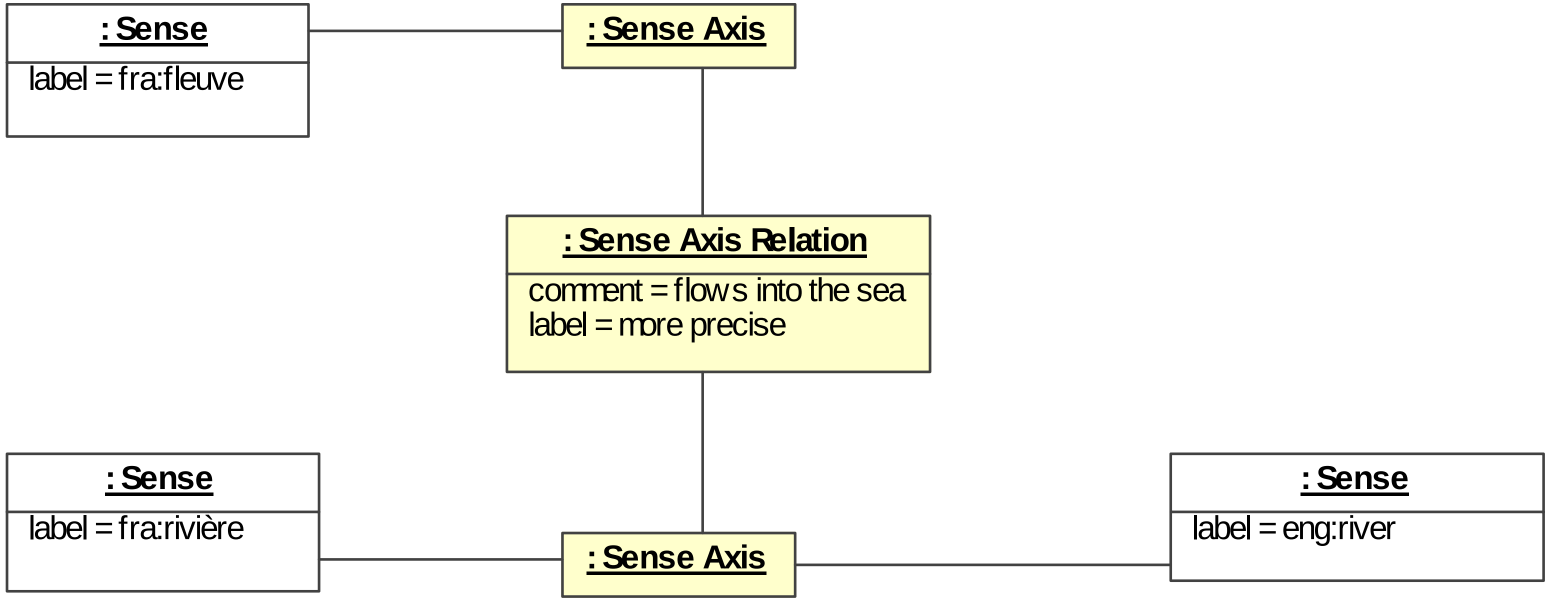
Package for multilingual notations

- For both interlingual pivots and transfer approach:
  - Sense+Synset (of different languages) may be linked by SenseAxis
  - SyntacticBehavior (of different languages) may be linked by TransferAxis
- Possibility to share or to duplicate Axis
- Possibility to add sourceTest or targetTest
- Possibility to link Examples (from different languages)

Multilingual notation data model



Multilingual notation example



Last slide

- **Acknowledgements:**
  The work presented here is partially funded by the EU eContent-22236 LIRICS project and partially by the French TECHNOLANGUE program.

- **Future readings:**
  - LMF-revision-9
  http://lirics.loria.fr/doc_pub/LMF%20rev9%2015March2006.pdf
  - for monolingual aspects:
  Lexical Markup Framework (LMF)
  in LREC2006 main conference
  - for multilingual aspects:
  LMF for multilingual, specialized lexicons
  in LREC2006 workshop: multilingual, specialized lexicons

Applicability of LMF to Syntactic Lexicons

Susanne Salmon-Alt (ATILF-CNRS)

1. The normative proposal

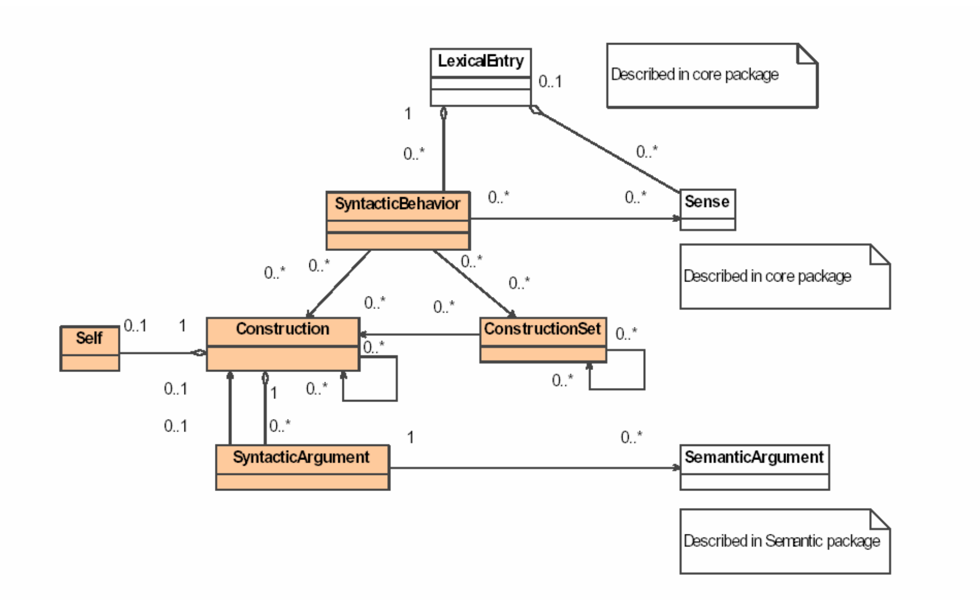


Figure 1 : The syntactic extension (LMF v9)

Components :

- SyntacticBehaviour
- ConstructionSet
- Construction
- SyntacticArgument
- Self
- Composition

Relations :

- Aggregation
- Association

2. Samples : different views on syntactic data

2.1. Le*fff* 2

Sagot Benoît, Clément Lionel, de La Clergerie Éric et Boullier Pierre. The Le*fff* 2 syntactic lexicon for French: architecture, acquisition, use. *LREC 06*. 2006.

aimer	v	[pred='aimer_____1<suj:(sn),obj:sinf>',cat=v,@CtrlSubj,@W]
aimer	v	[pred='aimer_____1<suj:(sn),obj:sn,att:sa>',cat=v,@AAObj,@W]
aimer	v	[pred='aimer_____1<suj:(sn),obj:sn|scompl|qcompl>',cat=v,@SCompSubj,@W]
aimer	v	[pred='aimer_____1<suj:(sn),obj:à-sinf>',cat=v,@CtrlSubjA,@W]

=> Extensional view : explicit description of every syntactic construction for every verb

2.2. VerbNet

Karin Kipper, Hoa Trang Dang, Martha Palmer. Class-Based Construction of a Verb Lexicon. *AAAI-2000 Seventeenth National Conference on Artificial Intelligence*, Austin, TX, July 30 - August 3, 2000.

admire-31.2
WordNet Senses : **love**(1 2 3)
Thematic Roles : Cause[] Experiencer[+animate]

Frames

Attribute Object Possessor-Attribute Factoring Alternation ()
"I admired the honesty in him"
Experiencer V Cause Prep(in) Oblique
emotional_state(E,Emotion,Experiencer) in_reaction_to(E,Cause)

Basic Transitive ()
"The tourists admired the paintings"
Experiencer V Cause
emotional_state(E,Emotion,Experiencer) in_reaction_to(E,Cause)

Possessor Subject Possessor-Attribute Factoring Alternation ()
"I admired him for his honesty"
Experiencer V Oblique Prep(for) Cause
emotional_state(E,Emotion,Experiencer) in_reaction_to(E,Cause)

Sentential Complement ()
"The children liked that the clown had a red nose"
Experiencer V Cause[+sentential]
emotional_state(Emotion,Experiencer) in_reaction_to(E,Cause)

Verbs in same (sub)class

[admire, adore, appreciate, cherish, enjoy, esteem, exalt, fancy, favor, idolize, like, love, miss, prize, respect, relish, revere, savor, stand, support, tolerate, treasure, trust, value, venerate, worship, abhor, deplore, despise, detest, disdain, dislike, distrust, dread, envy, execrate, fear, hate, lament, loathe, mourn, pity, regret, resent, rue, believe, suffer]

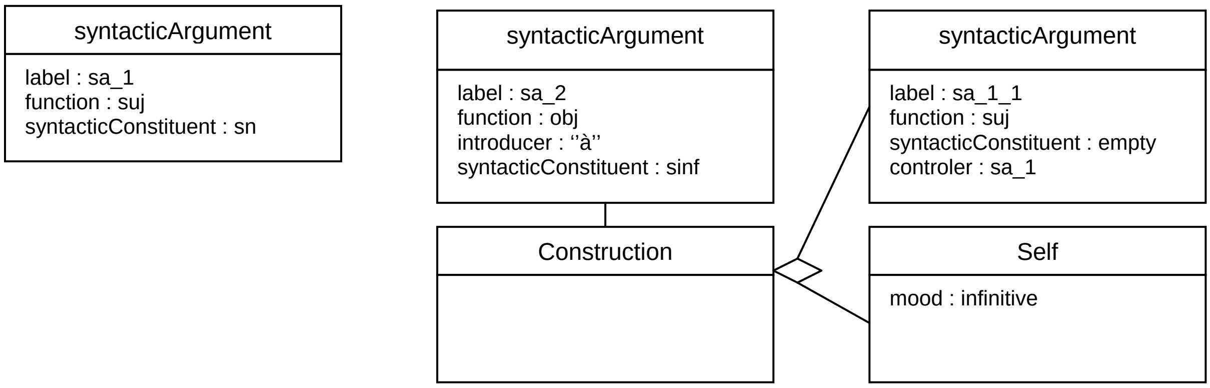
=> Intensional view : factorized description of syntactic constructions for verb classes

3. Instanciation of LMF proposals

3.1. Describing Syntactic Arguments

aimer   v   [pred='aimer_____1<suj:(sn),obj:à-sinf>',cat=v,@CtrlSubjA,@W]

Ex :   Max aime à regarder la lune.

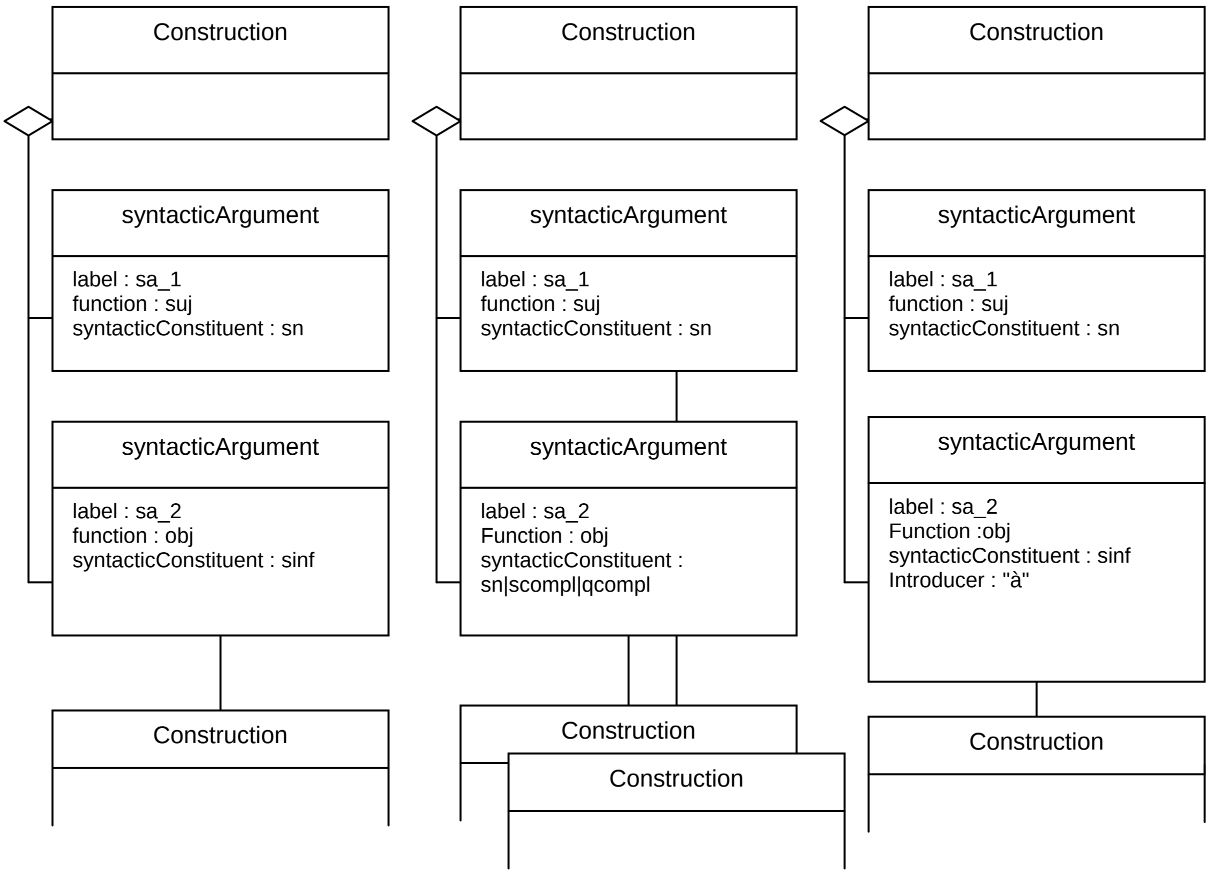


Topics for discussion

- candidates for standardized syntactic functions and constituents
  - cf. EAGLES, Carroll et al. 1998 ?
- need of splitting functions and constituents
- sentential complements as embedded syntactic constructions with own arguments
- encoding of empty constituents (e.g. subjects in infinitival phrases, pro-drop in romance languages)
- encoding of subject controlling
- specify the use of *label* : id, thematic role, numbering of positions, ...
- terminology : *Self* vs. *syntacticPredicate*

3.2. Describing Syntactic Constructions

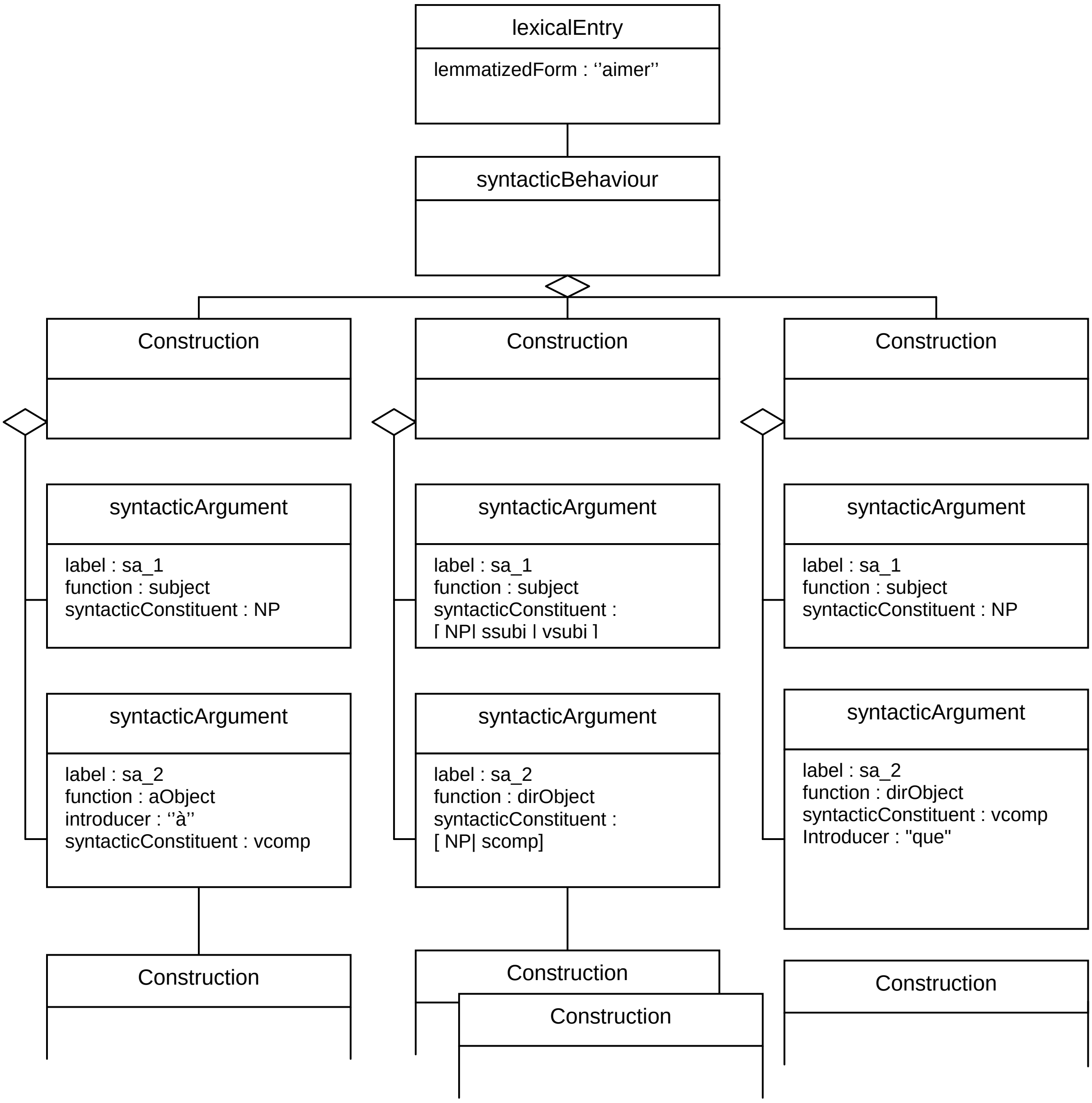
aimer	v	[pred='aimer_____1<suj:(sn),obj:sinf>',cat=v,@CtrlSubj,@W]
aimer	v	[pred='aimer_____1<suj:(sn),obj:sn|scompl|qcompl>',cat=v,@SCompSubj,@W]
aimer	v	[pred='aimer_____1<suj:(sn),obj:à-sinf>',cat=v,@CtrlSubjA,@W]
aimer	v	[pred='aimer_____1<suj:(sn),obj:sn,att:sa>',cat=v,@AAObj,@W]



Topics for discussion

- level of granularity for constructions
  - deep vs surface constructions, alternations and transformations
  - how many constructions in the example ?
- factorizing argument properties
  - e.g. syntactic constituents (cf. construction 2)
- ordering of arguments

3.3. The extensional view : describing extensionally the constructions of a verb



3.4. The intensional view : factorizing the description of constructions

- build a (potentially hierarchical) database of constructions (using principles of 3.2) and let entries refer to those constructions or classes of constructions
- example databases : Lefff2 (intensional version), VerbNet

Attribute Object Possessor-Attribute Factoring Alternation ()
"I admired the honesty in him"
Experiencer V Cause Prep(in) Oblique
emotional_state(E,Emotion,Experiencer) in_reaction_to(E,Cause)

Figure 2 : One of the VerbNet frames for "love"

```
<FRAME>
    <DESCRIPTION descriptionNumber="2.13.2" primary="NP-PP" secondary="Predicate Object, Possessor-PP"
xtag=""/>
    <EXAMPLES>
      <EXAMPLE><![CDATA["I admired the honesty in him"]]></EXAMPLE>
    </EXAMPLES>
    <SYNTAX>
      <NP value="Experiencer"><SYNRESTRS/></NP>
      <VERB/>
      <NP value="Predicate"> <SYNRESTRS/></NP>
      <PREP value="in"><SELRESTRS/></PREP>
      <NP value="Theme"><SELRESTRS/></NP>
    </SYNTAX>
    <!-- ... -->
 </FRAME>
```

Figure 3 : XML encoding of syntactic information of the VerbNet frame for "love"

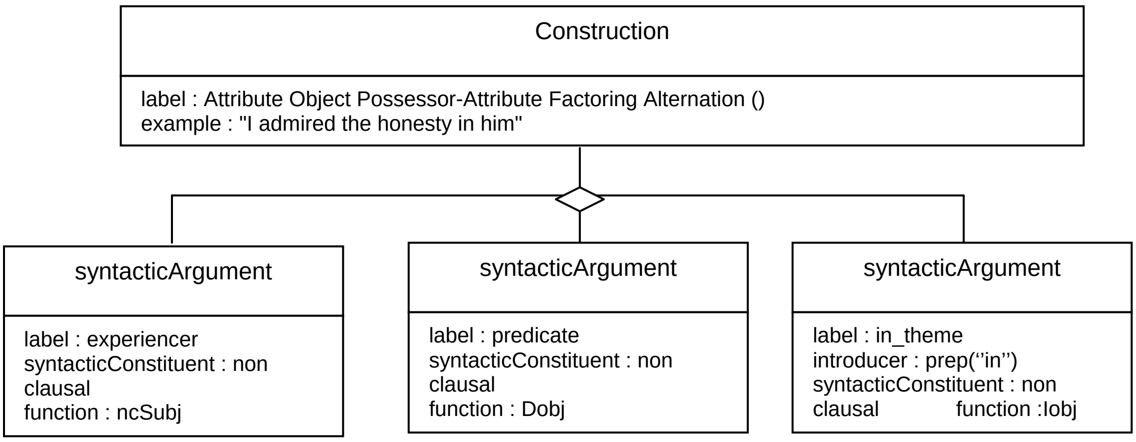


Figure 4 : LMF encoding of syntactic information of the VerbNet frame for "love"
(after merging arguments and inference of syntactic functions, cf. Crabbé et al. 2006)

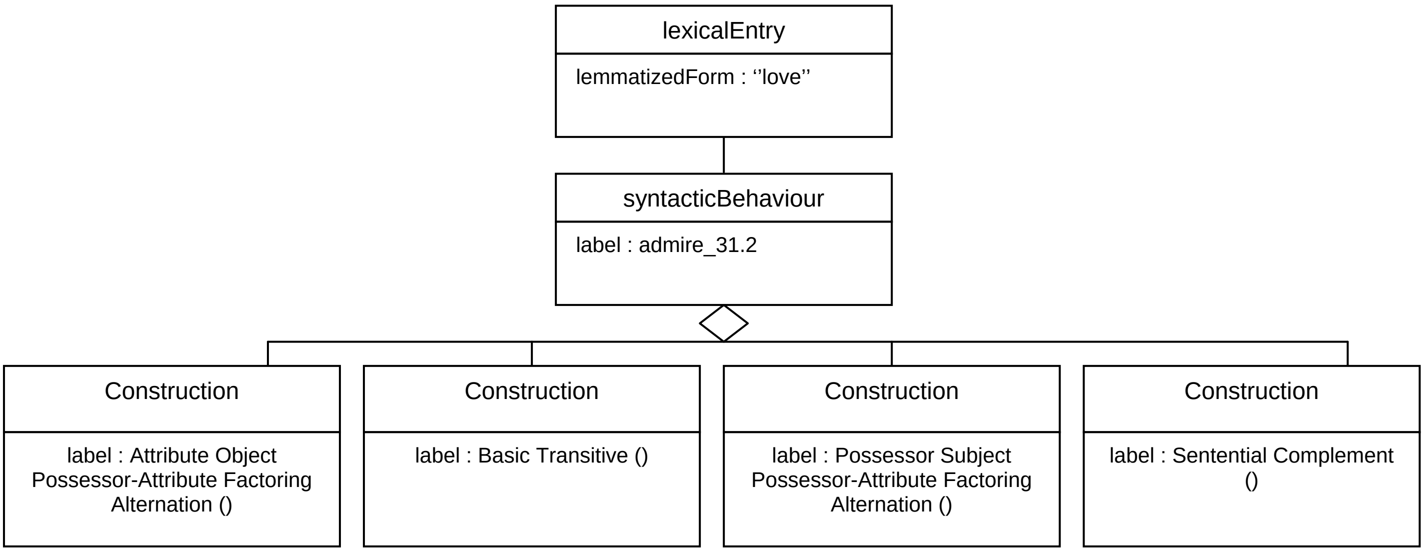


Figure 5 : An LMF lexical entry for VerbNet – constructions are referred to rather than described extensionally

```
<lexicalDatabase>
   <!-- syntactic database -->
   <constructionSet label="admire_31.2" id="cS_1">
      <construction label="Attribute Object Possessor-Attribute Factoring Alternation ()" id="c_1">
         <example>I admired the honesty in him.</example>
         <syntacticArgument label="experiencer" function="ncSubj" syntacticConstituent="non_clausal"/>
        <!-- <self/>-->
        <syntacticArgument label="predicate" function="Dobj" syntacticConstituent="non_clausal"/>
        <syntacticArgument label="theme" function="Iobj" syntacticConstituent="non_clausal" introducer="prep(in)"/>
      </construction>
      <construction> <!-- further constructions--> </construction>
   </constructionSet>
   <constructionSet> <!-- further constructionSets--> </constructionSet>
   <!-- lexical entries  -->
   <lexicalEntry>
        <lemmatizedForm>love</lemmatizedForm>
        <syntacticBehaviour>
           <constructionSet target="id(cS_1)"/>
        </syntacticBehaviour>
   </lexicalEntry>
   <lexicalEntry>
      <lemmatizedForm>admire</lemmatizedForm>
      <syntacticBehaviour>
         <constructionSet target="id(cS_1)"/>
      </syntacticBehaviour>
   </lexicalEntry>
   <lexicalEntry> <!-- further lexicalEntries--> </lexicalEntry>
</lexicalDatabase>
```

Figure 6 : LMF XML encoding : syntactic database and lexical entries

4. Some perspectives for discussion

- Is the terminology clear and cross-theoretically understandable ?
- Is the UML schema easily understandable ?
- Is the relation between the extensional (i.e. full list of constructions for each entry) and intensional (i.e. factorising of constructions) views clear ?
- Which (tentative) lists of needed data categories (e.g. syntactic functions and constituents) ?

LMF semantic package and mapping of existing semantic lexicons

Monica Monachini
monica.monachini@ilc.cnr.it
ILC-CNR (Italy)

22nd May 2006

Summary of presentation

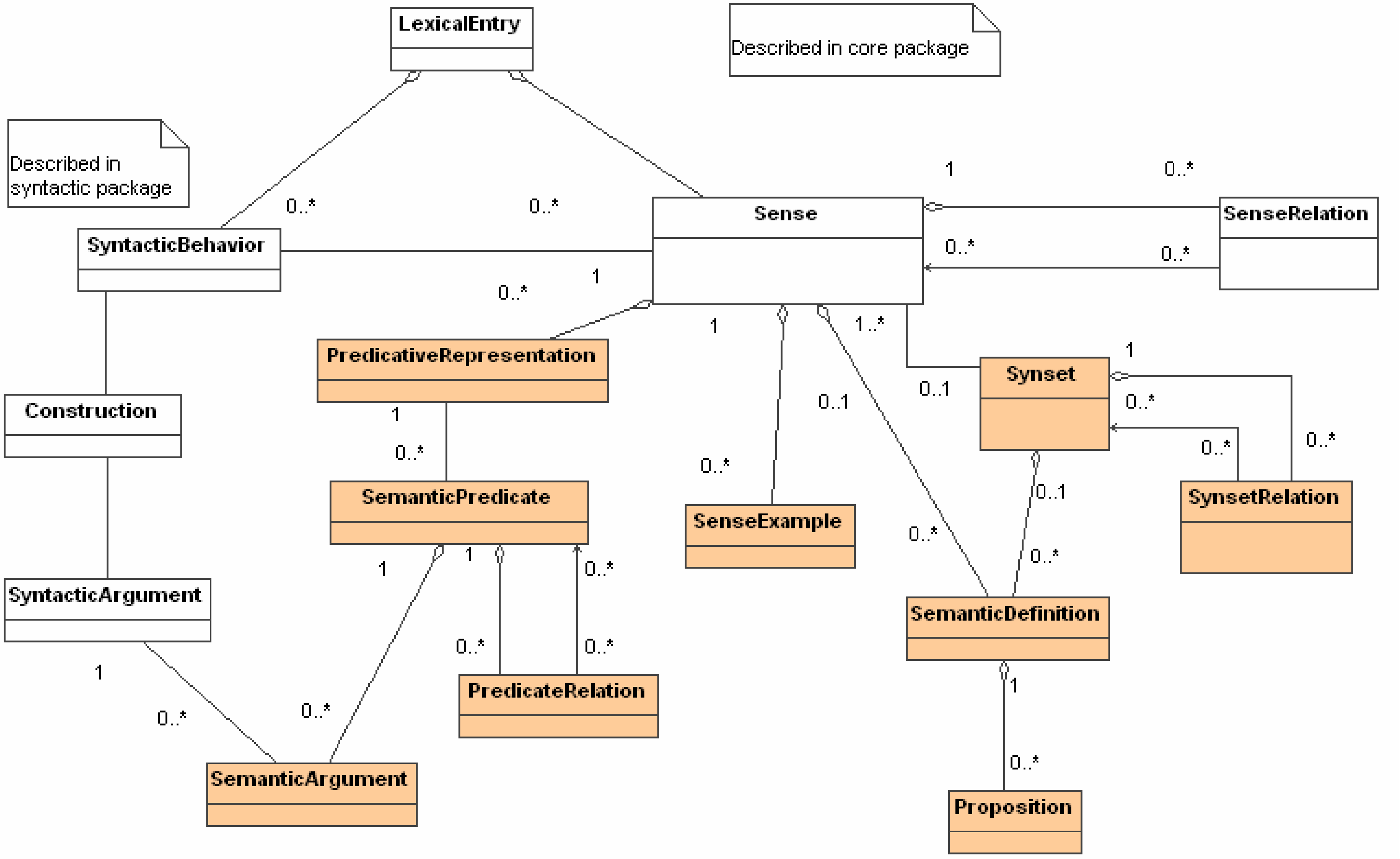
- 1 Scope of the semantic extension
- 2 The semantic data model
- 3 Description of Semantic Lexical Classes
  - The « power» of Predicate
  - Connection of syntactic and semantic layers
- 4 LMF and existing semantic lexicons
  - PAROLE-SIMPLE-CLIPS
  - WordNet
  - FrameNet

3 Scope

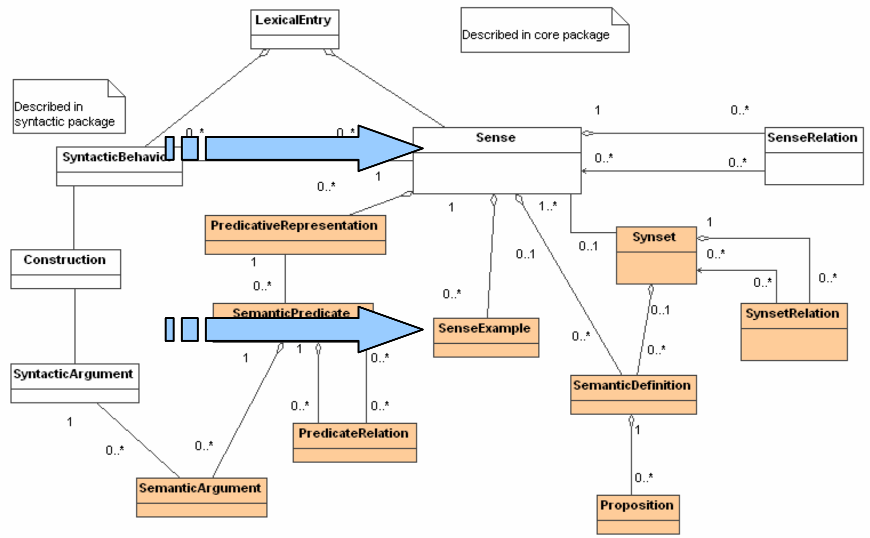
The scope of the NLP semantic extension package is to:

- describe one sense and its relations with other senses of the same language
- due to the intricacies of syntax and semantics, to represent the connection to syntax

Data model: semantics

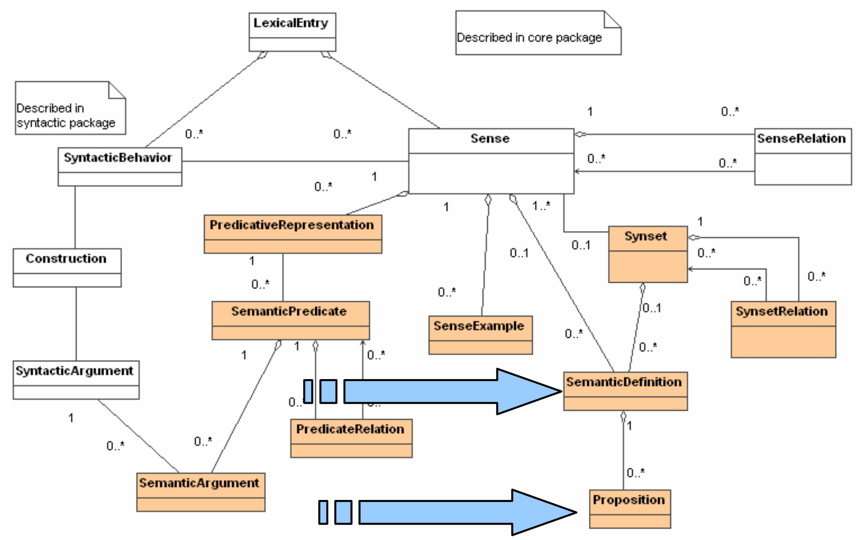


Description of Classes



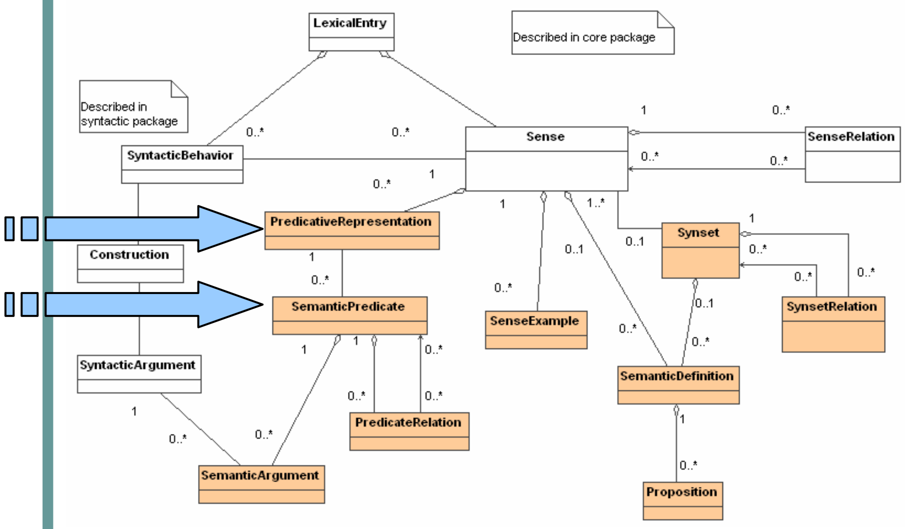
class name	Example of attributes	comment
Sense	Dating Style frequency geography Animacy	Sense is described in the core package. This class is not shared among two different lexical entries
Sense Example	Text source Language	It is used to describe usages of the particular meaning of the Sense element. For instance, a lexicon in Bambara can hold examples expressed with usual orthography and examples with tones added, in order to permit beginners to understand and pronounce the example.

Description of Classes



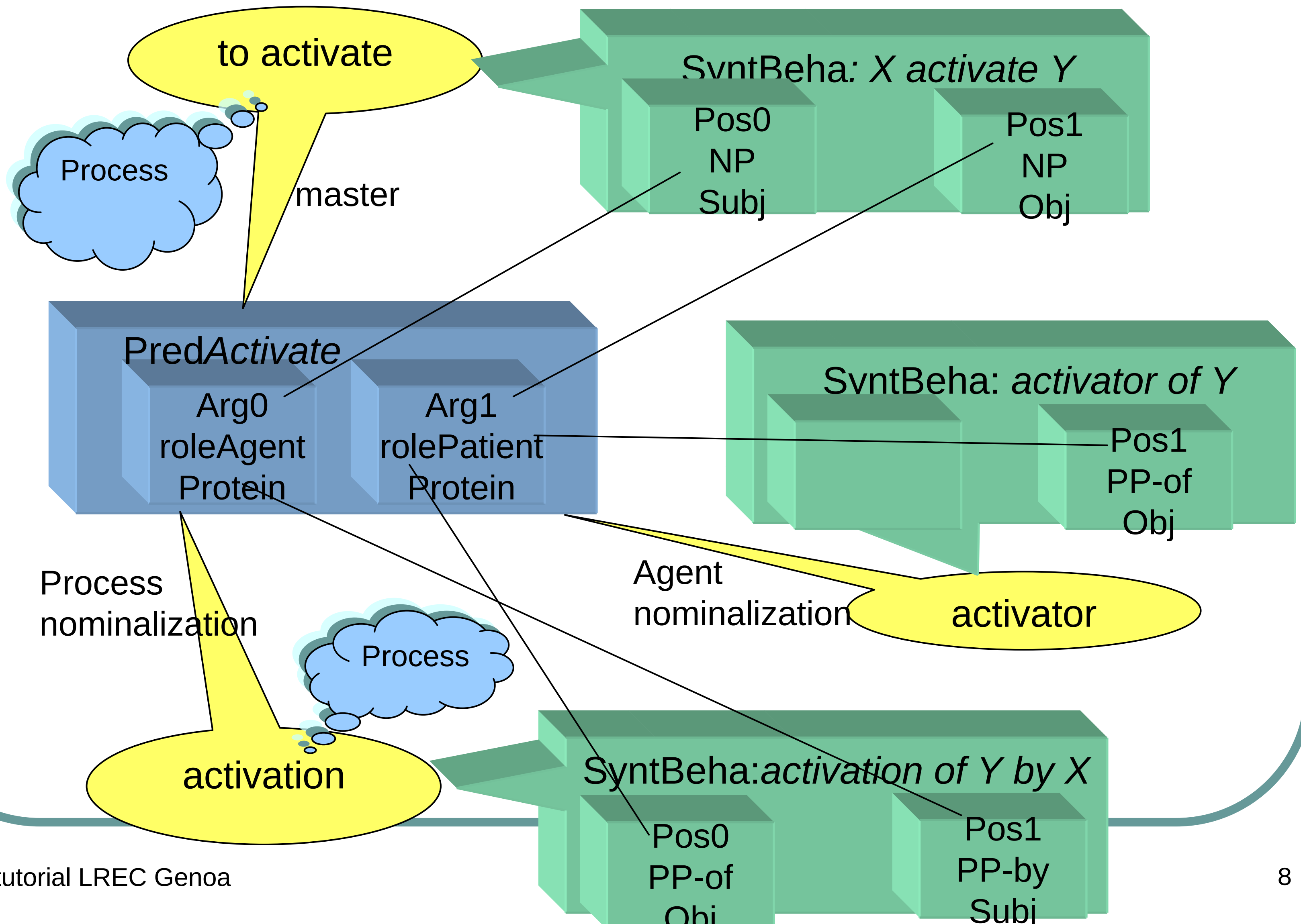
Semantic Definition	text source language view	*Semantic Definition* is an element for a narrative description of a *Sense* or a *Synset*. *Semantic Definition* is not provided for use by programs. *Semantic Definition* is provided to ease the maintenance by human beings and could be displayed to the final user. A sense or a synset can have zero to many definitions. The narrative description could be expressed in another language than the one of the lexical entry
Proposition	label type text	*Proposition* is an element that refines *SemanticDefinition*. Optionally, a definition can be defined by several propositions.

Description of Classes

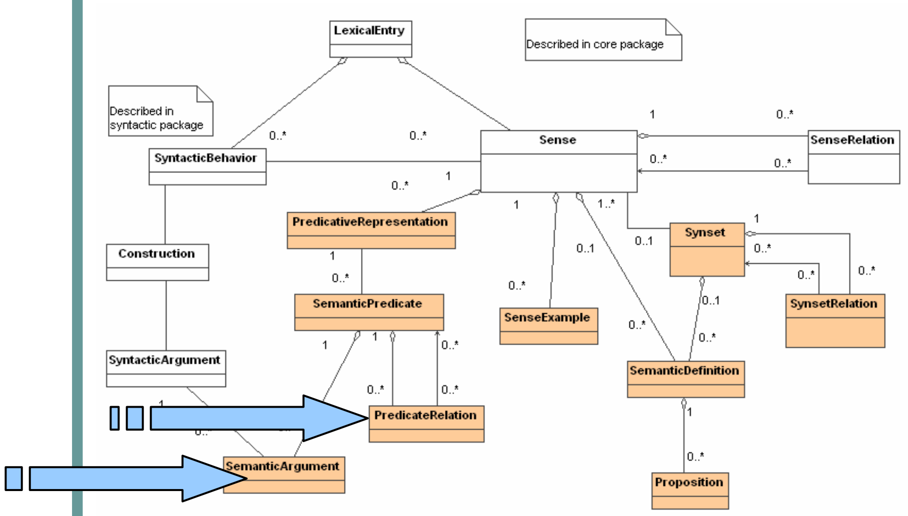


Semantic Predicate	Label definition	*Semantic Predicate* is an element that describes an abstract meaning together with the association with *Semantic Arguments*. A semantic predicate may be used to represent the common meaning between different senses that are not necessarily fully synonyms. These senses may be linked to lexical entries whose parts of speech are different.
Predicative Representation	Type comment	*Predicative Representation* describes the link between *Sense* and *Semantic Predicate*. A semantic derivation between a sense of a noun and a sense of a verb can be linked to a shared predicate. the predicative representation of the sense of the noun can be qualified as /verbNominalization/.

The *power* of Predicate: clustering of semantic entries

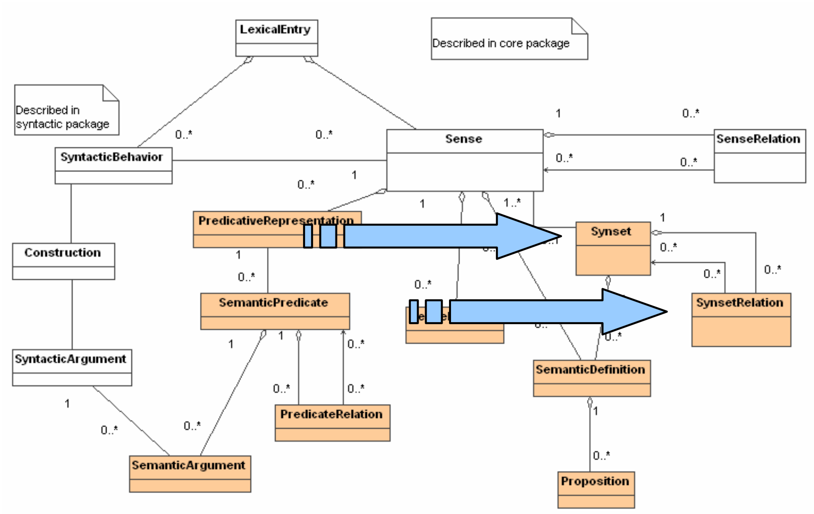


Description of Classes



Semantic Argument	semanticRole restriction	*Semantic Argument* is an element entering into the specification of a Predicate. It is dedicated to the linking of a semantic actant with a syntactic actant that is expressed by means of a *Syntactic Argument*.
Predicate Relation	Label Type	*Predicate Relation* permits to describe the relations between two or more semantic predicates

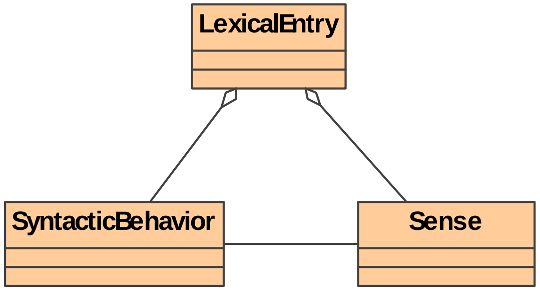
Description of Classes



Synset	Label source	*Synset* links synonyms. *Synset* is an element that describes a common and shared meaning within the same language. *Synset* may link senses of different lexical entries with the same part of speech.
Synset Relation	Label Type	*Synset Relation* permits to link two or more *Synsets*

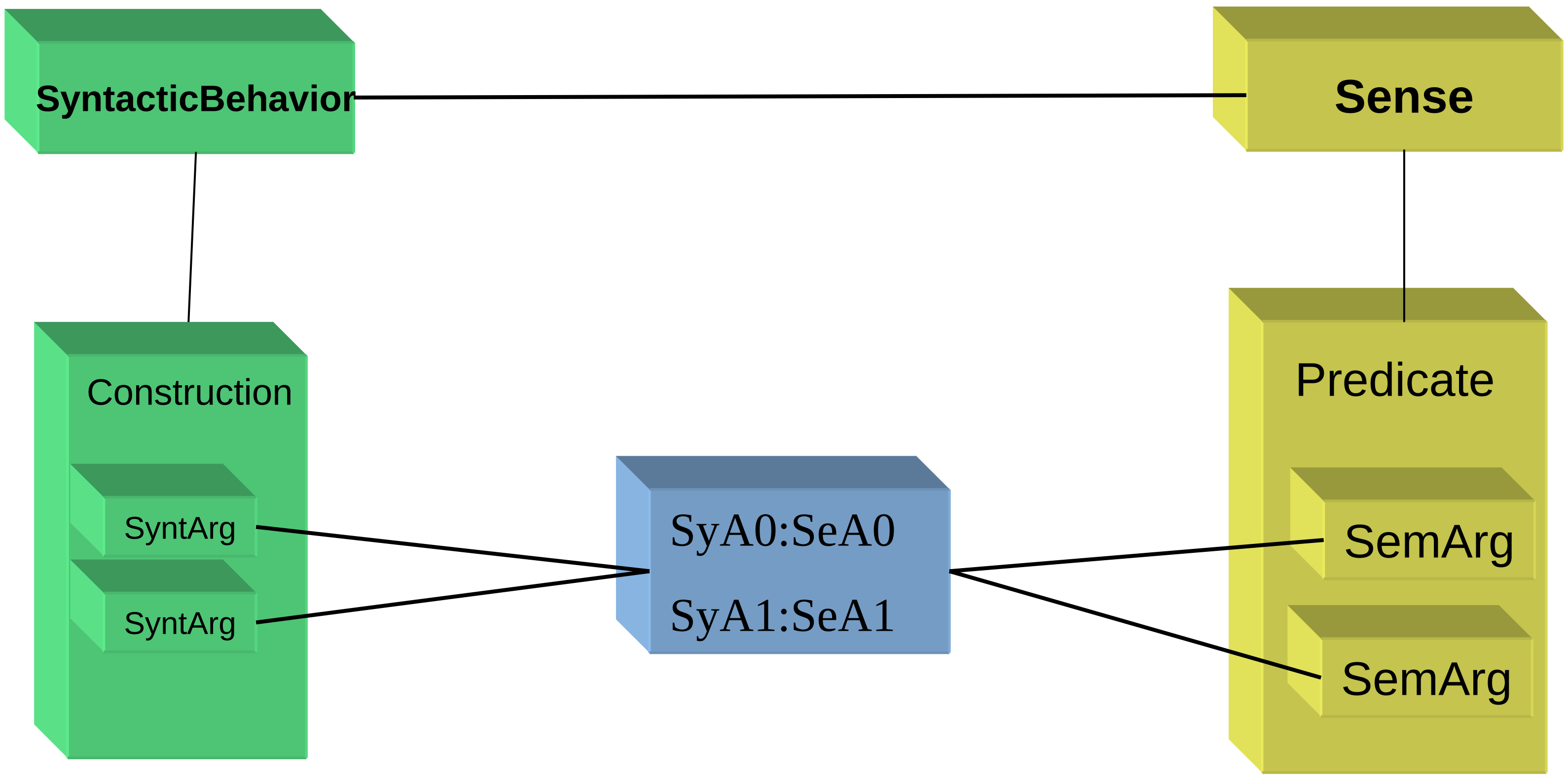
Syntax/semantic connection

- Connection between the triangle LexicalEntry SyntacticBehavior and Sense ... but not only ...

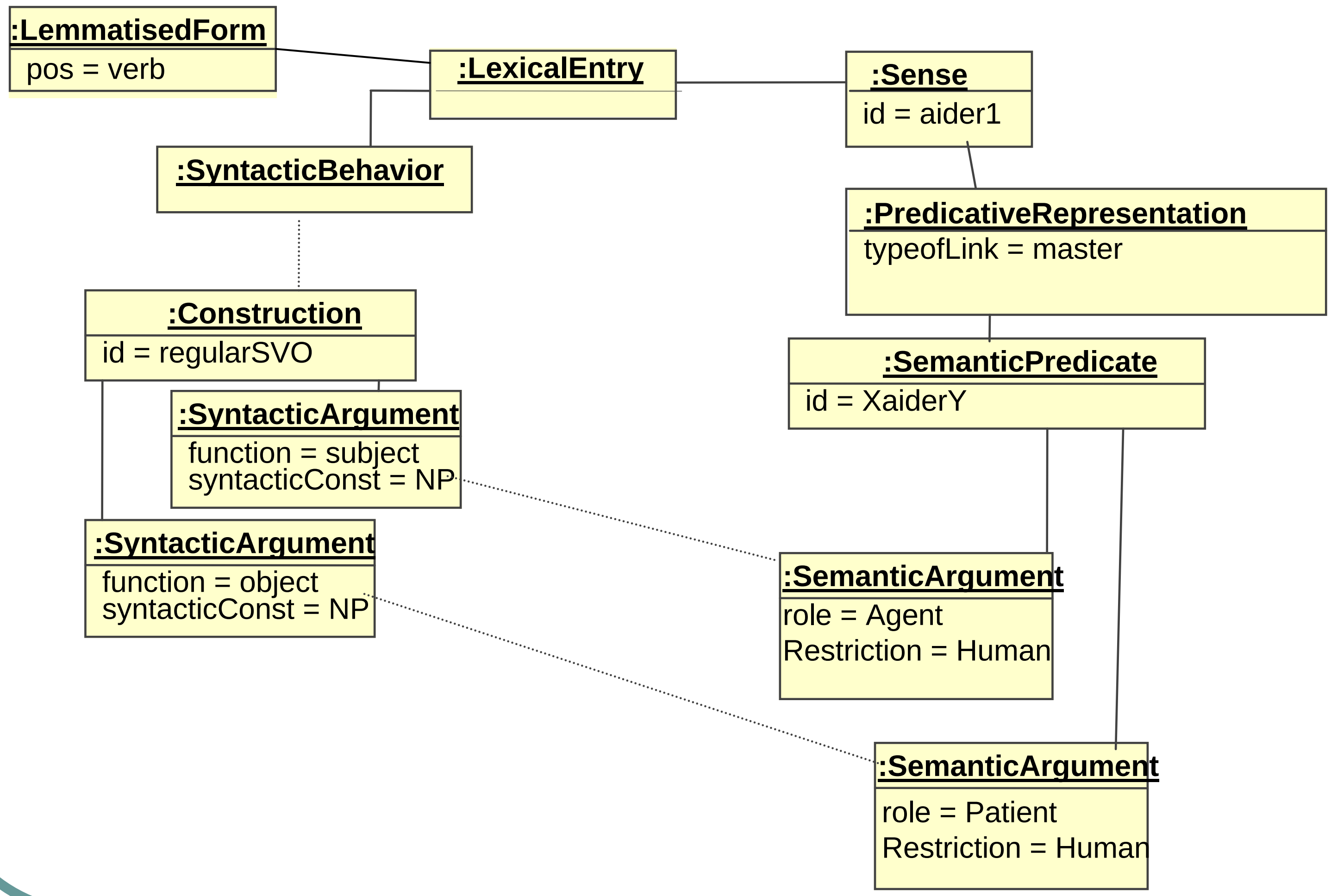


- i.e. ... not only the Sense and the SyntacticBehaviour can be connected but the connection can be pushed down

The mechanism for linking the predicate and the construction



Syntax/semantic connection: a UML instance



XML representation

```
…             <LexicalEntry>
              <LemmatisedForm>
                  <DC att="pos" val="verb"/>
              </LemmatisedForm>
              <Sense id="aider1">
                  <PredicativeRepresentation>
                      <DC att="typeofLink" val="master"/>
                      <SemanticPredicate id="XaiderY">
                          <SemanticArgument id="arg0">
                              <DC att="role" val="Agent"/>
                              <DC att="restriction" val="Human"/>
                          </SemanticArgument>
                          <SemanticArgument id="arg1">
                              <DC att="role" val="Patient"/>
                              <DC att="restriction" val="Human"/>
                          </SemanticArgument>
                      </SemanticPredicate>
                  </PredicativeRepresentation>
              </Sense>
              <SyntacticBehavior constructions="regularSVO">
              </SyntacticBehavior>
              </LexicalEntry>
              <Construction id="regularSVO">
                  <SyntacticArgument semargs="arg0">
                      <DC att="function" val="subject"/>
                      <DC att="syntacticConst" val="NP"/>
                  </SyntacticArgument>
                  <SyntacticArgument semargs="arg1">
                      <DC att="function" val="object"/>
                      <DC att="syntacticConst" val="NP"/>
                  </SyntacticArgument>
              </Construction> …
```

LMF and PAROLE-SIMPLE-CLIPS

- “Corpora e Lessici dell’Italiano Parlato e Scritto” (CLIPS) is a wide-coverage and multipurpose computational lexical database for Italian. To date, the lexicon is the largest Italian computational lexical resource, consisting 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.
- It extends 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
- 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.
- Conformity of the data to the model is ensured by an XML DTD, whereas internal formal validation is performed by an XML parser.

LMF and PAROLE-SIMPLE-CLIPS

- A Lexical Entry is a chain of lexical elements starting from morphology, crossing syntax and ending in semantics.
- The chain begins with a phonological unit that is simple (i.e. a PhU), a morphological unit (i.e. a Mu), a syntactic unit (i.e. a SynU),a semantic unit (i.e. a SemU), an intermediate object links syntax and semantics called a CorrespSynUSemU.
- Two features are presented here :
  - The linkage between syntax and semantics for the entry “costruire” (“to build” in English).
  - The semantic derivation around “costruire” (to build).
    - one entry is the verb “costruire” (to build)
    - the second entry is the noun “costruzione” (building)

The CLIPS entry *costruire_V*

```
<!—Syntax of the entry costruire →
<SynU id=”SYNUcostruireV”
            example=”costruire un ponte ; - una storia ; - una frase”
            description=”t-xa”>

<CorrespSynUSemU targetsemu=”USemD585costruire”
            correspondance=”ISObivalent”/>
</SynU>
<Description id=”t-xa”
            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 (not expande) →
    <Self id=”SELFVxa”
            syntagmatl=”STVxa”/>
            featurel=”TAUXavere”/>
```

The CLIPS entry *costruire_PhysicalCreation*

```
<!—Semantics of the entry costruire→
<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=”Human”/>
      <Argument            id=”ARG1costruire-1”      semanticrolel=”RoleProtoPatient”
                                                      informargl=”Building”/>
      <SemanticRole        id=”RoleProtoAgent”
             example=”soggetto di pensare, fare, sapere"
             comment="Usually 'translates' as SUBJECT in surface. 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"/>
```

Correspondence Syntax-Semantics

<!—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>

< !—Subcategorization frame
<InstantiatedPositionC range=**”0”** optional=**”YES”** positionC=**”Psubj”**/>
<InstantiatedPositionC range=**”1”** optional=**”NO”** positionC=**”Pobj”**/>
<!---Argument structure
<Argument id=**”ARG0costruire-1”** semanticrolel=**”RoleProtoAgent”**
informargl=**”Human”**/>
<Argument id=**”ARG1costruire-1”** semanticrolel=**”RoleProtoPatient”**
informargl=**”Building”**/>

The entries *costruzione*_N2 *costruzione*_*Building*

<!—Syntax of the second entry →

<SynU id=**”SYNUcostruzioneN”**

example=**”la cosa costruita”**

description=**“nv-0-xc”**

<!—description is not expanded for this entry→

<!—Semantics of the second entry→

<SemU id=**”Usem4174costruzione”**

example=**” una orribile costruzione deturpava il paesaggio”**>

semantictype=**”Building”**>

<PredicativeRepresentation typeoflink=**”VerbNominalization”**

predicate=**”PREDcostruire-1”**/>

The two entries are bound at the semantic level by two sorts of links:

a) they share the same predicate

b) a relation states that “costruzione” is the *resulting state* of “costruire”.

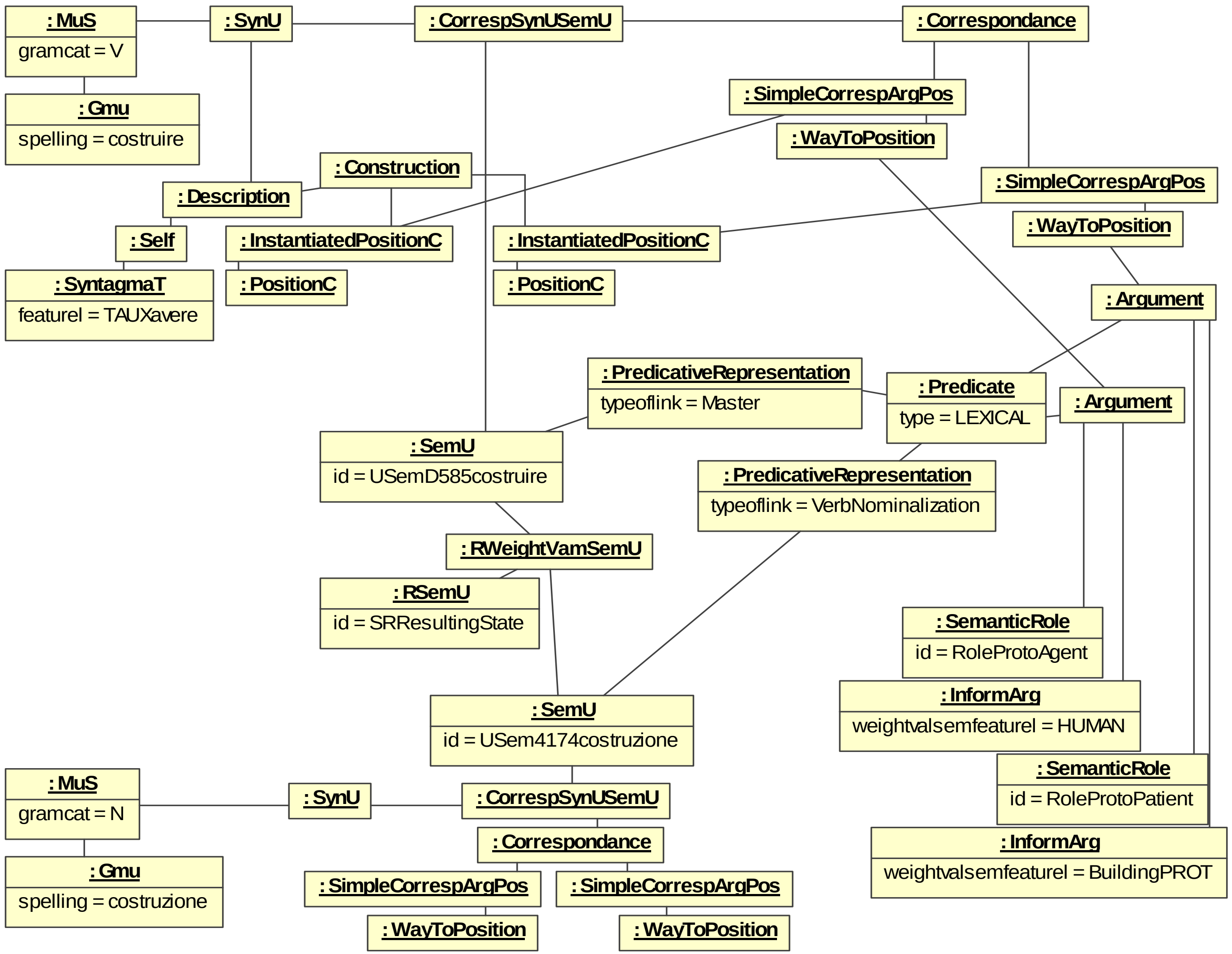
Linking *costruzionePhysicalCreation* to Pred*costruire*

<CorrespSynUSemU targetsemu=**”Usem4174costruzione”** correspondance=**”Aug0to2obj”**/>

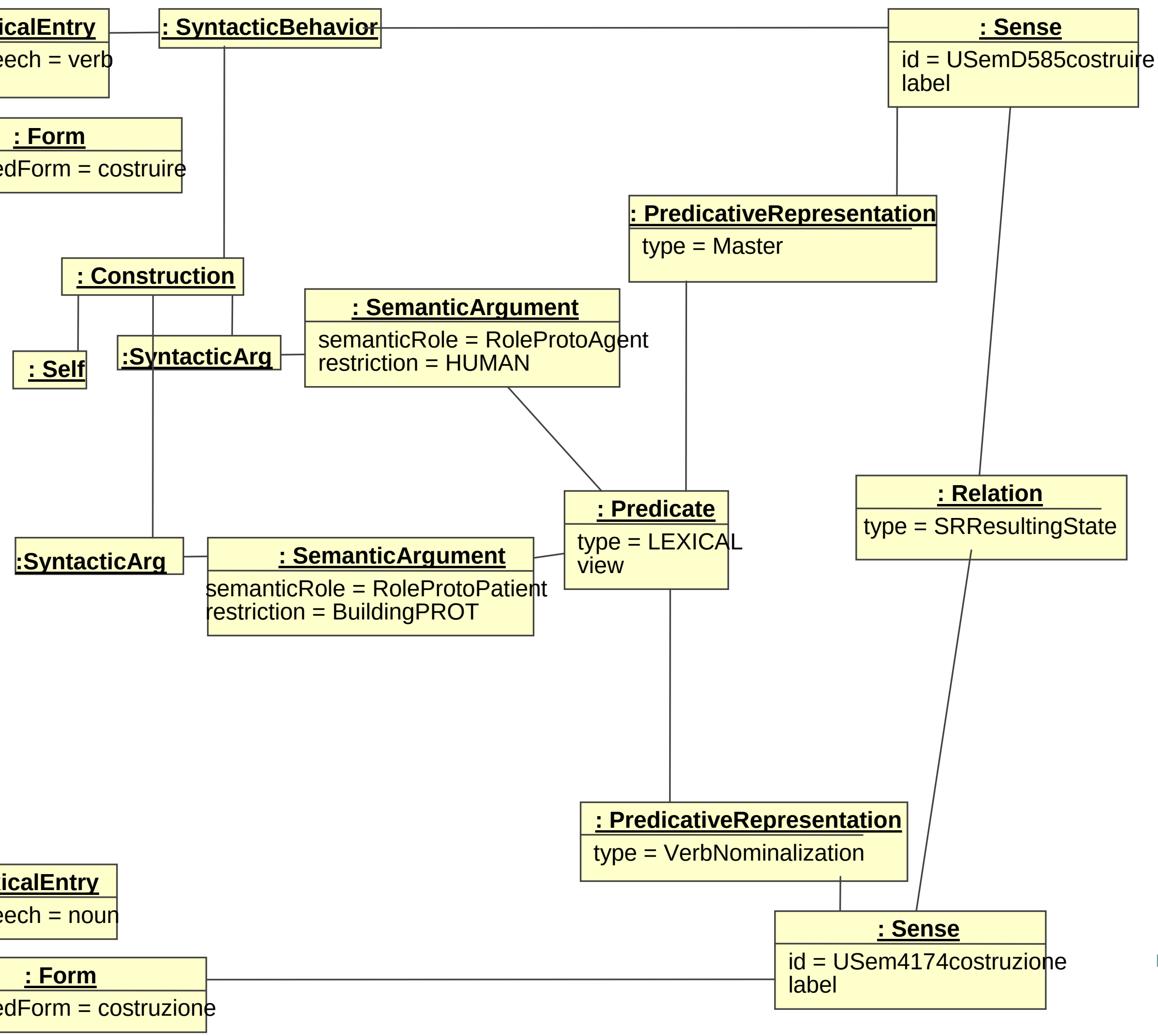
<!—Type of linkage between syntax and semantics→
<Correspondance id=**”Aug0to2obj”** comment=**”augmented mapping where the two semantic arguments are not realized on the surface and the head absorbes the patient/object argument”**/>

<!---Argument structure
<Argument id=**”ARG0costruire-1”** semanticrolel=**”RoleProtoAgent”** informargl=**”Human”**/>
<Argument id=**”ARG1costruire-1”** semanticrolel=**”RoleProtoPatient”** informargl=**”Building”**/>

Migration into UML



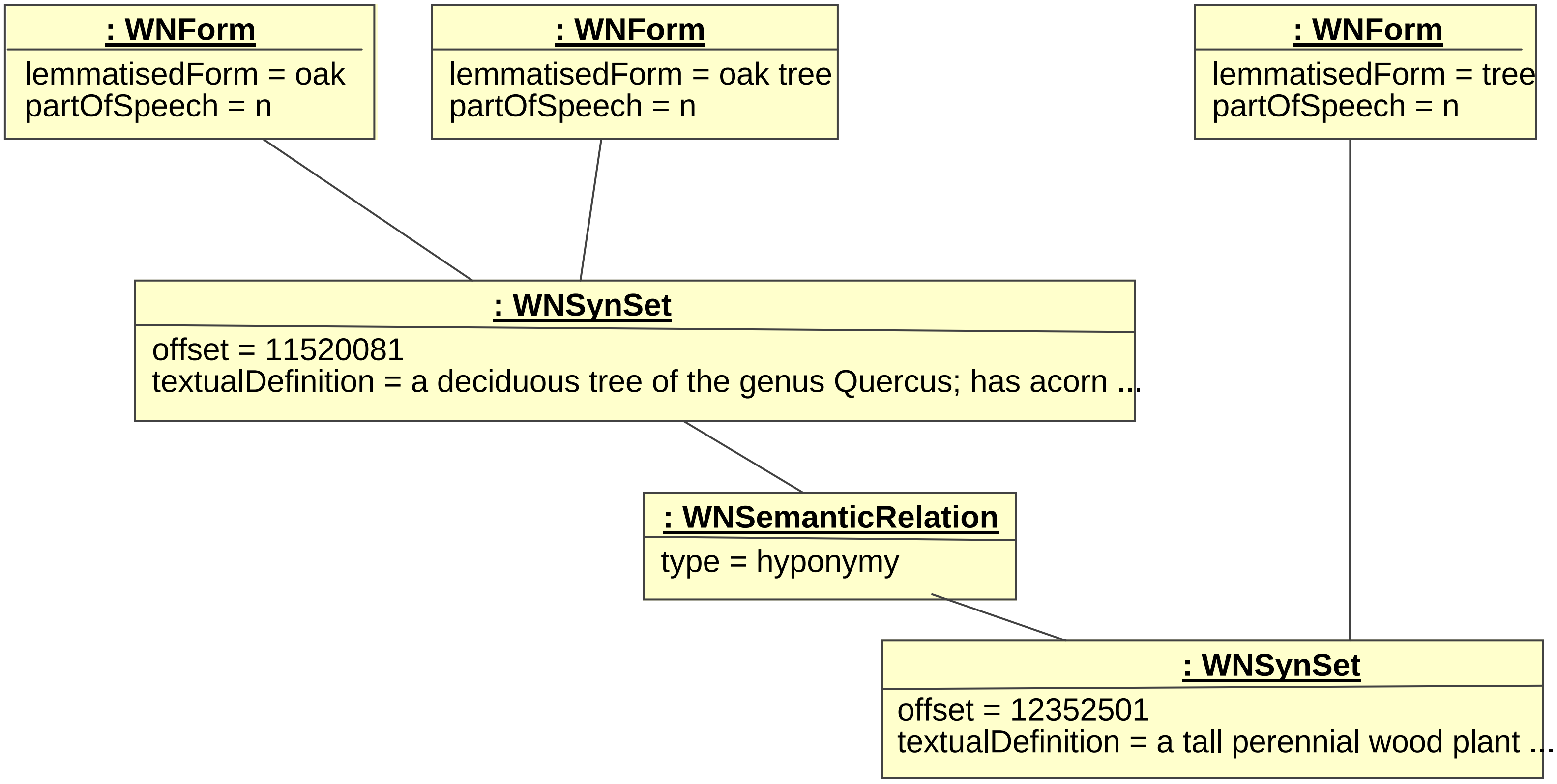
Migration into LMF



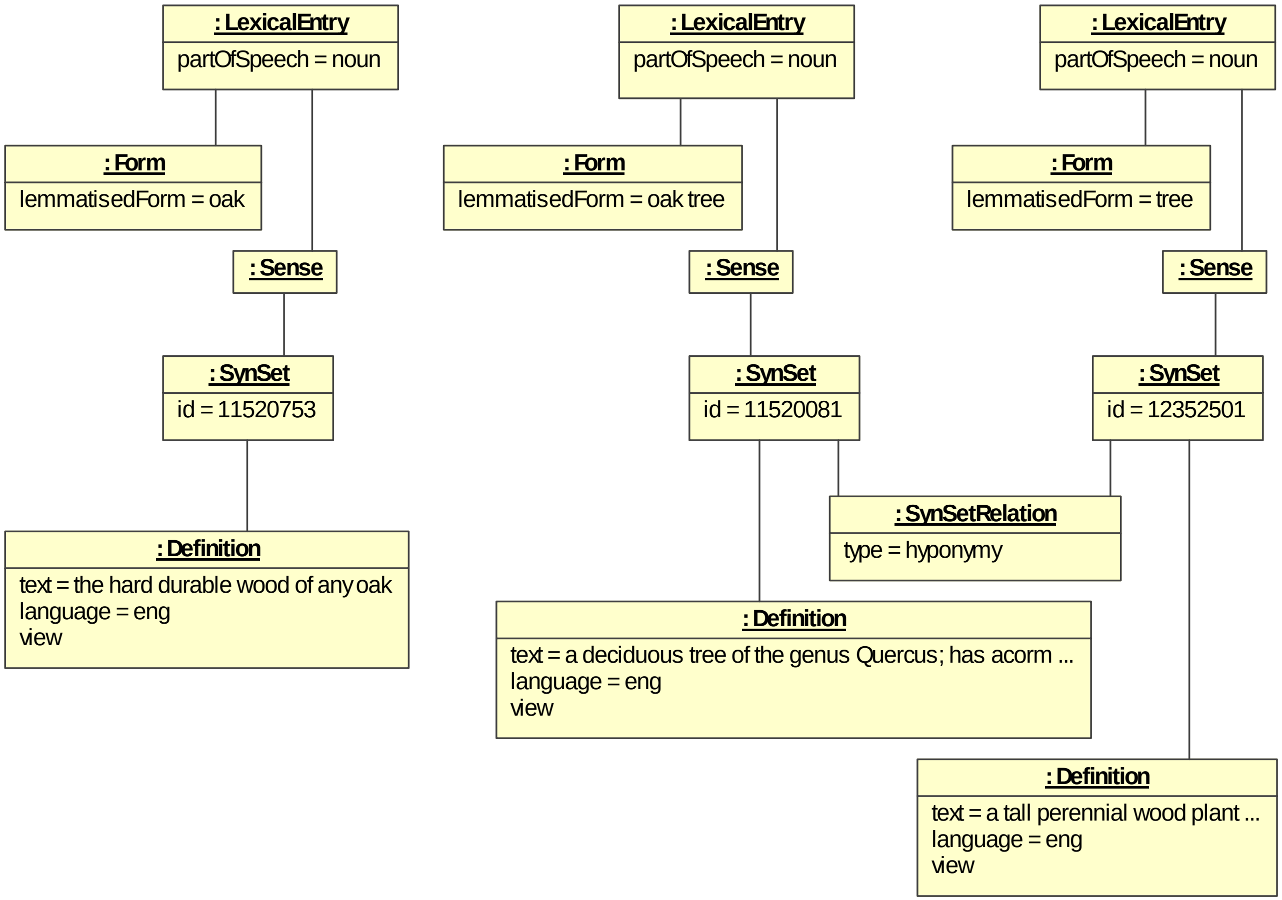
LMF and WordNet

- WordNet is an online English lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory
- WordNet is one of the most popular lexical databases. (http://wordnet.princeton.edu).
- Information in WordNet is organized around logical grouping called synsets.
- English nouns, verbs, adjectives and adverbs are described. Each synset consist of a list of synonymous words and pointers that describe the relation between this synset and other synsets.
- A word or collocation 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.
- Semantic relations include hypernymy/hyponymy, antonymy, entailment, meronymy/holonymy. These semantic relations link the synonym sets in an ontological structure.

A WordNet example: the synset *oak, oak tree*



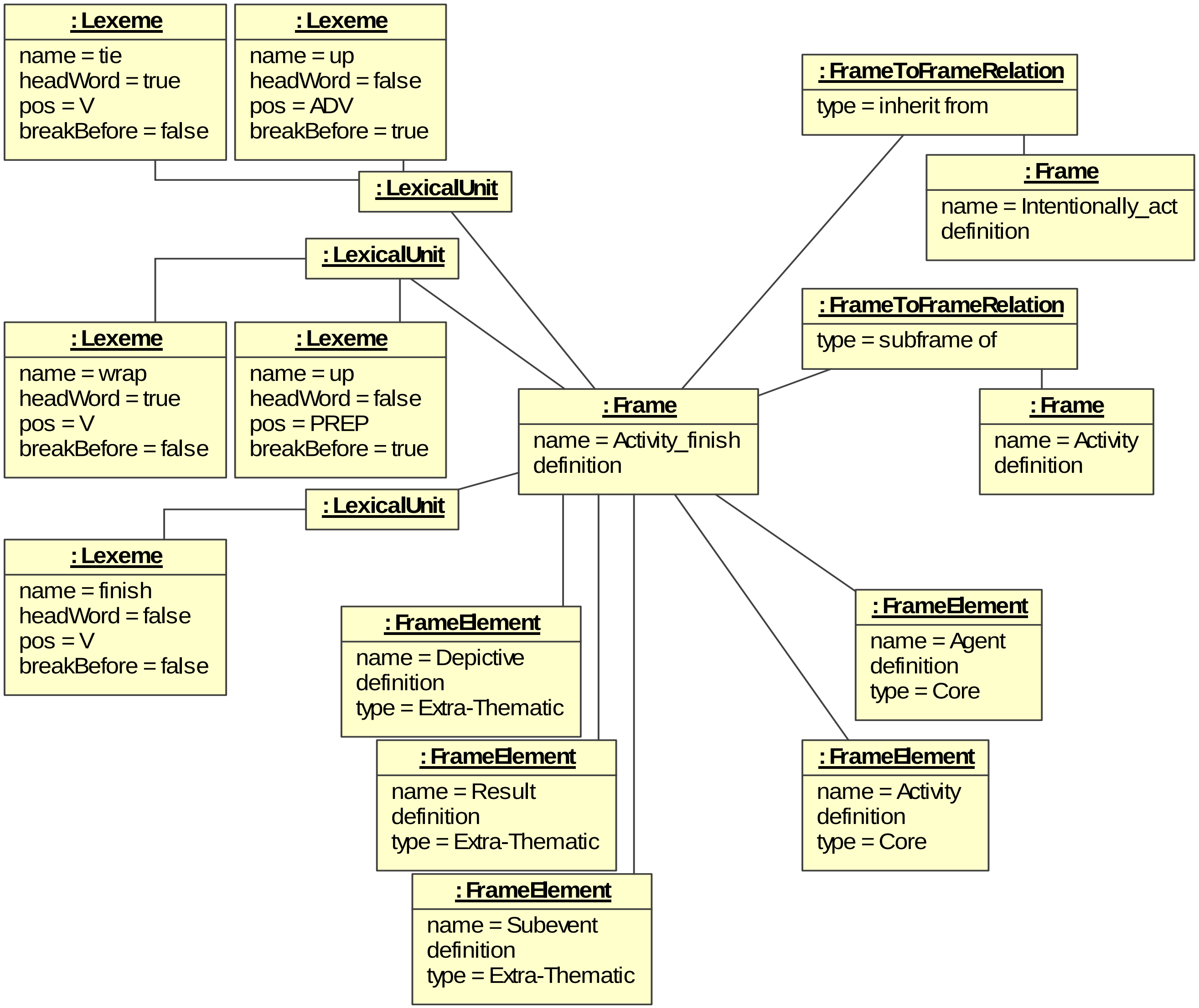
Migration into LMF



LMF and FrameNet

- FrameNet is an on-line lexical resource for English based on frame semantics and supported by corpus evidence
- 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.
- 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 frame is a script-like structure of inferences that characterize a type of situation, object or event.
- In the case of predicates, 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).
- 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 (PATIENT), a grammatical function (OBJECT) and a phrase type (NP).

A FrameNet example: the frame *Activity_finish*



Migration into LMF