Reference number of working document: **ISO/TC 37/SC 4 N130 Rev.9**

Date: 2006-03-15

ISO CD 24613:2006

Committee identification: ISO/TC 37/SC 4

Secretariat: KATS

Language resource management—Lexical markup framework (LMF)

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International standard
Document subtype: if applicable
Document stage: 30.00
Document language: en

Table of contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard 24613 was prepared by Technical Committee ISO/TC 37, *Terminology and other language resources*, Subcommittee SC 4, *Language resource management*.

ISO 24613 is designed to coordinate closely with ISO Draft Revision 12620, *Computer applications in terminology – Data categories –Data category registry,* and ISO DIS 16642, *Computer applications in terminology – TMF (Terminological Markup Framework)*.

Annexes A-G form an integral part of this International Standard.

Introduction

Optimizing the production, maintenance and extension of lexical resources is one of the crucial aspects impacting human language technologies (HLT) in general and natural language processing (NLP) in particular, as well as human-oriented translation technologies. A second crucial aspect involves optimizing the process leading to their integration in applications. The Lexical Markup Framework (LMF) is an abstract metamodel that provides a common, standardized framework for the construction of computational lexicons. LMF ensures the encoding of linguistic information in a way that enables reusability in different applications and for different tasks. LMF provides a common, shared representation of lexical objects, including morphological, syntactic, and semantic aspects.

The goals of LMF are to provide a common model for the creation and use from small to large scale lexical resources, to manage the exchange of data between and among these resources, and to enable the merging of large numbers of different individual electronic resources to form extensive global electronic resources. As an XML-based format, LMF utilizes Unicode (ISO 10646) in order to represent the scripts and orthographies used in lexical entries, including all corresponding equivalents, regardless of language. The ultimate goal of LMF is to create a modular structure that will enable true content interoperability across all aspects of lexical resources.

LMF is comprised of the following components:

- The core model comprises a metamodel, i.e., the structural skeleton of LMF, which describes the basic hierarchy of information included in a lexical entry. The core model is supplemented by various resources that are part of the definition of LMF. These resources include:

  — Specific data categories used by the variety of resource types associated with LMF, both those data categories relevant to the metamodel itself, and those associated with the extensions to the core model;

  — The constraints governing the relationship of these data categories to the metamodel and to its extensions;

  — Standard procedures for expressing these categories in XML and thus for anchoring them on the structural skeleton of LMF and relating them to the respective extension models;

  — The vocabularies used by LMF to express related informational objects as XML elements and attributes and methods for describing how to extend LMF through linkage to a variety of specific lexical resources (extensions) and methods for analyzing and designing such linked systems.

- Extensions of the core model, which are documented in this standard in annexes, include:

  — Machine readable lexicons

  — Natural Language Processing lexicons

LMF extensions are expressed in a framework that describes the reuse of the LMF core components (such as structures, data categories, and vocabularies) in conjunction with the additional components required for a specific lexical resource.

Types of individual instantiations of LMF can include such lexical resources as fairly simple lexical databases, NLP and machine-translation lexicons, as well as electronic monolingual, bilingual and multilingual lexical resources. LMF provides general structures and mechanisms for analyzing and designing new lexical resources, but LMF does not specify the structures, data constraints, and vocabularies to be used in the design of specific lexical resources. LMF also provides mechanisms for analyzing and describing existing lexical resources using a common descriptive framework. For the purpose of both designing new lexical resources and describing existing lexical resources, LMF defines the conditions that allow the data expressed in any one lexical resource to be mapped to the LMF framework, and thus provides an intermediate format for lexical data exchange.

1 Scope

This International Standard describes the Lexical Markup Framework (LMF), a high level model for representing data in lexical resources used with multilingual computer applications.

LMF shall provide mechanisms that allow the development and integration of a variety of lexical resource types. These mechanisms shall be able to represent existing lexicons as far as possible. If this is impossible, problematic information must be identified and isolated.

This standard is designed to be used in close conjunction with the metamodel presented in ISO 16642:2003, *Terminology Markup Framework* and with ISO 12620, *Terminology and other language resources — Data categories.*

It supports specific linguistic processing environments such as the NLP model defined in *AFNOR/TC37/SC4/N090 Proposition de Norme des Lexiques pour le traitement automatique du langage* and existing lexical resource models such as the EAGLES International Standards for Language Engineering (ISLE) and Multilingual ISLE Lexical Entry (MILE) model.

2 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of ISO 24613. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on ISO 24613 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 639-1:2002, Codes for the representation of names of languages – Part 1: Alpha-2 Code.

ISO 639-2:1998, Code for the representation of languages – Part 2: Alpha-3 Code.

ISO DIS 639-3:2005, Codes for the representation of languages – Part 3: Alpha-3 Code for comprehensive coverage of languages.

ISO 1087-1:2000, Terminology – Vocabulary – Part 1: Theory and application.

ISO 1087-2:1999, Terminology – Vocabulary – Part 2: Computer application.

ISO/IEC 10646-1:2003, Information technology – Universal Multiple-Octet Coded Character Set (UCS).

ISO/IEC 11179-3:2003, Information Technology – Data management and interchange – Metadata Registries (MDR) – Part 3: Registry Metamodel (MDR3)

ISO 15924:2004, Information and documentation – Code for the representation of names of scripts.

ISO 16642:2003, Computer applications in terminology – TMF (Terminological Markup Framework).

3 Definitions

For the purposes of this International Standard, the terms and definitions given in ISO 1087-1, ISO 1087-2 and the following apply:

abbreviated form
form whose some letters, numerals, pictograms or words have been omitted from a longer form

affix
morpheme added to a form or a stem and which changes the meaning of the word

antonym
word that means the opposite of another word in the same language

autonomous word
word that can appear as a single word or as a component of a multiword expression

Example: “father” in the multiword expression “father-in-law”

Note: opposed to non-autonomous word

circonstant
non-essential element associated with a verb when viewed from a theoretical perspective as opposed to **syntactic actants**

Example: Alfred (syntactic actant) read a book (syntactic actant) today (circonstant)

Note: Adverbs are possible circonstants for a sentence

closed data category
data category whose content is constrained by a list of permissible instances which comprise its **conceptual domain**

NOTE: A typical closed data category might be /grammatical number/, which can have as its content the values: /singular/, /plural/ or /dual/.

collocation
the habitual co-occurrence of individual lexical entries

Example: In English, “auspicious” and “occasion” frequently co-occur. In French, the adjective “aîné” is to be used with “frère” or “soeur” (older brother, older sister) as opposed to "âgé" which is used to mean "older" in other contexts.

collocational verb
See **support verb**

combination of morphological features
association of any two or more distinct morphological features

NOTE: An example of a combination of morphological features would be the pair: /grammatical number/ and /grammatical gender/.

complex data category
data category that can have content values

NOTE: Complex data categories include both **closed data categories** and **open data categories**.

compound word
word that contains other words

NOTE: A compound word is both a word and a MWE.

conceptual domain
set of permissible values associated with a **closed data category**

Note: The conceptual domain of the data category /grammatical number/ can be defined as /singular/, /plural/ and /dual/.

database
collection of data organized according to a pre-established structure [from ISO 1087-2]

data category
result of the specification of a given data field or the content of a closed data field

NOTE: A data category is to be used as an elementary descriptor in a linguistic structure or an annotation scheme. Examples are: /term/, /definition/, /part of speech/ and /grammatical gender/. Data categories for the management of lexical resources and terminology are comparable to data element concepts in ISO/IEC 11179-3:2003.

derivation
result of change in the form of a word to create a new word, usually by modifying the base/root or affixation

NOTE: Sometimes derivation signals a change in part of speech, such as "nation" to "nationalize". Sometimes the part of speech remains the same as in “nationalization” vs. “denationalization”.

elision
result of leaving out of a part of the form based on speech

Example: In rapid speech in English, “factory” is often pronounced as [‘fæktri]

Note: In certain languages, elision is written like in French: “le” + “enfant” yields “l’enfant”.

electronic lexical resource
ELR
lexical database
lexical resource
database consisting of individual data entries each of which documents a word and provides data pertinent to the senses associated with that word, as well as in some cases equivalent words in one or more languages [adapted from ISO 1087]

NOTE: Lexical resources can include features for spellchecking and grammar checking, parsing, concordancing, speech recognition and generation, semantic taxonomies and disambiguation, text segmentation, knowledge management, and other NLP functions.

electronic terminological resource
ETR
database consisting of individual data entries each of which documents a concept and provides data pertinent to the terms associated with that concept in one or more languages

etymology
information on the origin of a word and the development of its meaning [ISO 12620]

form
sequence of morphemes and affixe forms

form operation

any modification of the form

full form
complete representation of a word for which there is an abbreviated form [ISO 12620].

grammatical category
See **part of speech.**

homograph
word that is written like another word, but that has a different pronunciation, meaning, and/or origin [adapted from ISO 12620]

NOTE: An example of difference in meaning for the same spelling of a word is *bark*: *1) the sound made by a dog; 2) outside covering of the trunk or branches of woody plants; 3) a sailing vessel.*

homonym
word that sounds the same and is written the same as another word, but is different in meaning

NOTE: An example is “bear” as a /noun/ and “bear” as a /verb/.

homophone
word that sounds like another word, but is different in writing or meaning

NOTE: An example of difference in spelling is “pair” compared to “pear” or “pare” in “The cook used a knife to pare the pair of pears”.

human language technology
HLT
technology as applied to natural languages

NOTE: At the broadest level, these technologies cover: applying language knowledge to human machine interaction; providing automated multi-linguality in systems. These technologies include: speech recognition, spoken language understanding (i.e. speech interpretation), and speech generation; speaker identification and verification; dialogue design and analysis-controlled language design and processing document image analysis, optical character recognition, and handwriting recognition: recognition and understanding of multi-modal human communication; computer assisted text creation and editing; language analysis and understanding; information extraction; automatic generation of summaries; (synthetic) speech generation; language identification, machine translation and computer aided translation; production of language resources and the tools to support them.

inflected form
form that a word can take when used in a sentence or a phrase

NOTE: An inflected form of a word is associated with a combination of morphological features, such as grammatical number or case.

inflectional paradigm
set of form operations that builds the various inflected forms of a lemmatised form

NOTE: An inflectional paradigm is not the explicit list of inflected forms.

interlingua
an abstract intermediary language used in the machine translation of human languages

lemmatised form
lemma
conventional form chosen to represent words or MWE

NOTE: In European languages, the lemmatised form is the /singular/ if there is a variation in /number/, the /masculine/ form if there is a variation in /gender/ and the /infinitive/ for all verbs. In some languages, certain nouns are defective in the singular form, in which case, the /plural/ is chosen. Certain words are also defective in the /masculine/ in which case, the /feminine/ is chosen. The lemmatised form can be graphical or phonetic.

lexical database
lexical resource
See **electronic lexical resource**

lexicon
resource comprising words, MWE and affixes

NOTE: A special language lexicon or a lexicon prepared for a specific NLP application can include a specific subset of language.

morpheme
smallest meaningful sequence of letters, pictograms and numerals

machine translation lexicon
electronic lexical resource in which the individual entries contain equivalents in two or more languages together with semantic information to facilitate automatic or semi-automatic processing of lexical units during machine translation.

morphological feature
category induced from the inflected form of a word

NOTE: ISO 12620 provides a comprehensive list of values for European languages. An example of a morphological feature is: /grammatical gender/.

morphology of a word
morpho-syntax of a word
description comprising the lemmatised form or forms of a word, plus additional information on its /part of speech/ data categories, possibly its inflectional paradigm or paradigms, and possibly its explicitly listed inflected forms.

NOTE: Despite the reference to syntax, morpho-syntactic information does not include syntactic information.

multiword expression
MWE
group of words that either:
- has properties that are not predictable from the properties of the individual words or their normal mode of combination
- are governed by a specific pattern

Note: A MWE can be a compound word, a fragment of a sentence or a sentence. The group of words making up an MWE can be continuous or discontinuous. It is not always possible to mark a MWE with a part of speech information.

Example: A group of words that has properties not predictable from the properties of the individual words is for instance: "to be over the moon" that means something different from what it appears to mean. Groups of words governed by a specific pattern are for instance: "apple pie", "pear pie" with respect to the pattern "<fruit> pie".

natural language processing
NLP
field covering knowledge and techniques involved in the processing of linguistic data

non-autonomous word
word that appears in multiword expressions but cannot appear alone

Example: In French “au fur et à mesure”, the component “fur” cannot appear alone. In English, “to take umbrage”, the component “umbrage” cannot appear alone.

See also autonomous word

object language
language of the lexical object being described [ISO 16642 definition 3.10]

open data category
data category whose content is completely optional

Example: Typical open data categories might include /term/, /lemma/, /definition/.

orthography
a way of spelling or writing words that conforms to a specified standard

Note: Aside from standardized spellings of alphabetical languages, such as standard UK or US English, or reformed German spelling, there can be variations such as transliterations or romanizations of languages in non-native scripts, stenographic renderings, or representations in the International Phonetic Alphabet. In this regard, orthographic information in a lexical entry can describe a kind of transformation applied to the form that is the object of the entry. The specific value /native/ represents the absence of transformation.

part of speech
grammatical category
word class
category assigned to a word based on its grammatical and semantic properties

NOTE: ISO 12620 provides a comprehensive list of values for European languages. Examples of such values are: /noun/ and /verb/.

polyseme
word with multiple meanings

romanization
transcription or transliteration from non-Latin script into Latin script

script
set of graphic characters used for the written form of one or more languages (ISO/IEC 10646-1, 4.14)

Note: The description of scripts ranges from a high level classification such as hieroglyphic or syllabic writing systems vs. alphabets to a more precise classification like Roman vs. Cyrillic. Scripts are defined by a list of values taken from ISO-15924. Examples are: Hiragana, Katakana, Latin and Cyrillic.

semantics of a word
description of the meanings of the word

simple data category
data category that is itself the possible content of a **closed data category**, but that cannot itself have content

Example: /masculine/, /feminine/, and /neuter/ are possible simple data categories associated with the conceptual domain of the closed data category /grammatical gender/ as it is associated with the German language.

single word

word that does not contain any other word

splitting conditions
the criteria why a linguistic phenomena is described by one element or by several elements

Example: The criteria used when deciding whether a particular word is a polyseme whose multiple meanings belong to one entry or a homonym with multiple etymologies, which usually requires multiple entries.

stem
the main part of a form or one of the main parts of a form

subcategorization frame
valency
set of restrictions on a verb indicating the properties of the syntactic actants that can or must occur with it

support verb
collocational verb
verb that has a generic semantic contribution and that combines with an noun to form a lexicalised unit

Note: Generally, the subject of the verb is a participant in an event most closely identified with the noun.

Examples: "take an exam" or "give an exam". In these examples, "take" and "give" do not have inherent meaning based on their semantics, but rather are used in a conventional, generic way to express a collocational conceptualization.

synonym
word with the same meaning as another word in the same language

syntactic actant
one of the essential and functional elements in a clause that identifies the participants in the process referred to by a verb

Example: Alfred (syntactic actant) read a book (syntactic actant) today (circonstant)

Note: The subject, indirect object and direct object are possible syntactic actants for a sentence.

See also circonstant

syntax of a word
description of the behavior of the word with respect to other words in a sentence or a phrase

transcription
form resulting from a coherent method of writing down speech sounds

transliteration
form resulting from the conversion of one writing system into another

usage note
note explaining the correct and/or incorrect use of a word

valency
See **subcategorization frame**

variant

one of the alternative forms of a word

word
linguistic unit composed of at least a part of speech and a lemma

NOTE: A word is either a single or a compound word. The description can be more complete with more morphological information and/or syntactic and semantic information.

word class
See **part of speech.**

word frequency
number of occurrences of a particular word in a certain corpus, divided by the number of words in this corpus

working language
language used to describe objects in a lexical resource [ISO 16642 definition 3.21]

4 Key standards used by LMF

4.1 Unicode

LMF shall be Unicode compliant and presumes that all data is represented in the Unicode standard.

4.2 ISO 12620 Data Category Registry (DCR)

The designers of a LMF conformant lexicon shall use data categories from the ISO 12620 DCR. If user-defined data categories are needed, the lexicon creators shall be responsible for negotiating the addition of user-defined data categories to the DCR. This supplemental set of data categories shall be represented and managed in conformance with ISO 12620.

4.3 Unified Modeling Language (UML)

LMF complies with the specifications and modeling principles of UML as defined by OMG [1]. LMF uses a subset of UML that is relevant for linguistic description.

5 The LMF Model

5.1 Introduction

LMF models consist of UML classes, associations among the classes, and a set of ISO 12620 data categories that function as attribute-value pairs. The data categories are used to adorn the UML diagrams that provide a high level view of the model. LMF specifications, textual descriptions that describe the semantics of the modeling elements, provide more complete information about classes, relationships, and extensions than can be included in UML diagrams.

In this process, the lexicon developer must use the classes that are specified in the **LMF core package** (section 5.2). Additionally, the developer can use classes that are defined in the **LMF extensions** (relevant annexes). The developer must define a data category selection as defined in the **LMF data category selection use** (section 5.4).

5.2 LMF Core Package

The LMF core package is a metamodel that provides a flexible basis for building LMF models and extensions.

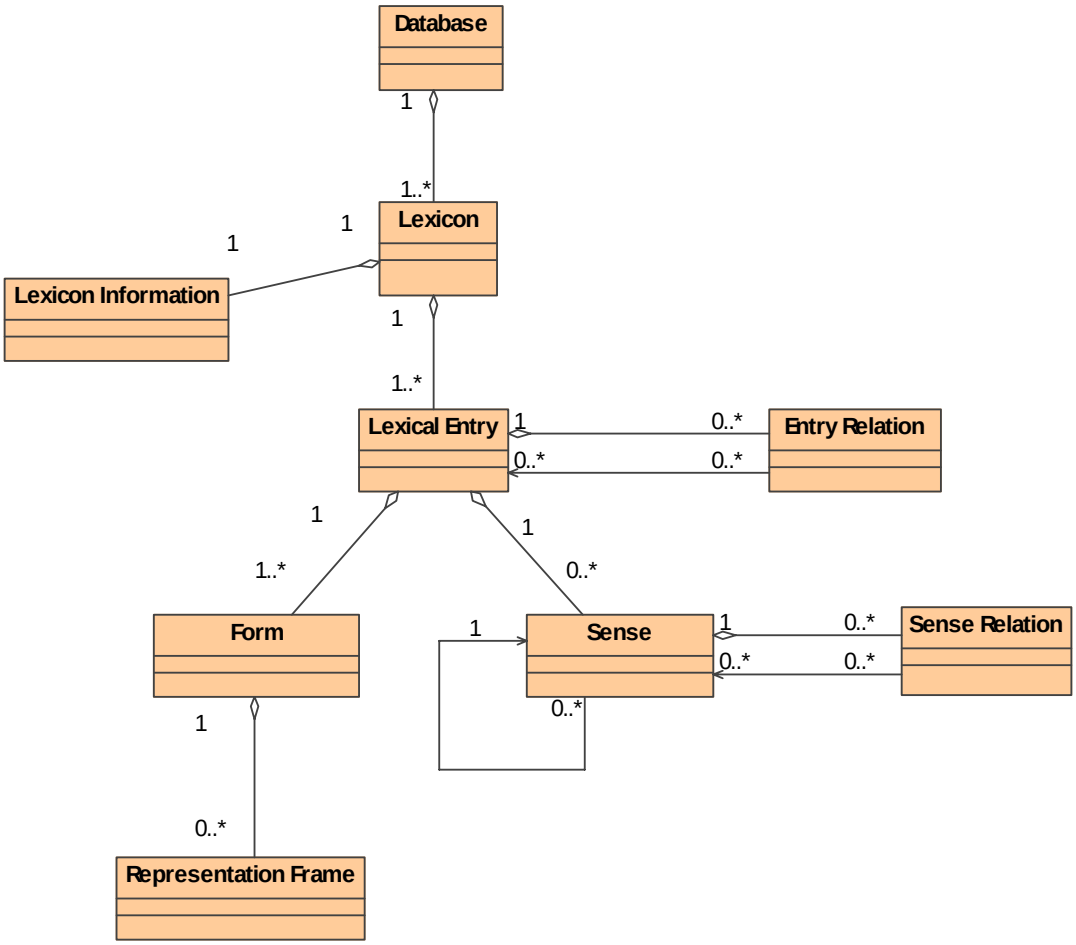


Fig 1: LMF Core Package

5.2.1 Database Class

The *Database* class is a singleton and represents the entire resource. The *Database* is a container for one or more lexicons.

5.2.2 Lexicon Class

The *Lexicon* class is the container for all the lexical entries of a source language within the database. A *Lexicon* must contain at least one lexical entry. The *Lexicon* class does not allow subclasses.

5.2.3 Lexicon Information Class

The *lexiconInformation* class contains administrative information and other general attributes. There is an aggregation relationship between the *Lexicon* class and the *lexiconInformation* class in that the latter describes the overall administrative information of each lexicon. The *lexiconInformation* class does not allow subclasses.

5.2.4 Lexical Entry Class

The *lexicalEntry* class represents a word, a multi-word expression, or an affix in a given language. The *lexicalEntry* is a container for managing the *Form* and *Sense* classes. Therefore, the *lexicalEntry* manages the relationship between the forms and their related senses. A *lexicalEntry* has one to many different forms, and may have from zero to many different senses. The *lexicalEntry* class does not allow subclasses.

5.2.5 Entry Relation Class

The *entryRelation* class is a cross-reference class that can link two to many LMF lexical entries within or across lexicons. The *entryRelation* class can contain attributes that describe the type of relationship.

5.2.6 Form Class

5.2.6.1 Form Class Specification

A *Form* class represents one lexical variant of the written or spoken form of the lexical entry. A *Form* contains a Unicode string that represents the word form and data categories that describe the attributes of the word form. The *Form* class itself may contain more than one orthographic variant (e.g. lemma, pronunciation, syllabification). The *Form* class allows subclasses.

5.2.6.2 Form Subclasses

The LMF core package includes two *Form* subclasses: the *lemmatisedForm* and the *inflectedForm*.

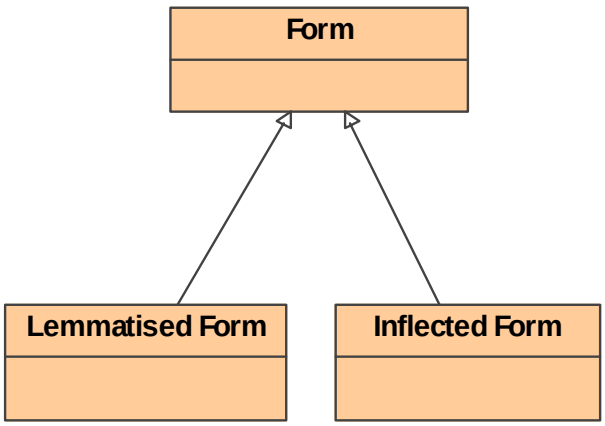


Fig 2: Form Subclasses

5.2.6.2.1 Lemmatised Form Class

The *lemmatizedForm* can only contain word forms that are of the type lemma.

5.2.6.2.2 Inflected Form Class

The *inflectedForm* can only contain word forms that are of the type inflected.

5.2.7 Representation Frame Class

If there is more than one orthography represented for the word form (Note: e.g., transliterations, Romanizations, pronunciations), the *Form* class may be associated with a *representationFrame* class. A *representationFrame* contains a specific orthography and one to many data categories that describe the attributes of that orthography.

5.2.8 Sense Class

The *Sense* class contains attributes that describe meanings of a lexical entry. The *Sense* class allows subclasses. The *Sense* class allows for hierarchical senses in that a part of a sense can be related to another part of the same sense.

5.2.9 Sense Relation Class

The *senseRelation* class is a cross-reference class that can link two to many LMF senses for one language within or across lexicons. The *senseRelation* class can contain attributes that describe the type of semantic relationship.

5.3 LMF Extension Use

All extensions conform to the LMF core package in the sense that a sub-set of the core package classes are extended. An extension cannot be used to represent lexical data independently of the core package. Depending on the kind of linguistic data, an extension can depend on another extension. From the point of view of UML an extension is a UML package. The dependencies of the various extensions are specified in the following diagram.

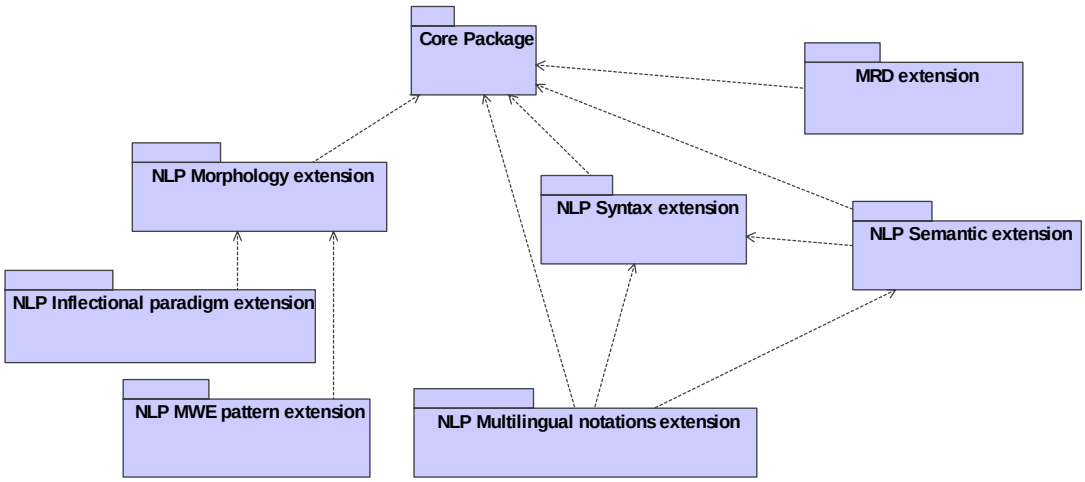


Fig 3: LMF Packages

Additional extensions may be developed over time. A new extension may be based on either the LMF core package itself, an existing extension to the core package, or may be a combination of extension mechanisms from the core package and existing extensions.

The extension mechanisms include:

- the creation of subclasses based on UML modeling principles
- the addition of new classes
- constraints on the cardinality and type of associations
- allowing different anchor points for the associations
- data category selections

The current LMF extensions are described in the annexes of this current standard. Creators of lexicons should select the subsets of these possible extensions that are relevant to their needs.

5.4 LMF data category selection use

5.4.1 LMF Attributes

All LMF attributes are complex data categories. Each value of an attribute is either a simple data category or a Unicode string.

5.4.2 Data Category Selection

The data category selection (DCS) lists and describes the set of data categories that can be used in a given LMF lexicon. The DCS also describes constraints on how the data categories are mapped to specific classes.

The kind of data categories that will be needed depends on:

- The design requirements of the lexicon developer, including the precision and extent of the data categories needed to describe the lexical features of the model.
- The languages selected and the complexity of the orthographic representations included.
- The constraints imposed by the core package and selected extensions.

5.4.3 Data Category Registry

The Data Category Registry (DCR) is a set of data category specifications defined by ISO 12620. The designers of any specific LMF lexicon shall rely on the DCR when creating their own data category selection.

5.4.4 User-defined Data Categories

Lexicon creators can define a set of new data categories to cover data category concepts that are needed and that are not available in the DCR. This supplemental set of data categories shall be registered with and managed in conformance with ISO 12620.

5.4.5 Lexicon Comparison

When two LMF conformant lexicons are based upon two different DCSs, comparison of the DCS in each lexicon provides a framework for identifying what information can be exchanged between one format and another, or what will be lost during a conversion. When LMF is used to describe an existing lexical resource, it will be necessary to map the existing lexical resource to corresponding data categories in the DCR.

5.5 LMF process

LMF shall be used according to the following steps.

Step 1: Define a LMF conformant lexicon

Step 2: Populate this lexicon

A LMF conformant lexicon is defined as the combination of a LMF core package, zero, one or more lexical extensions and a set of data categories. The combination of all these elements is described in the following UML activity diagram:

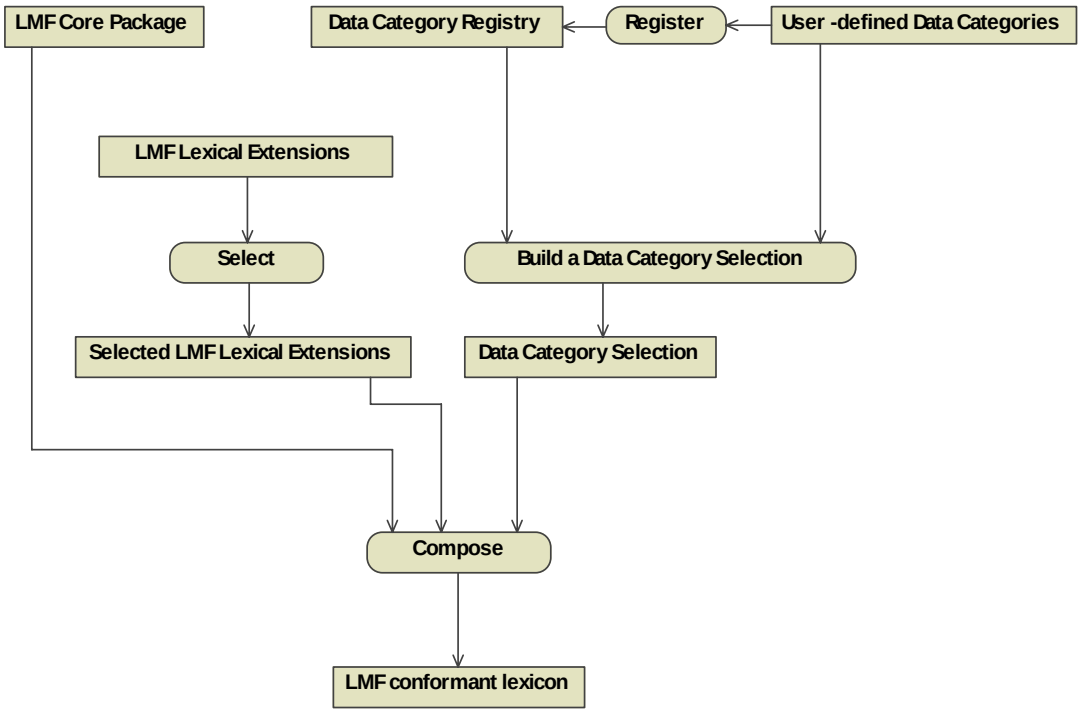


Fig 4: LMF Process

Annex A (normative) Machine Readable Dictionary Extension

A.1 Introduction

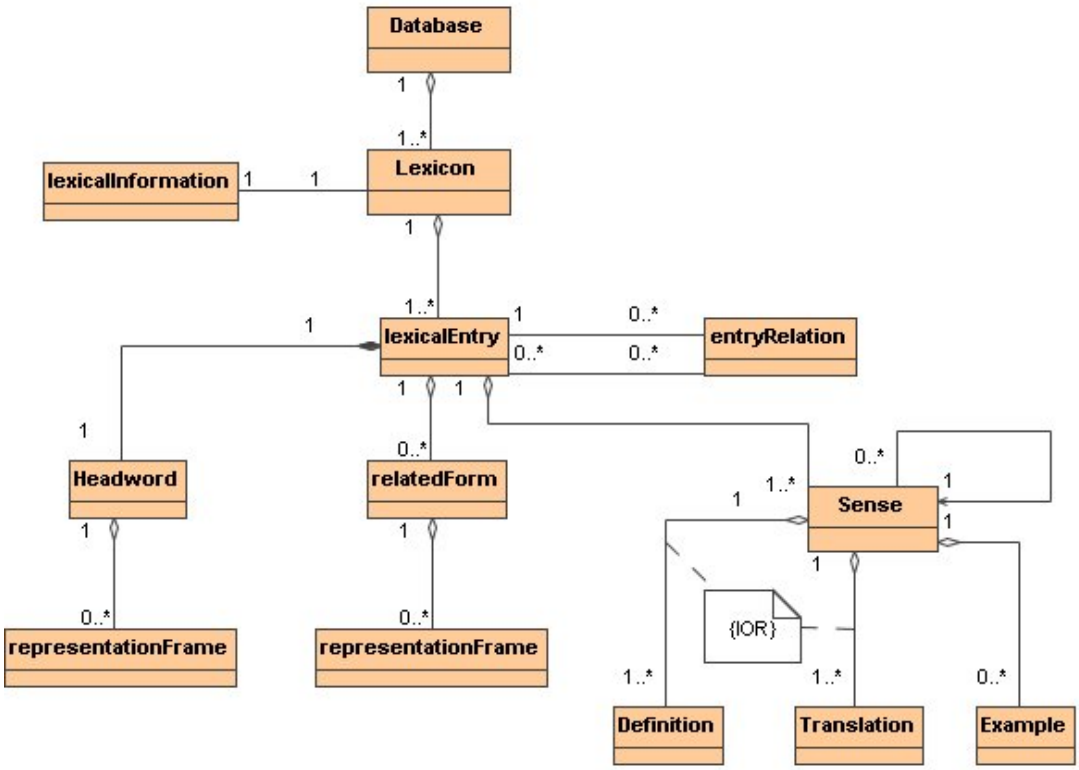
The MRD extension provides a meta model to represent data stored in machine readable dictionaries. The extension supports electronic machine readable dictionary access for both human and machine consumption. Since the MRD is based upon the LMF core package, it is designed to interchange data with other LMF extensions where applicable. The MRD extension uses the ISO 12620 DCR to represent core and MRD extension data categories.

The MRD extension utilizes the following extension mechanisms:

- Subclasses
- New classes
- constraints on the cardinality and type of associations
- data category selections

A.2 MRD Extension Package

The MRD extension package models monolingual and bilingual formats. The following UML diagram depicts the classes and subclasses for the MRD extension:



A.2.1 Core Package Classes in MRD

The MRD meta model utilizes the following core package classes:

- *Database* class
- *Lexicon* class
- *lexiconInformation* class
- *lexicalEntry* class
- *lemmatizedForm* class
- *inflectedForm* class
- *Sense* class
- *representationFrame* class

A.2.2 Subclasses in MRD

A.2.2.1 Headword Class Specification

A Headword class is a *Form* subclass that can only exist as a one to one relationship with the lexical entry in that a lexical must have at least one and only *Headword.* The *Headword* contains a Unicode string that represents the word form and data categories that describe the attributes of the word form.

A.2.2.2 relatedForm Class Specification

A lexical entry may be associated with zero or more *relatedForm* classes. The *relatedForm* is a *Form* subclass containing a word form that can be related to the Headword in one of a variety of ways, i.e. inflection, variation or abbreviation. This word form can also appear as a Headword in a separate lexical entry. There is no assumption that *relatedForm* is associated with the *Sense* in the lexical entry.

A.2.3 New Classes

A.2.3.1 Definition Class Specification

The *Definition* class contains a narrative description of the meaning of the Headword in the same language as the Headword.

A.2.3.2 Translation Class Specification

The *Translation* class provides an equivalent of the Headword in a target language.

A.2.4 Constraints on Associations and Cardinality

A.2.4.1 Definition and Translation Classes

A lexical entry can have zero or more definitions and zero or more translations, but must contain at least one of either.

A.2.5 Data Category Selections

A.2.5.1 MRD Entry Relation Class

The MRD *entryRelation* class extends the core package *entryRelation* class by admitting attributes that address *relatedForm* cross references. If a lexical entry contains a *relatedForm* that references another *lexicalEntry,* the *entryRelation* class contains pointers from a *relatedForm* to the lexical entry where it is originally contained.

Annex B (normative) Extension for NLP morphology

B.1 Objectives

The purpose is to provide the mechanisms to support the development of NLP lexicons that describe the extensional morphology of lexical entries.

B.2 Options

There appears to be no consensus on the approach for representing morphology, but it is possible to synthesize the situation by listing three different options:

- Option-1: Inflected forms are explicitly represented;
- Option-2: The Lexical Entry is connected to an inflectional paradigm that is fully and analytically described within the lexicon. The inflectional paradigm is considered as a pattern that is shared by a great number of words;
- Option-3: The inflectional paradigm refers to an external automaton or an opaque compiled program;

When option-3 is selected, it is impossible to modularize or exchange data.

B.3 Description of morphological model

B.3.1 Introduction

LMF NLP morphology is based on the assumptions that:

- For certain languages, it is possible to explicitly represent all the inflected forms (i.e. option-1). This is the purpose of the current extension.
- It is possible to fully describe the inflectional paradigm (i.e. option-2) by means of a symbolic description based on other elements. This is the purpose of the Inflectional Paradigm extension.

B.3.2 Connexion with core package

Instead of referring to *Lexical Entry* class, the various descriptive mechanisms in morphology refer to *Lemmatised Form* class. As an additional specification from core package, *Inflected Form* class is aggregated inside *Lemmatised Form* class.

B.3.3 Element description

Stem

A *Stem* is an element that holds a part of the *Lemmatised Form*. A *Lemmatised Form* may have zero, one or several stems.

List of Components

A multiword expression is comprised of autonomous or non-autonomous components. The components are ordered and aggregated by means of *ListOfComponents* class.

The mechanism can also be applied recursively, that is a multiword expression may be comprised of components that are themselves multiword expressions.

Inflectional Paradigm

An *Inflectional Paradigm* is an element that specifies how to associate a certain type of lemmatised form to its inflected forms.

B.4 Class diagram

The following UML diagram specifies the classes of the NLP morphological model[1].

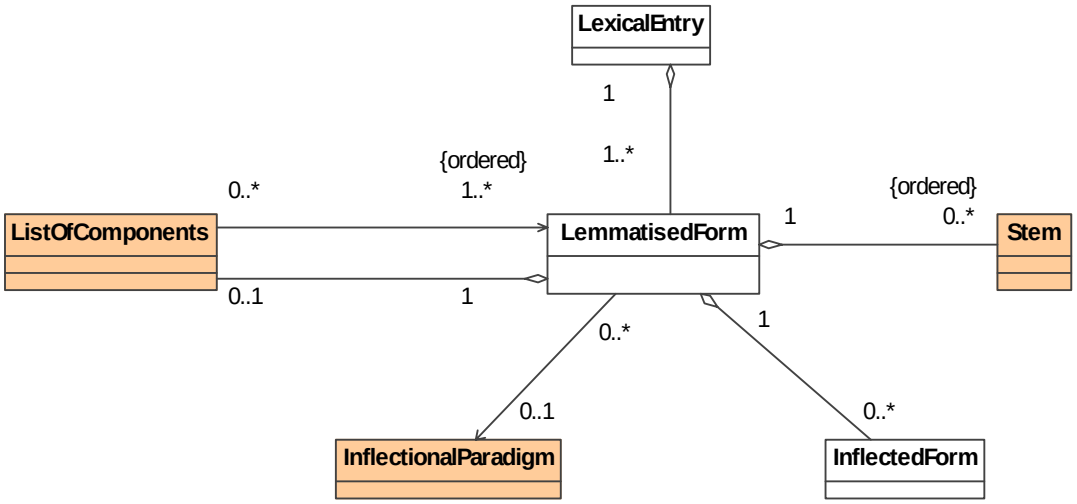


Fig B-1: morphological model

[1] In order to ease the reading, morphological classes are colored coded and classes taken from another section are white coded.

Annex C (normative) Extension for NLP syntax

C.1 Objectives

The purpose of this annex is to describe the properties of the word to be combined with other words in a sentence. The syntactic model describes specific syntactic properties of words and does not express the general grammar of a language.

C.2 Absence versus presence of syntax in a lexicon

The syntactic description is attached to the lexical entry unit and to the sense unit. Syntactic description is optional, so it is possible to describe morphology and semantics without any syntactic description. Instead of having a layer structure with three layers (i.e. Morphology, Syntax and Semantics) the associations form a triangle comprising three sub-parts: Morphology, Syntax and Semantics. Each vertex holds a central object that is respectively the Lexical Entry, the Syntactic Behavior and the Sense. Only the Lexical Entry is mandatory, the others are optional. Such a structure is modelled as follows:

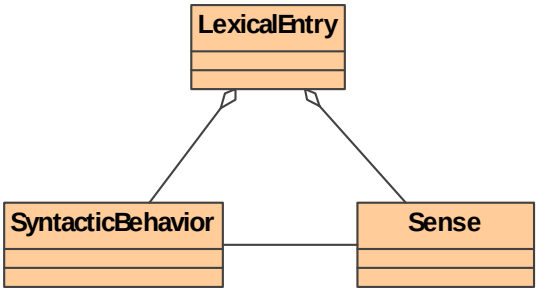


Fig C-1: triangle

C.3 Description of syntactic model

Syntactic Behavior

Syntactic behavior is an element that represents one of the possible behaviors of one or several senses. The presence of one syntactic behavior for a word means that this word can have this behavior in the given language. The detailed description of the syntactic behavior is defined in *Construction*.

Construction

Construction is the element that describes one syntactic construction. *Construction* is an element that is shared by all words that have the same syntactic behavior in the same language. A *Construction* can inherit relations and attributes from another more generic *Construction* by a reflexive link. So it is possible to integrate a hierarchical ontology of constructions.

Self

Self is the element that describes the central node of the Construction. Being connected to *Construction*, *Self* is an element that is shared by all words that have the same syntactic behavior. *Self* is the element that refers to the current lexical entry.

Syntactic Argument

Syntactic Argument is an element that describes a syntactic actant. A *Syntactic Argument* can be linked recursively to a Construction in order to describe deeply complex arguments. *Syntactic Argument* allows the connection with a semantic actant by means of *Semantic Argument.*

Construction Set

Construction Set element describes a set of syntactic constructions and possibly the relation that undergoes these Constructions. A *Construction Set* can inherit relations and attributes from another more generic *Construction Set* by a reflexive link. So it is possible to integrate a hierarchical ontology of construction sets.

Certain languages have simple syntax and other languages have complex syntax. In the latter, describing every behavior precisely is a huge task. The mapping from a representation where a predicate-argument structure is meant to describe 'deep syntactic' relations into a representation of surface grammatical relations or functions is subject to certain morpho-syntactic rules (active/passive voice) and to lexically determined features of the predicates (transitive, ergative, pronominal verbs). These mappings, **when regular,** can be described resorting to types or to sets of frames that a verb can enter into, and help to reduce redundant information in the lexicon. For this purpose, *Construction Set* is provided.

Construction Set is an element that regroups together various *Syntactic Constructions* that appear frequently for certain sets of words; the objective being to factorize syntactic descriptions and to have a minimum of syntactic behavior elements in the lexicon.

C.4 Class diagram

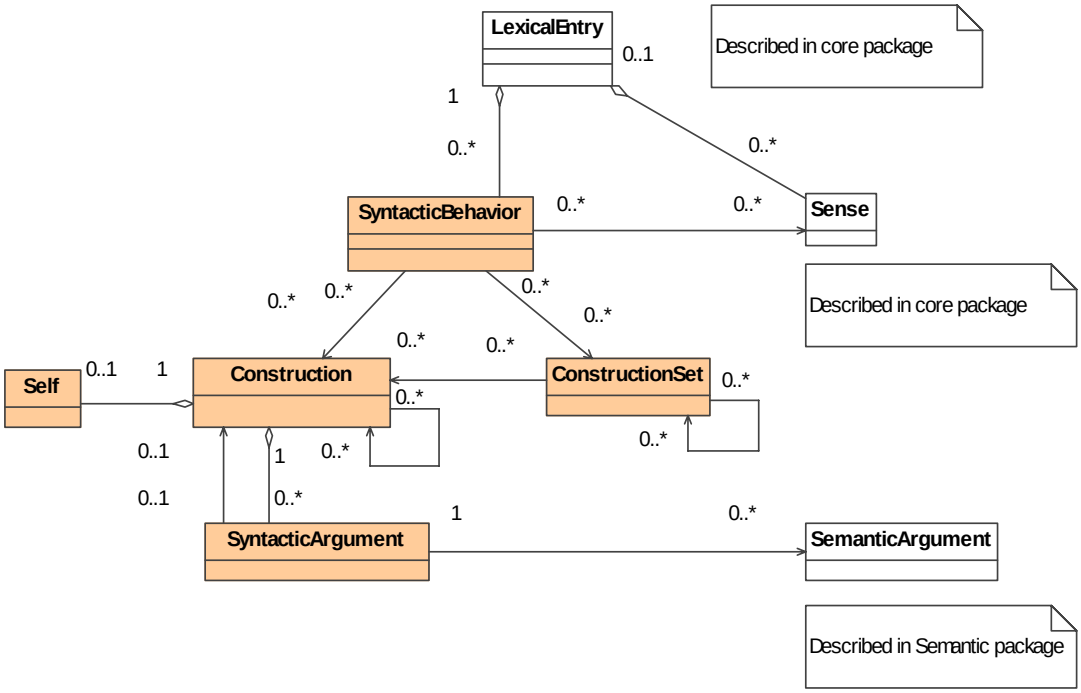


Fig C-2: syntactic model

Annex D (normative) Extension for NLP semantics

D.1 Objectives

The purpose of this section is to describe one sense and its relations with other senses belonging to the same language. Due to the intricacies of syntax and semantics in most languages, the section on semantics comprises also the connection to syntax. The linkage of senses belonging to different languages is to be described by using the multilingual section.

D.2 Description of semantic model

Sense

The *Sense* element is described in the core package. The *Sense* element being contained in the *Lexical Entry* element, *Sense* is not shared among two different lexical entries.

Sense Example

Sense Example is an element used to describe usages of the particular meaning of the *Sense* element. A sense can have zero to many examples. The language is the same as the one of the lexical entry but the text could be expressed in a more or less explicit way.

Semantic Definition

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

Proposition is an element that refines *SemanticDefinition*. Optionally, a definition can be defined by several propositions.

Semantic Predicate

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

Predicative Representation describes the link between *Sense* and *Semantic Predicate*.

Semantic Argument

Semantic Argument is an element that is dedicated to the linking of a semantic actant with a syntactic actant that is expressed by means of a *Syntactic Argument*.

Predicate Relation

Predicate relation permits to describe the relation between two or more semantic predicates.

Synset

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

Synset Relation permits to link two or more *Synsets*.

D.3 Class diagram

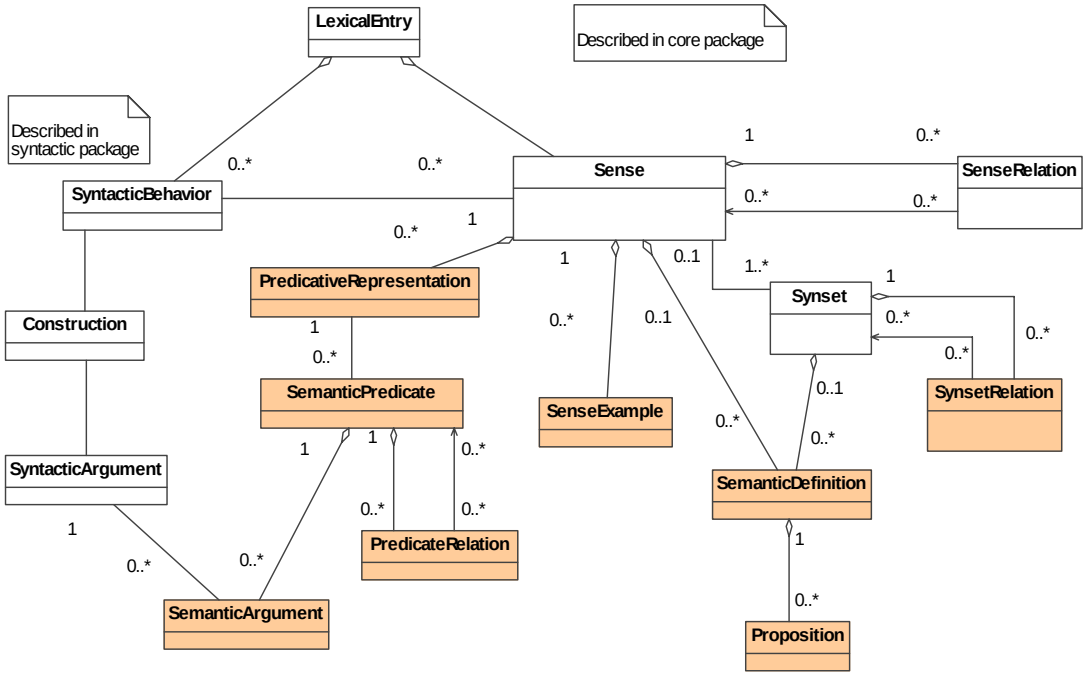


Fig D-1: semantic model

Annex E (normative) Extension for NLP multilingual notations

E.1 Objectives

The purpose is to describe the translation of a sense or a syntactic behavior from one language into one or several other languages.

E.2 Absence versus presence of multilingual notations in a lexicon

The multilingual model can be used for a lexical database describing two or more languages. There is no need to use the multilingual notations in a monolingual lexicon.

E.3 Options

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

Point 1: diversification and neutralization

In certain circumstances, simple bijection from one language to the next does not work very well because the precision of the source language is not the same as that of the target language.

Point 2: 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: the number of links explodes to unmanageable proportions.

Point 3: transfer or interlingual pivot

There are two approaches to multilingual translation in NLP that are transfer and interlingual pivot. Transfer operates based on syntax and interlingual pivot operates based on semantics. As a consequence, the model presented here must allow for both approaches. In the model, the pivot approach is implemented by a Sense Axis. The transfer approach is implemented by a Transfer Axis.

Point 4: representation of similar languages

A situation that is not very easy to deal with is how to represent translations to languages that are similar. Instead of managing two distinct copies, it is more effective to distinguish variations through a limited number of specific Axis, the vast majority of Axis being shared.

Point 5: direction and tests

Some multilingual lexicons are very declarative in the sense that every translation is represented by an interlingual object. But some other lexicons are very procedural in the sense that the translation is restricted by logical tests. These tests can be applied at the source language level or at the target language level.

E.4 Description of multilingual notations model

The model is based on the notion of Axis that link Senses, Syntactic Behavior and examples pertaining to different languages. Axis can be organized at the lexicon manager convenience in order to link directly or indirectly objects of different languages. A direct link is implemented by a single axis. An indirect link is implemented by several axis and one or several relations.

The model is based on three main classes:

- Sense Axis
- Transfer Axis
- Example Axis

Sense Axis

Sense Axis links different closely related senses in different languages. This element is used to implement the approach based on the interlingual pivot. The purpose is to describe the translation of words from one language to another. Optionally, *Sense Axis* may refer to an external knowledge representation system.

Sense Axis Relation

*Sense Axis Relatio*n permits to describe the linking between two different *Sense Axis*.

Transfer Axis

Transfer Axis is designed to represent multilingual transfer. The linkage between two languages is at the level of syntactic descriptions.

Transfer Axis Relation

Transfer Axis Relation links two *Transfer Axis*.

Source Test

Source Test permits to express a condition about the translation on the source language side.

Target Test

Target Test permits to express a condition about the translation on the target language side.

Example Axis

Example Axis provides documentation for sample translations.

E.5 Class diagram

The system is applicable to bilingual and multilingual lexicons.

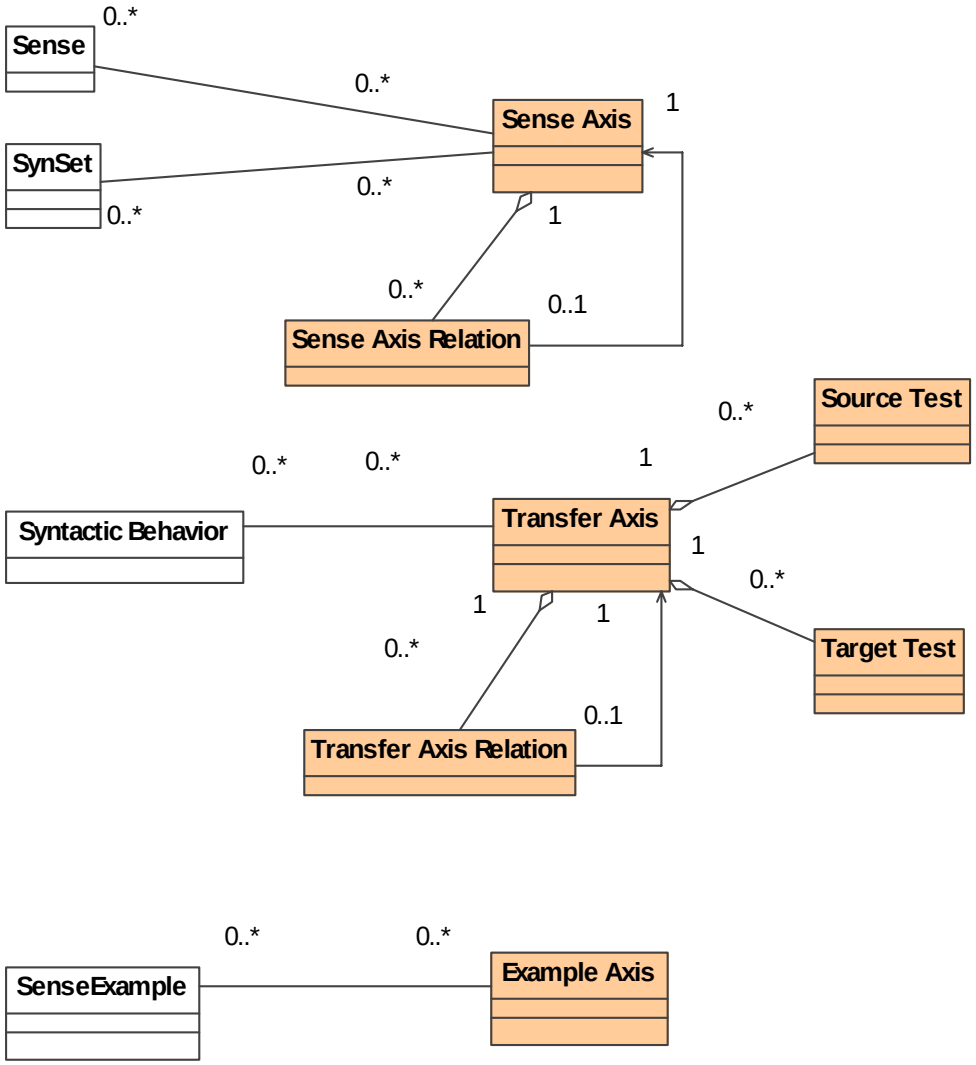


Fig E-1: multilingual notations model

E.6 Summary

The model:

a) allows the representation of transfer and interlingual pivot approach;

b) permits to share or duplicate multilingual notations;

c) is suited for both bilingual and multilingual lexicons;

Annex F (normative) Extension for NLP inflectional paradigms

F.1 Objectives

The purpose is to provide the mechanisms to support the development of NLP models that describe the intensional morphology of lexical entries. The inflected forms are not explicitly listed but the Lexical Entry is associated with a shared inflectional paradigm.

The goal is to describe all the pairs:

1) combination of morphological features (definition in section 3)
2) a mechanism to produce an inflected form

For the verb "*go*", for instance, one of these pairs will be:

1) (/third person/ + /singular/ + /present/)
2) a mechanism to produce "goes".

F.2 Absence versus presence of inflectional paradigms in a lexicon

Compared to the strategy of listing all inflected forms in a lexicon, the use of an inflectional paradigm has the following important advantages:

- Description of languages with complex morphology is possible. Otherwise, it is not possible.
- The linguistic knowledge describing how to associate a lemmatised form to an inflected form is factorized on a specific and explicit element instead of being spread in all entries.

F.3 Description of inflectional paradigm model

F.3.1 Introduction

For a given language, a paradigm is the description of the association between a lemmatised form and its inflected forms.

F.3.2 Inflectional paradigms for single words

The inflectional paradigm is the set of pairs that connects a combination of morphological features with a mechanism capable of computing an inflected form. The inflectional paradigm is shared by all the forms that have the same morphological pattern.

The mechanism for producing an inflected form is the following:

- The computation refers to the lemmatised form or a list of stems.
- The operations of the computation are specified in order to indicate that the string obtained by the previous point needs to be modified.

F.3.3 Inflectional paradigms for multiword expressions

Inflectional Paradigm element can be used to describe multiword expressions that do not rely on the grammar of the given language. The mechanism for creating a multiword expression can be applied to an agglutinative compound word that is considered to be a multiword expression without any graphical separator.

When used for a MWE, an *Inflection Paradigm* is defined by a set of *Morphological Features Combos* each of these being linked to one or several *Composers*.

F.3.4 Inflectional paradigms for hybrid combinations

Inflectional Paradigm element can combine a specification for single words and multiword expressions in the same paradigm.

F.3.5 Element description

Morphological features combo

The element combines *Inflected Form Calculators* with *Morphological Features*.

Inflected form calculator

InflectedFormCalculator class regroups a double set of operators: one for graphical computation and one for phonetic computation. *Operations* are ordered. Each operation is applied once.

InflectedFormCalculator class has at least the following attribute:

- /stem/ that is a reference to the lemmatised form or to a stem. The zero value indicates the use of the lemmatised form. A strictly positive integer value means a reference to the stems attached to the lexical entry.

Operation

The *Operation* class represents one form operation (definition in section 3). An operation is either a graphical operation or a phonetic operation. Each operation is associated with an ordered list of arguments.

Operation Argument

An *Operation Argument* is an element associated with *Operation*.

An Operation Argument specifies either a textual content or a position.

Morphological Feature

The element represents a morphological feature (definition in section 3).

Composer

A *Composer* is an element that represents the presence of a specific component in a multiword expression.

Composer class has at least the following attribute:

- /rank/ that refers to a specific component described by the ListOfComponents element.

F.4 Class diagram

The following UML diagram shows the classes of the inflectional paradigm model.

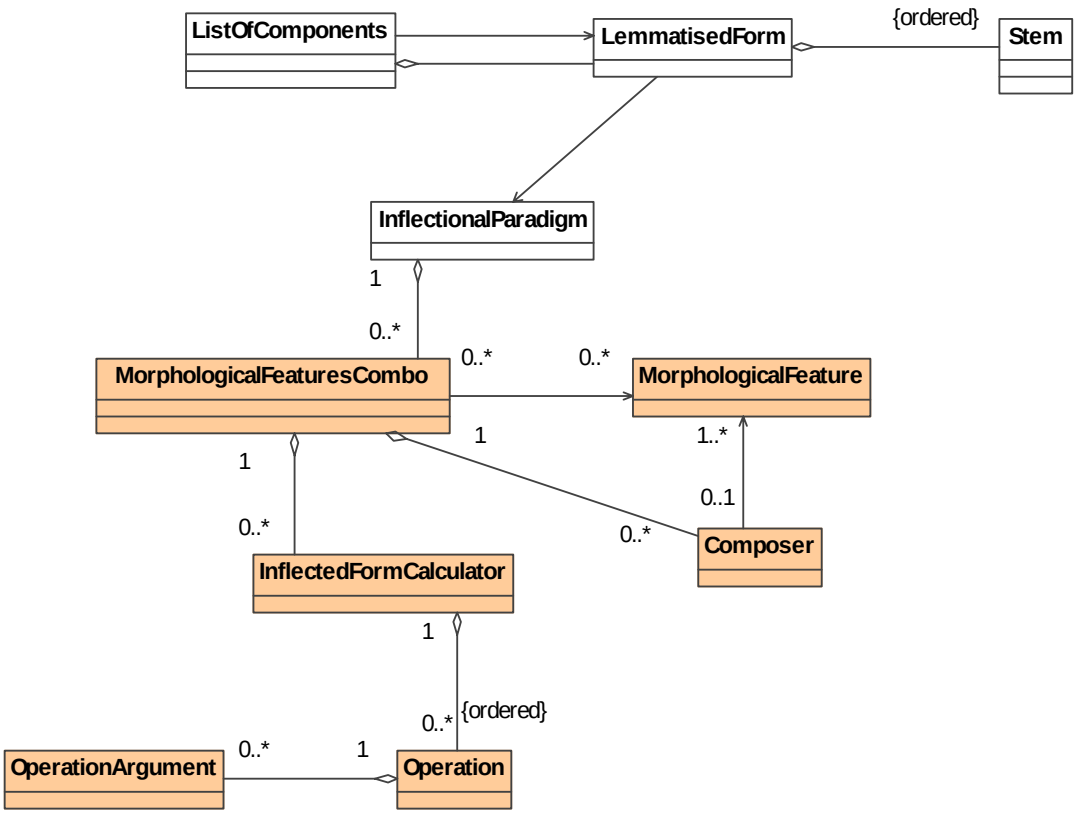


Fig F-1: inflectional paradigm model

F.5 Summary

The model presented here permits the description of inflectional morphology. The model is the same for languages with simple morphology and for languages with complex morphology.

Annex G (normative) Extension for NLP multiword expression patterns

G.1 Objectives

In all languages, MWEs comprise a wide-range of distinct but related phenomena like idioms, phrasal verbs, noun-noun compounds and many others. Even though some MWEs are fixed, and do not present internal variation such as "ad hoc", others are much more flexible and allow different degrees of internal variation and modification.

The purpose of this section is to allow a representation of the internal (semi-fixed or flexible) structure of MWEs in a given language.

G.2 Absence versus presence of MWE patterns

This section is based on the assumptions that:

- MWEs are decomposable;
- This decomposition can be described by the use of a symbolic pattern.

There is another possiblity to describe MWEs, that is in using the Inflectional Paradigm extension. But in this case, MWEs are limited to simple situations without any variation. On the contrary of this option, the current section permits to specify that a portion or the totality of the expression is to be interpreted with respect to the grammar of the language.

G.3 Description of MWE expression pattern model

MWE pattern

A *MWE Pattern* is an element that allows the description of a certain type of lexical combination phenomena. A pattern always refers to the list of components of the lexical entry. A MWE Pattern is not to be used for lexical entries that are not MWE. A pattern is described by means of Combiners.

Combiner

A *Combiner* is an element that allows the adornment of data categories in order to give details about the structure of MWEs. A *Combiner* can be connected to zero, one or several *CombinerArguments*.

Combiner Argument

A *CombinerArgument* is a smaller element information than the *Combiner* element. A *Combiner Argument* may itself be connected recursively to a *Combiner*.

G.4 Class diagram

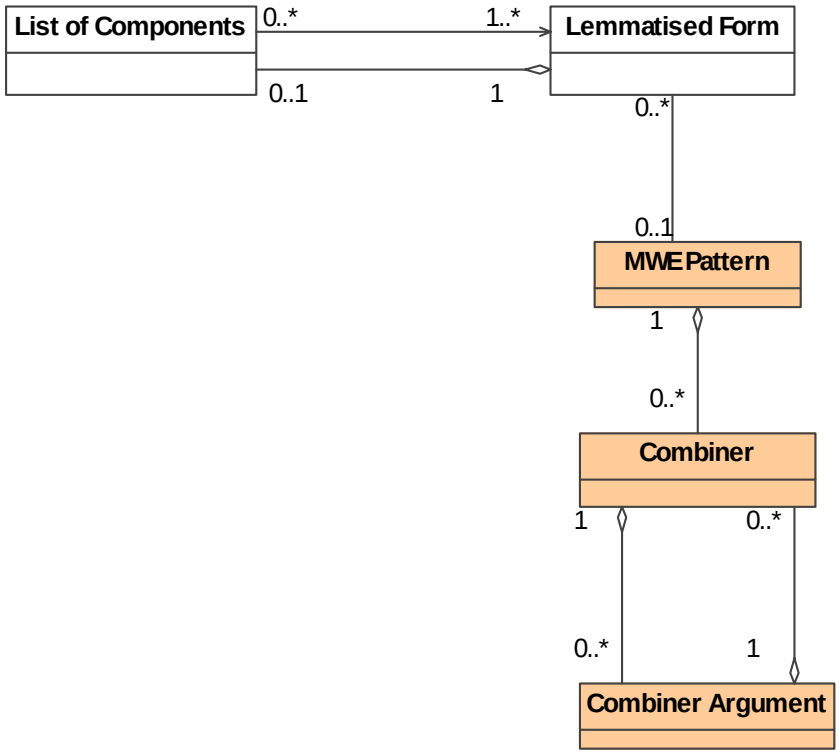


Fig G-1: MWE pattern model

Annex H (informative) Machine Readable Dictionary Examples

H.1 Introduction

This extension provides examples of how to develop Machine Readable Dictionaries MRD models and instantiations using the LMF core package and the MRD meta model extension (Annex A).

The extension illustrates the development of three types of MRD instantiations:

- A simple monolingual MRD
- A bilingual MRD with multiple representations
- A MRD for morphology that can be used either for human or machine consumption

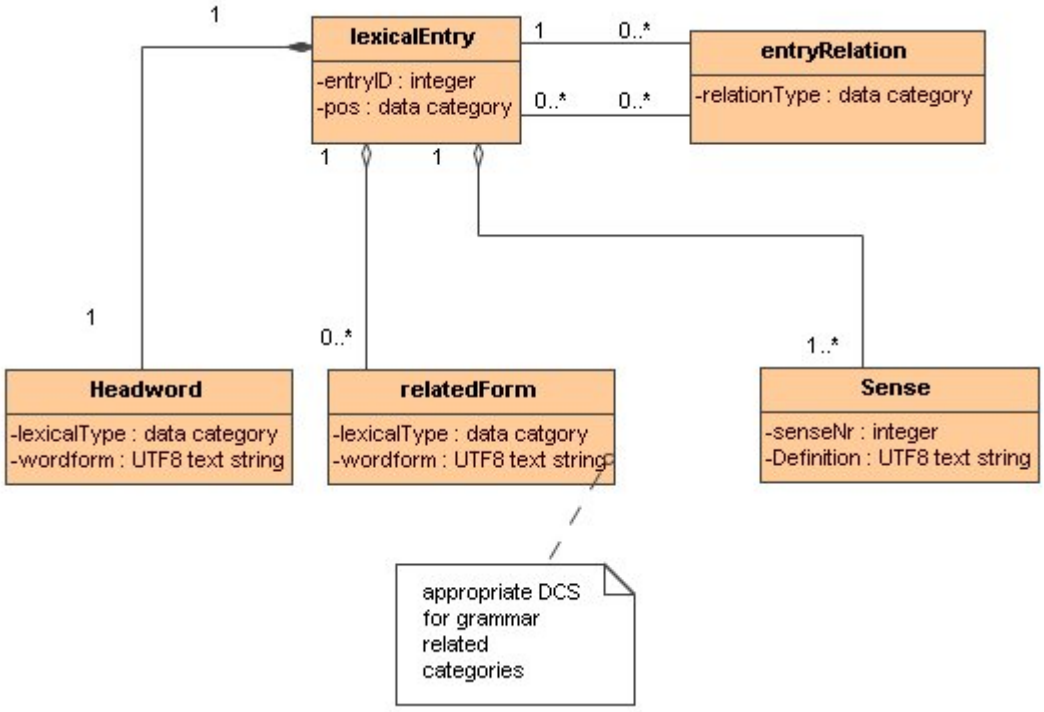
The extension will show how to tailor the core package and MRD extension meta models to meet the specific design needs of the lexicon developers using the following methods:

- Selection of a subset of classes appropriate to the design within the allowable scope of the LMF metamodels
- Modification of the associations and cardinality to meet design needs within the allowable scope of the LMF metamodels
- Data Category Selection

H.2 Example of a monolingual MRD

H.2.1 Introduction

This example assumes that the design goal is to create a very simple MRD that contains a headword, definition, related form, and cross references among headwords using the *entryRelation* class. The example illustrates the differences between the *Headword* and the *relatedForm*, and shows how the *relatedForm* and *entryRelation* can be used to achieve different design goals.



H.2.2 Class and Subclass Selection

Because the design goal is to create an English language monolingual MRD, the *representationFrame* class and the *Translation* class are not needed. The lexicon developer has also chosen not to include hierarchical senses or examples.

H.2.3 Data Category Selection

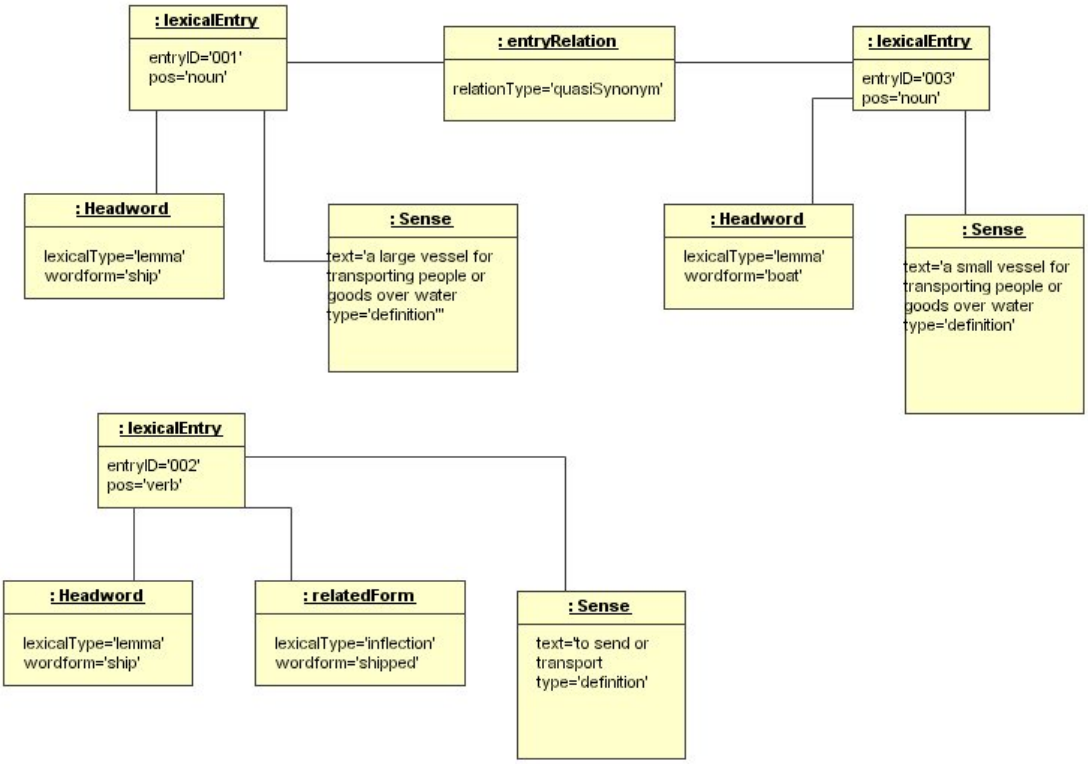
With the exception of the grammatical categories (which will vary depending on the part of speech), the Data Category Selection is relatively simple.

H.2.4 Global Design Considerations

The lexicon developer has implemented a flat structure in the lexicon design by allocating the part of speech to the Lexical Entry level, which allows homographs, synonyms, antonyms, and other related forms to be stored in separate entries. The Entry Relation class then provides a cross reference function to manage the related entries.

H.2.5 Instantiation Example

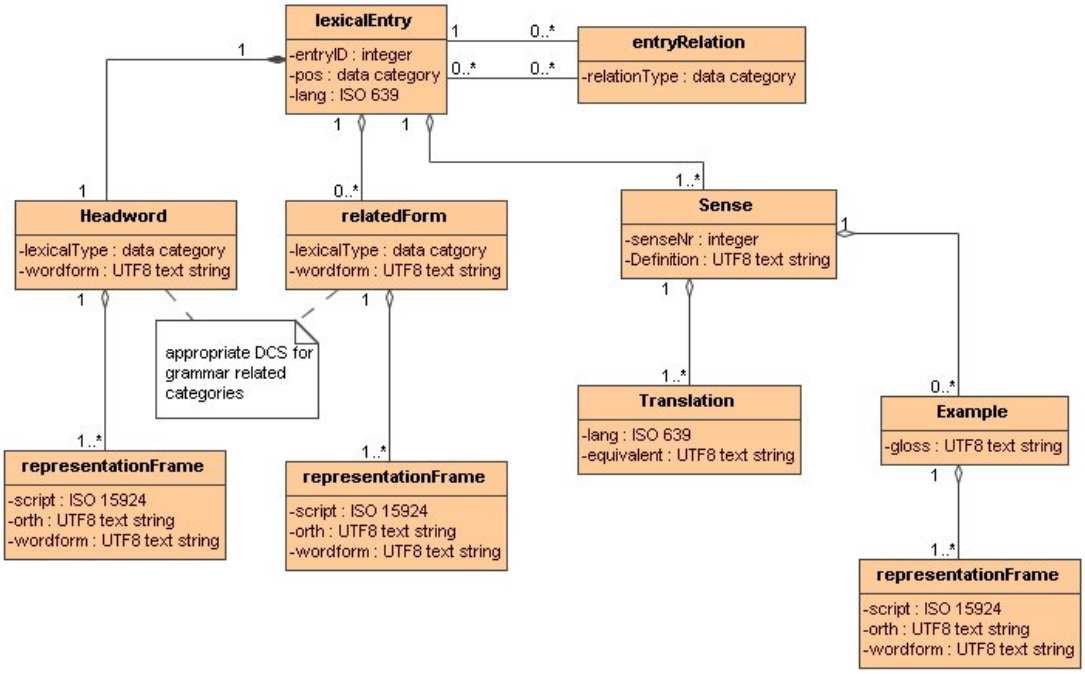
In the following example, two quasi synonyms, the common nouns, ‘ship’ and ‘boat’, are each contained in a separate entry and cross referenced through the *entryRelation*. The verb, ‘ship’, is in a separate entry that is not cross referenced through the entryRelation. The design intent is that, when implemented in a system, the capabilities of the Information Retrieval system will support the management of homographs. This design reflects an editorial choice and does not preclude the linking of homographs through the entryRelation.



H.3 Example of a Bilingual MRD with Multiple Representations

H.3.1 Introduction

This example assumes that the design goal is to create a bilingual MRD for students who need to see the word forms and examples in Arabic script, a transliteration, and a transcription.



H.3.2 Class and Subclass Selection

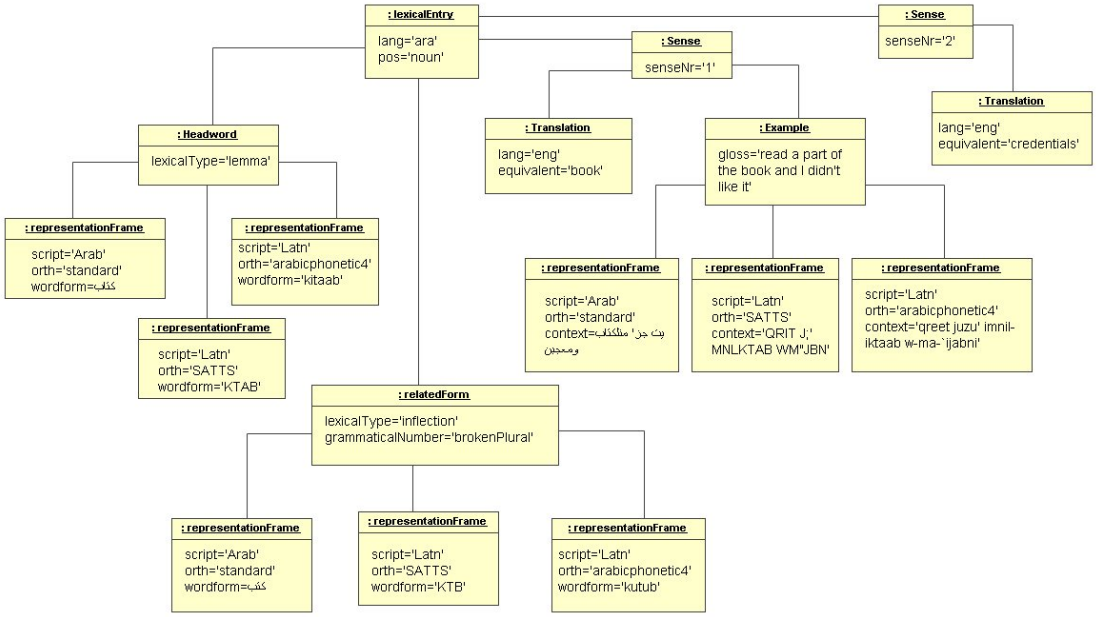
Because the design goal is to create an Arabic-English dictionary containing multiple representations, the model includes *representationFrame* class and the *Translation* class. The lexicon developer has chosen not to include hierarchical senses.

H.3.3 Data Category Selection

In order to specify the attributes of the word forms in Arabic script, the transliteration, and the transcription, the *representationFrame* includes data categories for the script and orthography. The decision to include the *representationFrame* class is an editorial choice determined by the goals of the lexicon developer. If the goal was to produce an Arabic-English MRD that contained only Arabic script for the Arabic word forms, the inclusion of *representationFrame* class would not be necessary.

H.3.4 Instantiation Example

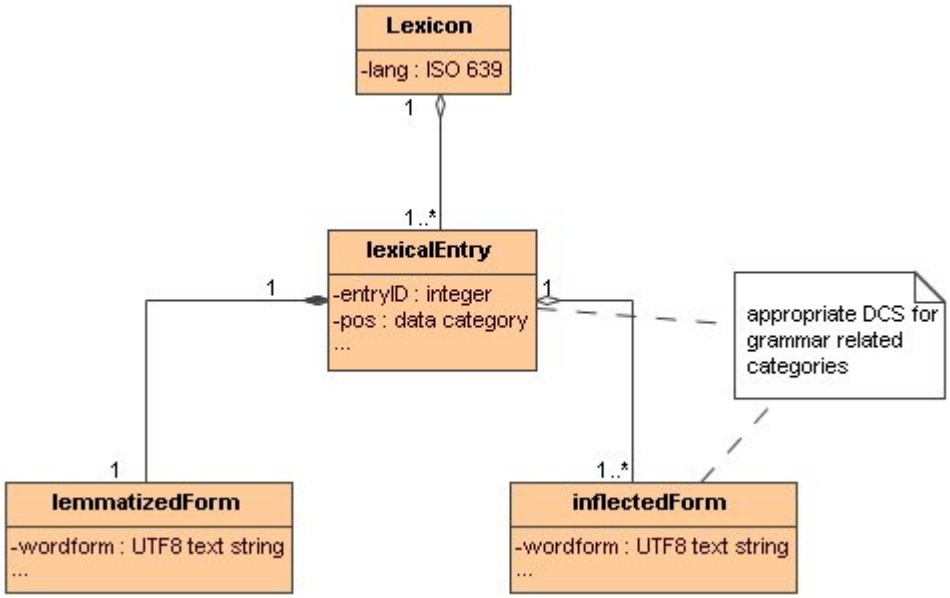
The following example shows an entry containing the Arabic word ‘kitaab’ and two equivalents in English, ‘book’ (the most common meaning) and ‘credentials’. The transliterations and transcriptions provide users more information about the pronunciation of the words and examples than can be derived from the Arabic script. In this example, the related form provides information about the form and pronunciation of the Arabic broken plural, which is an irregular inflection



H.4 MRD for Morphology

H.4.1 Introduction

This example assumes that the design goal is to create a MRD for Welsh morphology for either human or machine consumption.

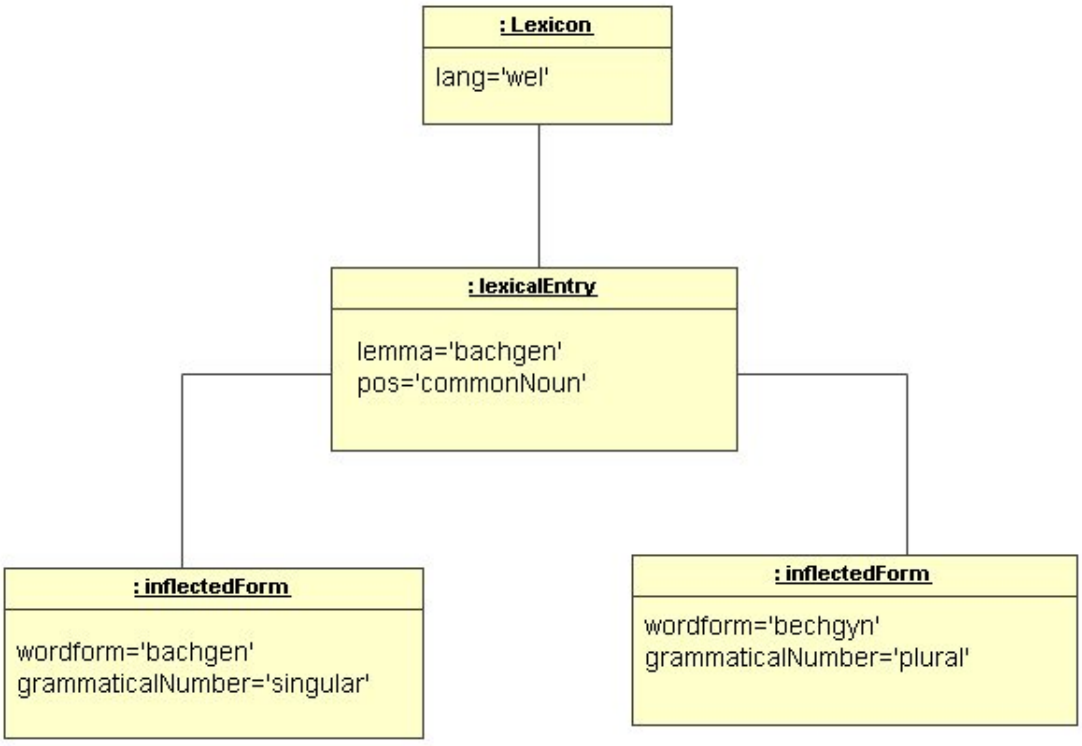


H.4.2 Design Choices

Because the range of the lexical types in a morphological lexicon is limited to the lemma and the inflected word forms, the use of the *lemmatizedForm* class and *inflectedForm* class reduces the number of data categories needed and simplifies the design. The *representationFrame* class is not needed for the Welsh morphology (but could be used for morphologies for other languages). The *Translation* class is not needed for a monolingual MRD, and the lexicon developer has chosen not to include the *Sense* class, which in a morphological lexicon would be used for informational purposes only.

H.4.3 Instantiation Example

The example shows lemma of the Welsh word for ‘boy’ and the singular and plural inflected forms of the word. Because the Lemmatized Form had no children and did not contain complex attributes, the lemma can be instantiated through a ‘lemma’ data category at the Entry Level (this could also be reflected in the model itself).



Annex I (informative) examples for NLP extensions

I.1 Extension for NLP morphology

I.1.1 Example of class adornment

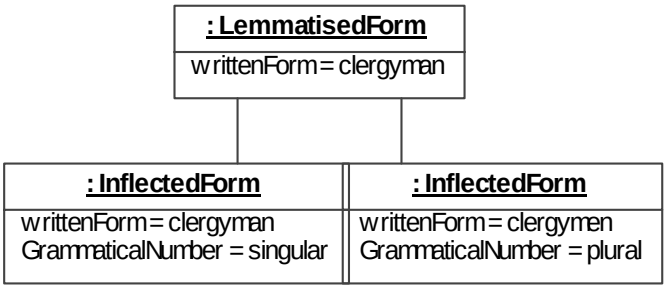
Classes may be adorned with the following attributes:

class name	example of attributes	comment
Lemmatised Form	writtenForm spokenForm transliteration	/writtenForm/ and /spokenForm/ are valued by a Unicode string. /transliteration/ specifies the type of transliteration, if any.
Stem	writtenForm spokenForm transliteration	/writtenForm/ and /spokenForm/ are valued by a Unicode string. /transliteration/ specifies the type of transliteration, if any.
InflectionalParadigm	id example	A paradigm is designed to be shared and referred, so usually, it holds an identifier.

I.1.2 Examples of word description

I.1.2.1 The English word “clergyman” without any inflectional paradigm

The following instance diagram illustrates a very simple example. The form is “clergyman” and two inflected forms are connected to this instance. The first inflected form is “clergyman” for singular and the second one is “clergymen” for plural.



I.1.2.2 The word “clergyman” with an underspecified inflectional paradigm

Regarding to the last diagram, another possibility is to use an *Inflectional Paradigm*. The *Lexical Entry* “clergyman” is declared as conforming to the *Inflectional Paradigm* “asMan”. This paradigm has a name but is not analytically described within the lexicon.

I.2 Extension for NLP syntax

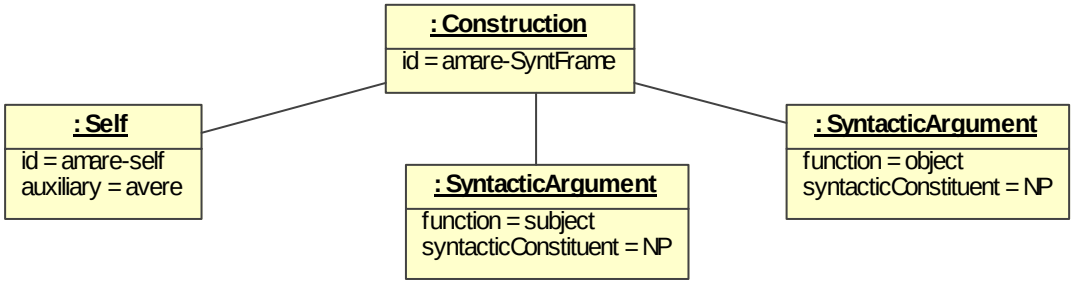
I.2.1 Example of class adornment

Classes may be adorned with the following attributes:

class name	example of attributes	comment
SyntacticBehavior	id label	
Construction	id label comment	
Self	partOfSpeech mood voice auxiliary	
SyntacticArgument	function syntacticConstituent introducer label restriction	The function may hold values like /subject/ or /object/. The constituent may hold values like /NP/ or /PP/ respectively for noun phrase and prepositional phrase. The introducer may specifier which required preposition is located at the beginning of the constituent.
ConstructionSet	id label example comment	For instance, in English, it is possible to have one *Construction Set* for ergative verbs. For “boil” in “he boils a kettle of water” and “the kettle boils”, this verb will have only one syntactic behavior (referring to a sole *Construction Set*) instead of two syntactic behaviors (one for “he boils a kettle of water” and one for “the kettle boils”).

I.2.2 Example of word description

This example is taken from the Parole/CLIPS lexicon (www.ilc.cnr.it). In this example, only syntactic structures are used, nothing in semantics is being described. This is a rather simple construction in Italian where both the subject and the direct object are Noun Phrase. The self object describes a verb that takes the auxiliary “avere”. A typical example of such a construction is “Gianni ama Maria”.



I.3 Extension for NLP semantics

I.3.1 Example of class adornment

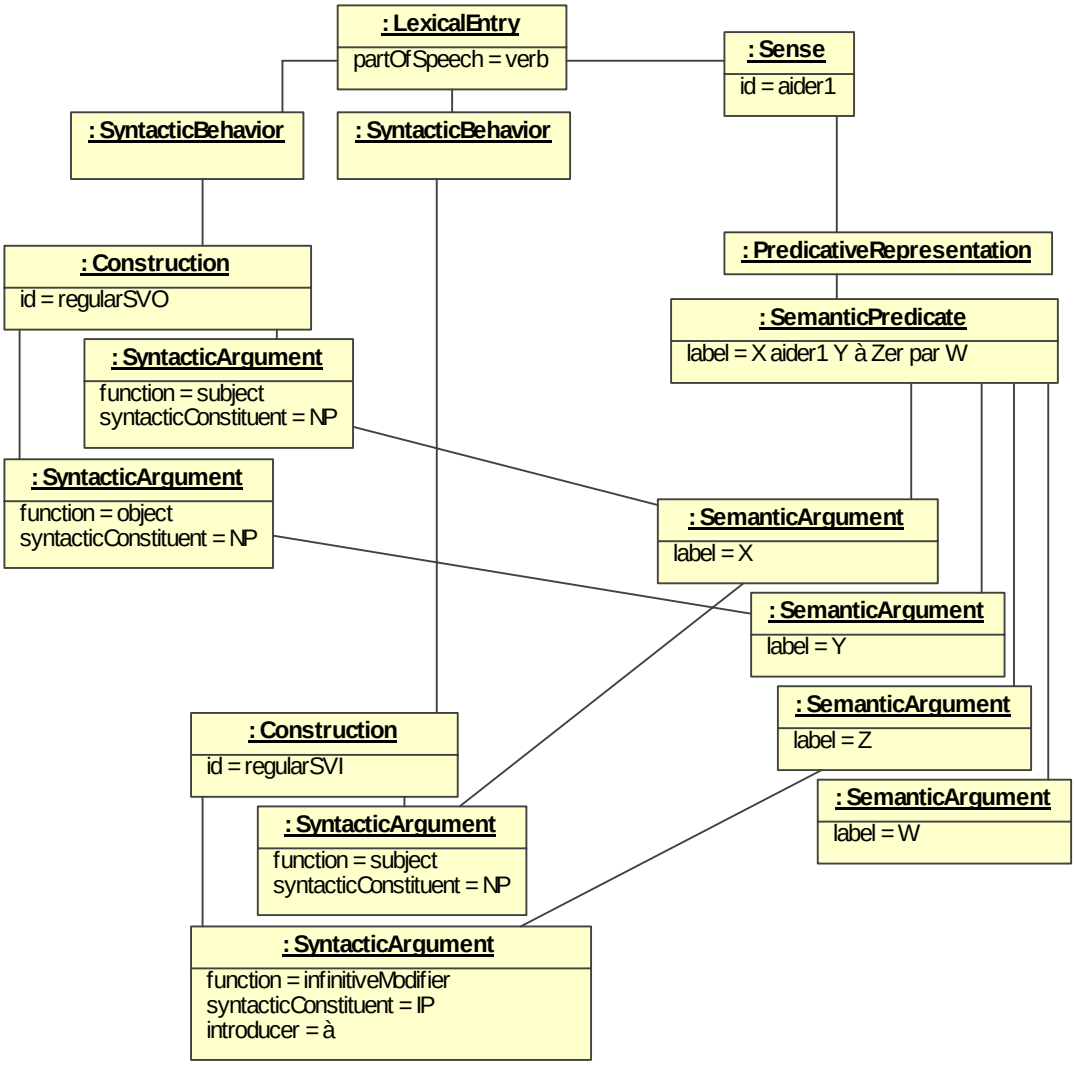
Classes may be adorned with the following attributes:

class name	example of attributes	comment
Sense	dating style frequency geography animacy	
Sense Example	text source language	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.
Semantic Definition	text source language view	
Proposition	label type text	
Semantic Predicate	label definition	
Predicative Representation	type comment	For instance, a semantic derivation between a sense of a noun and a sense of a verb can be linked to a shared predicate. In such a situation, the predicative representation of the sense of the noun can be typed as /verbNominalization/.
Semantic Argument	semanticRole restriction	
Predicate Relation	label	

	type	
Synset	label source	
Synset Relation	label type	

I.3.2 Example of word description

The following French example presents the syntax of the sense “Aider1” taken from “Dictionnaire Explicatif et Combinatoire” [2]. “Aider1” is linked to the semantic actants: “X aide Y à Z-er par W” as in “il vous aidera par son intervention à surmonter cette épreuve”. This entry yields eight different syntactic constructions. We supply the representation for the two first ones: "La Grande-Bretagne aide ses voisins" and "La Grande-Bretagne a aidé à créer l'ONU" with a special focus on syntactic and semantic representation linking. The two constructions are related to a common semantic predicate. This predicate has its semantic arguments (X, Y, Z and W) which are shown to be related to particular syntactic arguments in the different constructions of the verb. That is, the constructions are not linked directly to the predicate, but a particular syntactic argument in each construction is linked to a particular semantic argument.



I.4 Extension for NLP multilingual notations

I.4.1 Example of class adornment

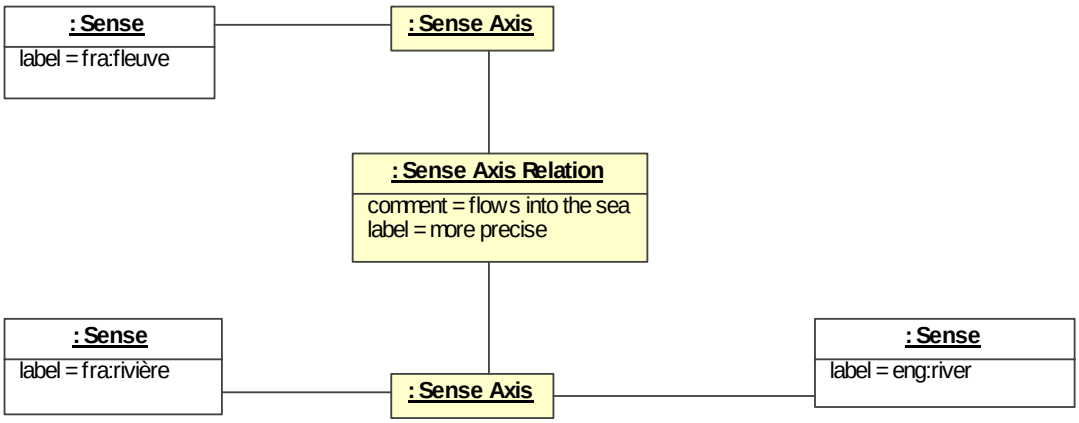
Classes may be adorned with the following attributes:

class name	example of attributes	comment
Sense Axis	label descriptiveSystem hook	A single word in the source language can be translated by a compound word into the target language. It is not the purpose of the multilingual extension to provide a complex system for knowledge representation which ideally should be structured as one or several external systems designed specifically for that purpose. However, /descriptiveSystem/ and /hook/ are provided to refer to respectively the name(s) of the external system and to

		the specific node of this given external system.
Sense Axis Relation	label view	The label enables the coding of simple interlingual relations like the specialization of “fleuve” compared to “rivière” and “river”. It is not, however, the goal of this strategy to code a complex system for knowledge representation.
Transfer Axis	label	This approach enables the translation of syntactic actants involving inversion, such as: fra:“elle me manque” => eng:“I miss her”. Due to the fact that a lexical entry can be a support verb, it is possible to represent translations that start from a plain verb (in the source language) to a support verb (in the target language) like from French to Japanese: fra:“Marie rêve” => jpn:"Marie wa yume wo miru".
Transfer Axis Relation	label variation	The element may be used to represent slight variations between closed languages. For instance, in order to represent slight variations between European Portuguese and Brazilian, different intermediate Transfer Axis can be created. The Transfer Axis relations hold a label to distinguish which one to use depending on the target language.
Source Test	text comment	
Target Test	text comment	
Example Axis	comment source	The purpose is not to record large scale multilingual corpora; the goal is to link a Lexical Entry with a typical example of translation.

I.4.2 Example of word description

This example illustrates how to use two intermediate sense axis in order represent a near match between “fleuve” in French and “river” in English. The sense axis on the top is not linked directly to any English sense because this notion does not exist in English. In the diagram, French is located on the left side and English on the right side.



I.5 Extension for NLP inflectional paradigms

I.5.1 Example of class adornment

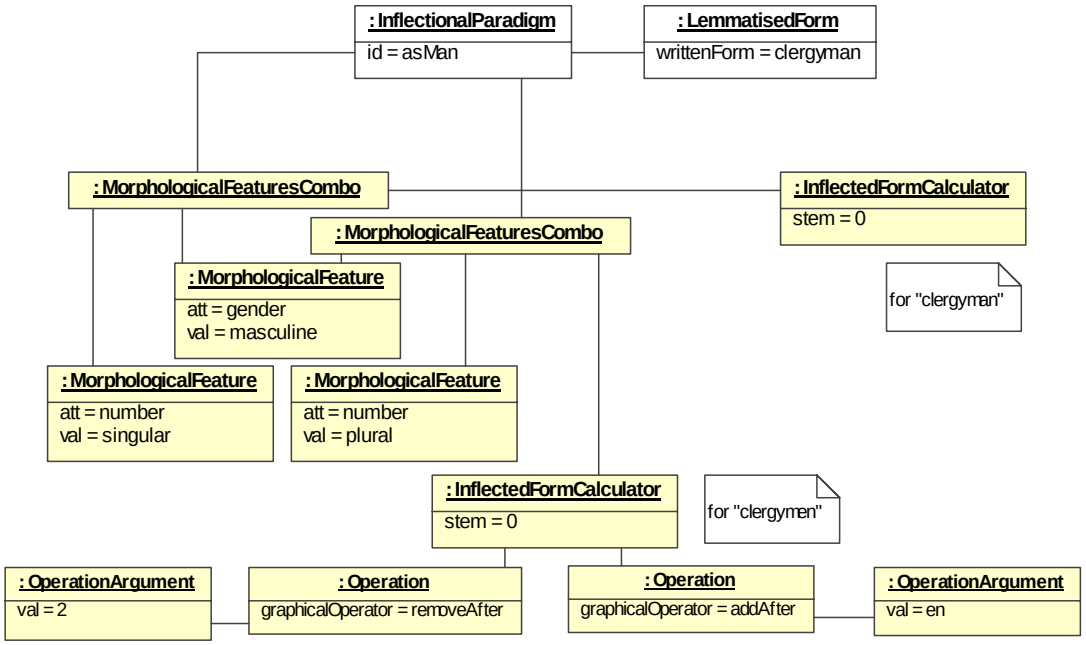
Classes may be adorned with the following attributes:

class name	example of attributes	comment
Morphological Features Combo		
Inflected Form Calculator	stem contextualVariation	/stem/ refers to the lemma or one of the stems. /contextualVariation/ may be used for instance to mark elision.
Operation	graphicalOperator phoneticOperator	The values for these attributes may be as follows: /addBefore/ meaning “add a string to the left” e.g. in German “lessen” => “gelessen”. /removeAfter/ meaning “remove N characters from the right” e.g. in French “chanter” => “chante”. /copy/ meaning “duplicate N characters from position X at position Y” e.g. the plural by means of duplication like in Indonesian “mata” (eye) => “mata-mata” (eyes).
Operation Argument	val	The following convention may be used for the position: a positive integer when starting from left and a negative integer when starting from right.
Morphological Feature	att val	The values can be for instance: /grammaticalGender/ and /feminine/
Composer	rank graphicalSeparator transformation	/rank/ refers to one of the ListOfComponents lemmatised forms.

I.5.2 Examples of word description

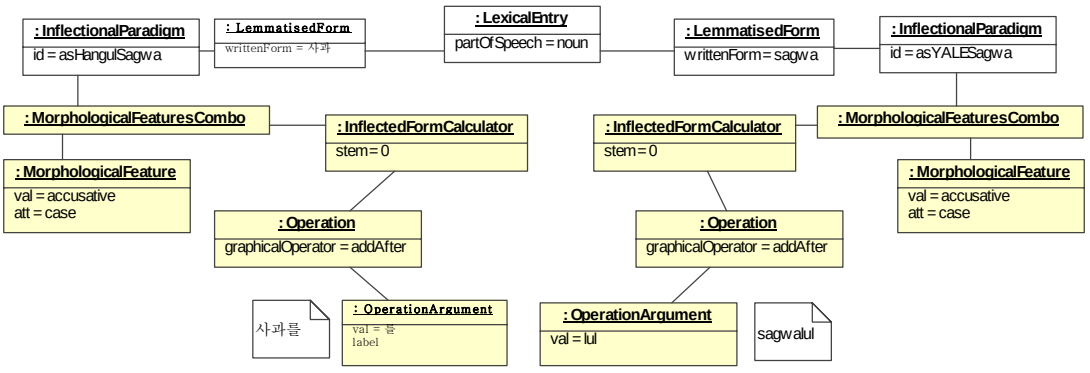
I.5.2.1 The word “clergyman” with a fully specified Inflectional Paradigm

Two letters are removed and two letters are added. The English morphology is relatively simple, so the representation is simple, which means it is not necessary to manage any stem and a reference to the lemmatised form can be used. Thus, the value for the stem attribute is zero. When applied to the entry “clergyman”, the singular gives “clergyman” and the plural gives “clergymen”.



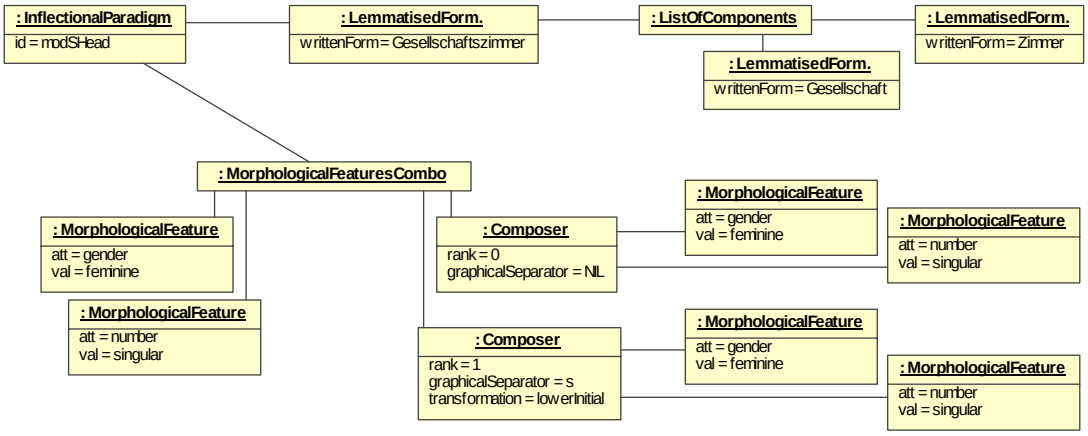
I.5.2.2 The Korean word “sagwa”

In Korean, there is more than one orthographic system and each system has its own inflection. The inflection paradigm being attached to the form, the paradigms can be different. The translation of the English word “apple” is written as “사과” in Hangul characters system and as “sagwa” in the Yale system. Korean language uses particles in order to indicate case. For instance, for accusative case, the inflected form will be “사과를” in Hangul and “sagwalul” in Yale system.



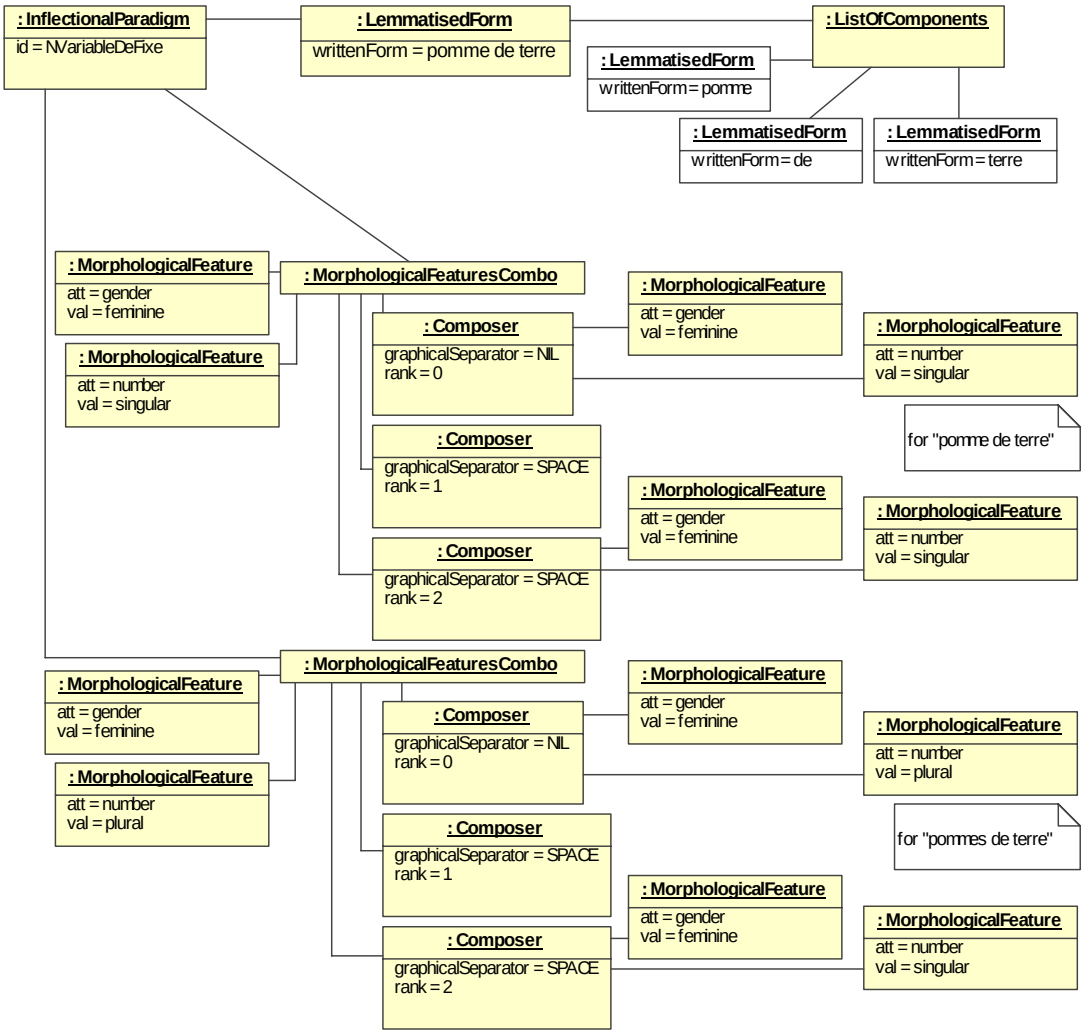
I.5.2.3 The German compound word "Gesellschaftszimmer"

This is an example of an inflectional paradigm applied to an agglutinative compound word. The inflected forms are deduced from the two components. A *Composer* specifies that an "s" is added and that the initial letter of "Zimmer" is transformed into a lower case letter.



I.5.2.4 The French MWE “pomme de terre”

This is an example of an inflectional paradigm for MWEs. The inflected forms are computed from the components of the multiword by the means of a reference to the combination of the morphological features for each of the components. Singular of “pomme de terre” is “pomme de terre”. Plural is “pommes de terre”. This is a common behavior in French for a pattern NdeN to exhibit this kind of variation on the sole head of the compound noun with a fixed modifier. The morphological feature combiners on the left side represent the number and gender of the compound. The preposition “de” is not bound to any data category because it has no morphological feature.



I.6 Extension for NLP multiword expression patterns

I.6.1 Example of class adornment

Classes may be adorned with the following attributes:

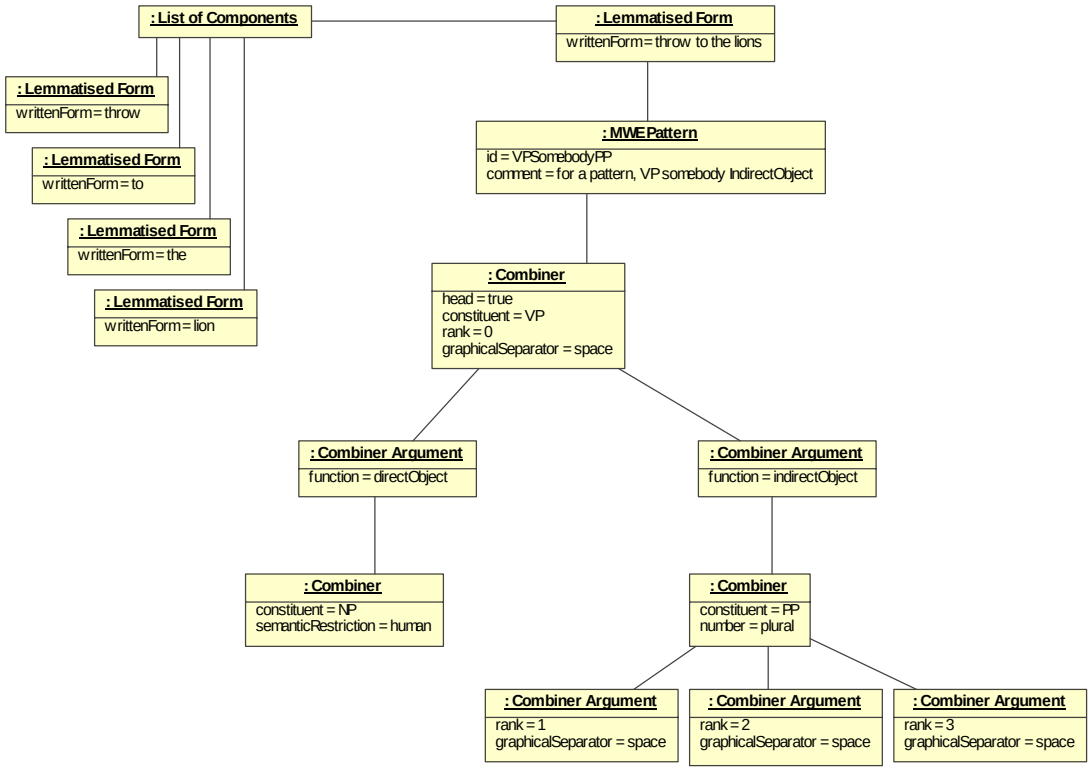
class name	example of attributes	comment
MWE Pattern	id comment	The purpose of a *MWE Pattern* is to be shared by all the lexical entries that have this structure. The objective of a pattern is to be shared, so it must be referred, so usually, it holds an identifier.
Combiner	head constituent rank graphicalSeparator	

	semanticRestriction number	
Combiner Argument	function rank	

I.6.2 Example of word description

The example is "to throw somebody to the lions". The structure contains three phrases:

- A fully specified verb phrase ("to throw"),
- A first noun phrase ("somebody"). This noun phrase is not fully specified in the sense that the only restriction that is expressed is that the head of the phrase must be of /human/ type.
- A fully specified second noun phrase ("to the lion"). This noun phrase is labelled as /plural/.



Annex J (informative) DTD for NLP

```
<?xml version='1.0' encoding="UTF-8"?>
        <!-- DTD for LMFNLP packages-->
        <!-- Core package-->
<!ELEMENT Database (DC*, Lexicon+, SenseAxis*, TransferAxis*, ExampleAxis*)>
<!ATTLIST Database
   dtdVersion CDATA #FIXED "1.0">
<!ELEMENT Lexicon (LexiconInformation, LexicalEntry+,  InflectionalParadigm*, MWEPattern*,
            Construction*, ConstructionSet*, SemanticPredicate*, Synset*)>
<!ELEMENT LexiconInformation (DC*)>
<!ELEMENT LexicalEntry (DC*, LemmatisedForm+, Sense*, EntryRelation*, SyntacticBehavior*)>
<!ATTLIST LexicalEntry
   id       ID #IMPLIED>
<!ELEMENT Sense (DC*, SenseRelation*, PredicativeRepresentation*, SenseExample*, SemanticDefinition*)>
<!ATTLIST Sense
   id       ID #IMPLIED
   inherit  IDREFS #IMPLIED>
<!ELEMENT EntryRelation (DC*)>
<!ATTLIST EntryRelation
   targets  IDREFS #REQUIRED>
<!ELEMENT SenseRelation (DC*)>
<!ATTLIST SenseRelation
   targets  IDREFS #REQUIRED>
        <!-- Package for Morphology -->
<!ELEMENT LemmatisedForm (DC*, ListOfComponents?, InflectedForm*, Stem*)>
<!ATTLIST LemmatisedForm
   id       ID #IMPLIED
   paradigm IDREF #IMPLIED
   pattern  IDREF #IMPLIED>
<!ELEMENT ListOfComponents (DC*)>
<!ATTLIST ListOfComponents
   targets  IDREFS #REQUIRED>
<!ELEMENT InflectedForm (DC*)>
<!ELEMENT Stem (DC*)>
        <!-- Package for inflectional paradigms -->
<!ELEMENT InflectionalParadigm (DC*, MorphologicalFeaturesCombo*)>
<!ATTLIST InflectionalParadigm
   id       ID #REQUIRED>
<!ELEMENT MorphologicalFeaturesCombo (DC*, Composer*, InflectedFormCalculator*,
                        MorphologicalFeature*)>
<!ELEMENT Composer (DC*, MorphologicalFeature*)>
<!ELEMENT InflectedFormCalculator (DC*, Operation*)>
<!ELEMENT Operation (DC*, OperationArgument*)>
<!ELEMENT OperationArgument (DC*)>
```

```
<!ELEMENT MorphologicalFeature EMPTY>
<!ATTLIST MorphologicalFeature
   att      CDATA #REQUIRED
   val      CDATA #REQUIRED>
        <!-- Package for MWE patterns -->
<!ELEMENT MWEPattern (DC*, Combiner*)>
<!ELEMENT Combiner (DC*, CombinerArgument*)>
<!ELEMENT CombinerArgument (DC*, Combiner*)>
        <!-- Package for Syntax -->
<!ELEMENT SyntacticBehavior (DC*)>
<!ATTLIST SyntacticBehavior
   id             ID #IMPLIED
   senses         IDREFS #IMPLIED
   constructions    IDREFS #IMPLIED
   constructionsets IDREFS #IMPLIED>
<!ELEMENT Construction (DC*, Self?, SyntacticArgument*)>
<!ATTLIST Construction
   id             ID #IMPLIED
   inherit        IDREFS #IMPLIED>
<!ELEMENT Self (DC*)>
<!ELEMENT SyntacticArgument (DC*)>
<!ATTLIST SyntacticArgument
   target         IDREF #IMPLIED
   semargs        IDREFS #IMPLIED>
<!ELEMENT ConstructionSet (DC*)>
<!ATTLIST ConstructionSet
   id             ID #IMPLIED
   constructions    IDREFS #IMPLIED
   inherit        IDREFS #IMPLIED>
        <!-- Package for Semantics -->
<!ELEMENT PredicativeRepresentation (DC*, SemanticPredicate*)>
<!ELEMENT SemanticPredicate (DC*, SemanticArgument*, PredicateRelation*)>
<!ATTLIST SemanticPredicate
   id             ID #REQUIRED>
<!ELEMENT SemanticArgument (DC*)>
<!ATTLIST SemanticArgument
   id             ID #REQUIRED>
<!ELEMENT PredicateRelation (DC*)>
<!ATTLIST PredicateRelation
   targets        IDREFS #IMPLIED>
<!ELEMENT SenseExample (DC*)>
<!ATTLIST SenseExample
   id             ID #IMPLIED>
<!ELEMENT SemanticDefinition (DC*, Proposition*)>
<!ELEMENT Proposition (DC*)>
<!ELEMENT Synset (DC*, SemanticDefinition*, SynsetRelation*)>
<!ATTLIST Synset
```

```
   id              ID #IMPLIED>
<!ELEMENT SynsetRelation (DC*)>
<!ATTLIST SynsetRelation
   targets         IDREFS #IMPLIED>
         <!-- Package for Multilingual notations -->
<!ELEMENT SenseAxis (DC*, SenseAxisRelation*)>
<!ATTLIST SenseAxis
   id              ID #IMPLIED
   senses          IDREFS #IMPLIED
   synsets         IDREFS #IMPLIED>
<!ELEMENT SenseAxisRelation (DC*)>
<!ATTLIST SenseAxisRelation
   targets         IDREFS #REQUIRED>
<!ELEMENT TransferAxis (DC*, TransferAxisRelation*,
                    SourceTest*, TargetTest*)>
<!ATTLIST TransferAxis
   id              ID #IMPLIED
   synbehaviors IDREFS #IMPLIED>
<!ELEMENT TransferAxisRelation (DC*)>
<!ATTLIST TransferAxisRelation
   targets         IDREFS #REQUIRED>
<!ELEMENT SourceTest (DC*)>
<!ELEMENT TargetTest (DC*)>
<!ELEMENT ExampleAxis (DC*)>
<!ATTLIST ExampleAxis
   examples IDREFS #IMPLIED>
         <!-- for datcat adornment -->
<!ELEMENT DC EMPTY>
         <!-- att=constant to be taken from the DCR -->
         <!-- val=free string or constant to be taken from the DCR-->
<!ATTLIST DC
   att      CDATA #REQUIRED
   val      CDATA #REQUIRED>
```

Bibliography

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

[2] Mel'cuk I., Clas A., Polguère A. 1995 Introduction à la lexicologie explicative et combinatoire. Duculot. Bruxelles.