International Standard

ISO 24613-1

Second edition
2024-01

Language resource management — Lexical markup framework (LMF) —

Part 1: Core model

Gestion des ressources linguistiques — Cadre de balisage lexical (LMF) —

Partie 1: Modèle de base

Reference number
ISO 24613-1:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 37, *Language and terminology*, Subcommittee SC 4, *Language resource management*.

This second edition cancels and replaces the first edition (ISO 24613-1:2019), which has been technically revised.

The main changes are as follows:

— several changes have been made to Figure 1 “LMF core package”, as follows:

  — the OrthographicRepresentation class associations with the Form and Definition classes previously had a cardinality of 1 to 1, which did not correctly represent the intent of the UML model; the revision of the cardinality to 1 to 0..* in each case now provides a correct model;

  — the type: intern/extern attribute-value pair is no longer included in the CrossREF class since it described linking processes relevant for implementations, not associations relevant for a metamodel;

  — the full names relationship values in the CrossREF class, “synonym/composition” replace the abbreviations, “syn/compo”;

  — the class names in Figure 1 are now harmonized with the LMF style;

— relevant information has been moved from the tables in ISO 24613-2:2020 to Table A.1, meaning that the latter now contains more complete examples of values and attributes allocated to classes first introduced in this document.

A list of all parts in the ISO 24613 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Optimizing the production, maintenance and extension of electronic lexical resources is one of the crucial aspects impacting human language technologies (HLTs) 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. 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 instances, including morphological, syntactic and semantic aspects.

The goals of LMF are:

— to provide a common model for the creation and use of electronic lexical resources ranging from small to large in scale;

— to manage the exchange of data between and among these resources; and

— to facilitate the merging of large numbers of different individual electronic resources to form extensive global electronic resources.

The ultimate goal of LMF is to create a modular structure that will facilitate true content interoperability across all aspects of electronic lexical resources.

LMF supports existing lexical resource models such as Genelex,[5] the EAGLES International Standard for Language Engineering (ISLE),[6] Multilingual ISLE Lexical Entry (MILE) models,[12] Text Encoding Initiative (TEI) guidelines,[10] Ontolex[9] and the Language Base Exchange (LBX) serialization together with the US Government Wordscape On-Line Dictionary system[7].

LMF uses unified modelling language (UML) modelling processes.[11] The LMF core package describes the basic hierarchy of information of a lexical entry, including information on the word form. The core package 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 package in additional LMF parts. See Annex A for data category examples);

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

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

— the vocabularies used by LMF that describe how to extend LMF through linkage to a variety of specific resources (extensions) and methods for analysing and designing such linked systems.

LMF parts 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 resource.

The ISO 24613 series is designed to coordinate closely with ISO 16642.

Language resource management — Lexical markup framework (LMF) —

Part 1: Core model

1 Scope

This document establishes the core model of the lexical markup framework (LMF), a metamodel for representing data in monolingual and multilingual lexical resources used with computer applications.

LMF provides mechanisms that allow the development and integration of a variety of electronic lexical resource types.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 639, *Code for individual languages and language groups*

ISO 15924, *Information and documentation — Codes for the representation of names of scripts*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
data category
DC
class of data items that are closely related from a formal or semantic point of view

EXAMPLE /part of speech/, /subject field/, /definition/.

Note 1 to entry: A data category can be viewed as a generalization of the notion of a field in a database.

Note 2 to entry: In running text, such as this document, data category names are enclosed in forward slashes (e.g. /part of speech/).

[SOURCE: ISO 30042:2019, 3.8, modified — admitted term “DC” added.]

3.2
word form
instantiation of a *lexeme* (3.5) in a syntactic context

3.3
grammatical feature
property associated with a *word form* (3.2) to describe one of its grammatical attributes

EXAMPLE grammaticalGender.

3.4
lemma
lemmatized form
canonical form
word form (3.2) chosen to represent a *lexeme* (3.5)

Note 1 to entry: In many European languages, the lemma is usually the singular for a noun 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. In Arabic, for a verb, the lemma is sometimes considered as being the third person singular with the accomplished aspect. In other approaches it is considered as being the root.

3.5
lexeme
abstract unit generally associated with a set of *word forms* (3.2) sharing common properties, such as morphologic, morphosyntactic, semantic, or phonetic properties

3.6
lexical resource
lexical database
database consisting of one or several *lexicons* (3.7)

3.7
lexicon
resource comprising lexical entries for one or several languages

Note 1 to entry: A special language lexicon or a lexicon prepared for a specific *natural language processing* (3.9) application can comprise a specific subset of a language.

3.8
multiword expression
MWE
lexeme (3.5) made up of a sequence of two or more lexemes that has properties that are not necessarily predictable from the properties of the individual lexemes or their normal mode of combination

EXAMPLE "To kick the bucket", an idiomatic expression which means to die rather than to hit a bucket with one's foot. An idiomatic expression is a subtype of MWE whose properties are not predictable from the properties of the individual lexemes.

Note 1 to entry: An MWE can be a compound, a fragment of a sentence or a sentence. The group of lexemes making up an MWE can be continuous or discontinuous. It is not always possible to mark an MWE with a *part of speech* (3.11).

3.9
natural language processing
NLP
computer science field covering knowledge and techniques involved in the processing and analysis of linguistic data by a computer

3.10
orthography
systematic way of spelling or writing *lexemes* (3.5) that conforms to a conventionalized use

Note 1 to entry: Usually, the notion of orthography covers standardized spellings of alphabetic languages, such as standard UK or US English, or reformed German spelling, as well as hieroglyphic or syllabic writing systems. For the purpose of this document, variations such as transliterations of languages in non-native *scripts* (3.12), stenographic renderings or representations in the International Phonetic Alphabet are also subsumed under the notion of orthography.

3.11
part of speech
lexical category
word class
category assigned to a *lexeme* (3.5) based on its grammatical properties

EXAMPLE Typical parts of speech for European languages include noun, verb, adjective, adverb, preposition, etc.

3.12
script
set of graphic characters used for the written form of one or more languages

EXAMPLE Hiragana, Katakana, Latin, Cyrillic.

Note 1 to entry: The description of scripts ranges from a high-level classification such as hieroglyphic or syllabic writing systems versus alphabets to a more precise classification like Roman versus Cyrillic. Scripts are defined by a list of values taken from ISO 15924.

[SOURCE: ISO/IEC 10646:2020, 3.48, modified — Example and Note 1 to entry have been added.]

4 Key standards used by LMF

4.1 Unicode

LMF is Unicode-compliant and presumes that all data are used according to the Unicode character encodings specified in ISO/IEC 10646.

4.2 Language coding

Language identifiers used in LMF-compliant resources shall conform to criteria specified in ISO 639. Some issues involving the combination of language and country codes have been addressed in external standards supported by the technology community. The current edition of IETF Best Common Practices (BCP) 47[8] should be consulted.

4.3 Script coding

When the script code is not part of the language identifier, script identifiers shall conform to criteria specified in ISO 15924.

4.4 Unified modelling language

LMF complies with the specifications and modelling principles of UML as defined by the Object Management Group (OMG).[11] LMF uses a subset of UML that is relevant for linguistic description (see ISO/IEC 19505-1 and ISO/IEC 19505-2).

5 The LMF model

5.1 General

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

In this process, lexicon developers shall use the classes that are specified in the LMF core package (see 5.3), and classes that are defined in other LMF parts or classes derived from any of these referenced classes using

documented LMF processes for class inheritance. Developers shall define a data category selection (DCS) as specified for LMF DCS procedures (see 5.2.3 and 5.2.4).

5.2 Class inheritance and data category selection procedures

5.2.1 Class inheritance

LMF specifies constraints on which classes allow subclasses.

5.2.2 LMF attributes

UML models such as LMF are populated or further described by UML attributes, which provide information about specific properties or characteristics associated with the model. All LMF attributes are complex data categories. For a given class, all attributes are different. Each value of an attribute is either a simple data category or a Unicode string. Each attribute has only one value.

5.2.3 Data category selection (DCS)

In the broadest sense, a DCS can comprise all the data categories used by a given domain in the field of language resources. A DCS can also list and describe 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.

5.2.4 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.

5.3 LMF core package

5.3.1 General

The LMF core package is a metamodel that provides a flexible basis for building LMF models and extensions, see Figure 1.

NOTE Each word in a class name begins with a capital letter with no intervening spaces or punctuation. This practice is not required by UML, but generally conforms with most UML documentation.

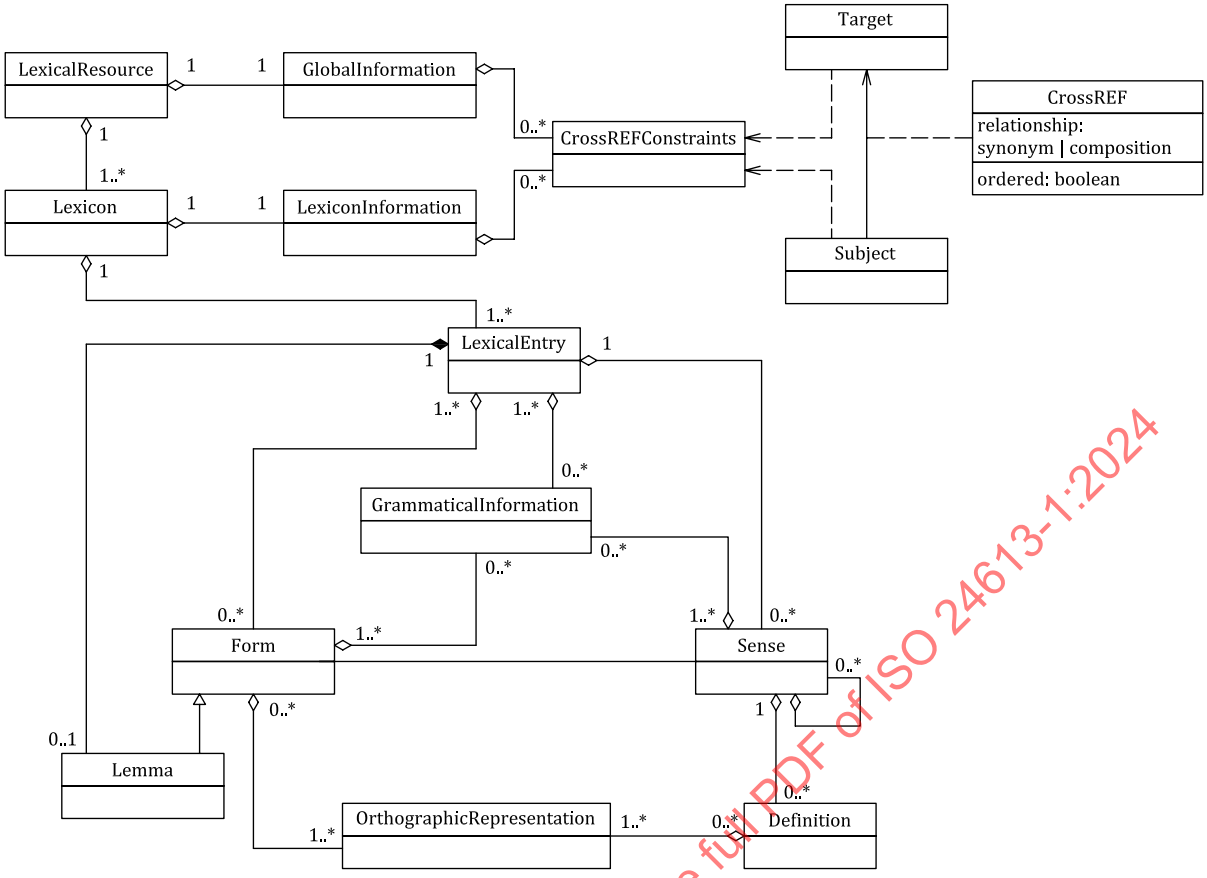

Figure 1 — LMF core package

5.3.2 LexicalResource class

LexicalResource is a class representing the entire resource. LexicalResource occurs once and only once. The LexicalResource instance is a container for one or more lexicons.

5.3.3 GlobalInformation class

GlobalInformation is a class representing administrative information and other general attributes. There is a one-to-one aggregate association between the Lexicon class and the GlobalInformation class in that the latter describes the administrative information and general attributes of the entire resource. The GlobalInformation class does not allow subclasses.

The GlobalInformation instance shall contain at least the following attributes:

— languageCoding: this attribute specifies which standard is used in order to code the language names within the whole LexicalResource instance.

The GlobalInformation instance can contain the following attributes:

— scriptCoding: this attribute describes the subset of script codings defined in ISO 15924 that are used within the whole LexicalResource instance;

— characterCoding: this attribute specifies which Unicode version is used within the whole LexicalResource instance.

NOTE Other standard related precisions can be specified on the GlobalInformation instance.

5.3.4 Lexicon class

Lexicon is a class containing one or more lexical entries. The Lexicon class does not allow subclasses.

EXAMPLE A lexicon can include Spanish language entries in the domain of graphic arts, or a lexicon can include Spanish language entries across multiple domains.

5.3.5 LexiconInformation class

LexiconInformation is a class representing administrative information and general attributes for a given Lexicon. There is a one-to-one aggregate association between the LexiconInformation class and the Lexicon class in that the former describes administrative information and attributes applicable to the entire lexicon. The LexiconInformation class does not allow subclasses.

EXAMPLE The LexiconInformation class can manage xml:lang, script and notation attributes when these are generally applicable to OrthographicRepresentation subclasses across the entire lexicon.

5.3.6 LexicalEntry class

The LexicalEntry class serves as a container for managing Form and Sense classes which have a close conceptual relationship. The derivation of subclasses and new classes in future parts will require modelling specifications that describe the associations between the forms and their related senses whenever these diverge from the close conceptual relation defined in the core model. A LexicalEntry instance can contain zero or more forms and can have zero or more different senses. The LexicalEntry class allows subclasses.

EXAMPLE A bilingual dictionary usually requires at least one Sense class, while a lexicon that explicitly describes inflected forms of a language [e.g. a lexicon containing all of or a substantial portion of the inflected forms of each lemma or root in the lexicon (extensional morphology)] does not require a Sense class, depending on the design goals of the developer.

5.3.7 Form class

The Form class groups and manages all the information about the written and spoken forms of a word, multiword expression, root, stem or morpheme. The Form class attributes are managed through a DCS process using data categories that describe the lexical features of the form (e.g. lexical role, such as lemma; lexical extent, such as word or affix; and other features, such as idiom). A Form class may be associated with one or more LexicalEntry classes. The Form class allows subclasses.

EXAMPLE Many LexicalEntry instances can contain the same Arabic root.

5.3.8 OrthographicRepresentation class

The OrthographicRepresentation class represents one or more variant orthographies of a word form or the textual definition of a sense. The OrthographicRepresentation class contains a Unicode string representing the form as well as, and if needed, unique attribute-value pairs (complex data categories) that describe the specific language, script, representationType (e.g. pronunciation, syllabification, transcription, transliteration) and different types of orthographic variants (i.e. a named orthographic subset).

5.3.9 GrammaticalInformation class

The GrammaticalInformation class groups and manages all the grammatical information (e.g. part of speech, grammatical features) that can be in aggregate association with the LexicalEntry, Form or Sense classes. The GrammaticalInformation class can contain or reference a grammatical feature structure. The GrammaticalInformation class allows subclasses.

5.3.10 Sense class

The Sense class represents one meaning of a lexical entry. The Sense class also allows for hierarchical senses in that one sense can be more specific than another sense of the same lexical entry.

5.3.11 Definition class

The Definition class contains a narrative description of a sense. A Sense instance can have zero or more Definition instances. Each Definition instance can be associated with one or more OrthographicRepresentation instances in order to manage the text definition in more than one language or script. The narrative description can be expressed in a different language and/or script than the one for the LexicalEntry instance.

EXAMPLE In a LexicalEntry for “abbess”, the narrative description can be “woman who is in charge of a convent”.

NOTE The purpose of allowing zero or more instances of a Definition class is to enable flexible modelling through other potential child classes in future parts.

5.4 Cross reference (CrossREF) model

5.4.1 General

The CrossREF package supports methods for constraining the links and the selection of linked classes in order to support specific design goals; see Figure 1.

5.4.2 CrossREF class

The CrossREF class is an association class that models the association from a Subject class to a Target class. The CrossREF class contains data categories that describe the terms and conditions of the association. The LexicalRelationshipType describes lexical or syntactic relationships (e.g. synonym, antonym, variant). Ordered is a Boolean data type setting the condition for two or more related associations that are in ordered sequence (e.g. components of an MWE). The CrossREF class can allow other data categories.

5.4.3 CrossREFConstraint class

The Subject and Target classes are superclasses whose generalizations can include classes that have potential lexical, syntactic or semantic relationships, such as the LexicalEntry, Form, Sense and Definition classes. The CrossREFConstraint class defines the allowable class instances and class associations of those classes for any design. The CrossREFConstraint class can also describe the IDType value for each defined association. The CrossREFConstraint class is in a zero-to-one aggregate association with the GlobalInformation and/or LexiconInformation class.

EXAMPLE IDType values can include internationalized resource identifier (IRI), uniform resource identifier (URI), uniform resource locator (URL) and possibly other types, including user-defined types.

5.5 Methods for data category selection and subclass creation

5.5.1 General

DCS and generalization (subclass creation) are interdependent mechanisms for building LMF models. Both mechanisms involve the allocation of data categories (attributes and values) and associations (including links) to classes to define the features of those classes.

5.5.2 Generalization

LMF uses the allocation of attributes and allowable values (data categories), associations, and links to redefine a superclass into a set of subclasses. For example, in Figure 2, ClassFeatures is a metaclass associated with an abstract superclass. The discriminator, a modelling process rather than a formal part of the metamodel, contains attributes or rules that denote all the combinations of attributes, values, associations and links common to two or more abstract subclasses. Each subclass represents a unique lexical concept characterized by the set of allocated features. Using this metamodel, the Superclass class can be

realized in a concrete superclass (e.g. Form), and each Subclass class can be realized in a concrete subclass that represents a lexical concept (e.g. Lemma).

NOTE ClassFeatures, a metaclass, does not contain the feature descriptions (data categories and association descriptions). The features are expressed through the instantiation of a concrete superclass and its subclasses.

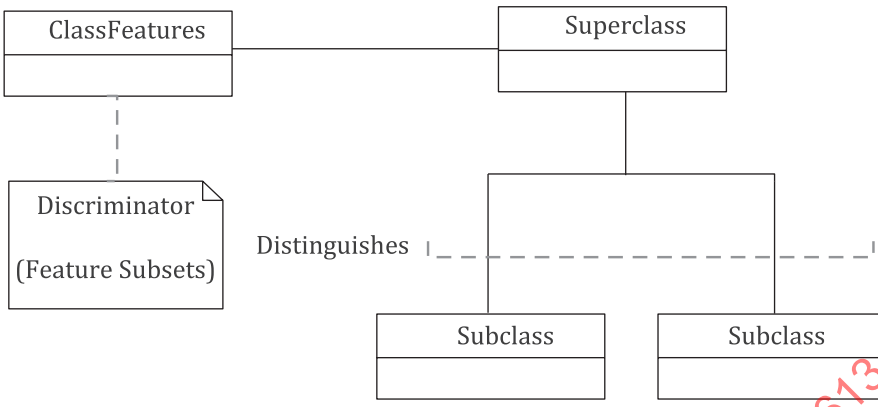

Figure 2 — Methods for data category selection and subclass creation

5.5.3 Object instantiation

Generalization enables design flexibility and provides a means for more easily verifying equivalences between different LMF realizations. There are two different processes at work: data modelling and the application of instantiation mechanisms that will vary depending on the serialization used. The pivotal role of well-defined discriminators, including data category allocation, in these mechanisms is the key to the consistent verification of model equivalency.

EXAMPLE In an XML serialization, a designer can use an eponymous Lemma subclass to instantiate a Lemma object, or the designer can assign a type with the value Lemma to a Form object. In both cases, the two objects are equivalent and both designs use DCS. In the latter case, a combination of the superclass name and the data category value fully describes the object and its assumed subclass.

NOTE The choice of serialization (e.g. XML, Jason) can limit the mechanism utilization to the technical features inherent in the implementation choice.

5.5.4 Design choices

Characteristics to be evaluated in determining design-specific data architectures include:

— data complexity and variability;

— language resource scope, complexity and variability (e.g. single lexicon, large scale database);

— system interoperability and data exchange;

— reuse.

Models that make a more extensive use of DCS are generally more suitable for describing complex, variable data, while models that make a more extensive use of subclasses can be better suited for less complex, less variable data (e.g. a monolingual lexicon or lexical resource).

5.5.5 Data categories for orthographic representation

Data categories for orthographic representation should be as comprehensive as needed to support design goals related to human understanding and machine processing. The LMF metamodel provides flexibility in how the data category allocation is applied to appropriate classes at one or more structural levels. Approaches include populating the OrthographicRepresentation class or its subclasses, or managing these

data categories in the LexicalInformation class when the data categories have global applicability across the lexicon.

5.5.6 Principles for model simplification

The requirement for model complexity is directly related to the design objectives, including type of lexicon (e.g. monolingual lexicons, bilingual lexicons), completeness and quality of data and metadata, and support for software application development (e.g. indexing, query strategies, results displays). Methods for simplifying models include:

— creating subclasses that entail a specific value-attribute pair;

— creating subclass names that are meaningful for the consumer (e.g. human translator, machine process);

— populating the model at the appropriate level (see above);

— using abbreviated class labels as an alias;

— realizing containers as value-attributes in a parent class while maintaining the conceptual relationships inherent in the metamodel.

5.6 LMF extension use

5.6.1 General

All extensions to the LMF model shall conform to this document, the core package, such that each extension shall be anchored in the set or a subset of the core package classes, or the core package, and a combination of existing extension mechanisms.

An extension shall not be used to represent lexical data independently of the core package. From the point of view of UML, an extension is a UML package. The dependencies of the various extensions are specified in Figure 3.

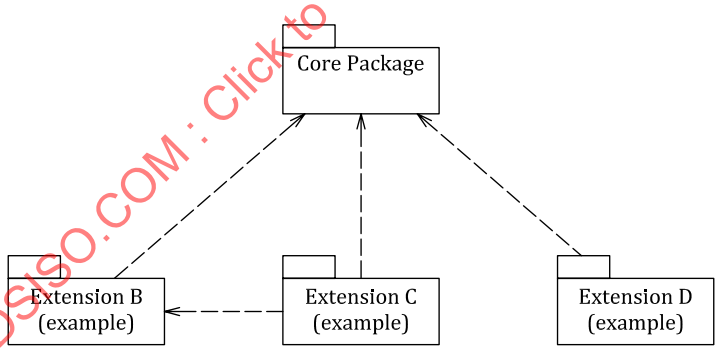

Figure 3 — Dependencies between the LMF core and sample extension packages

The extension mechanisms shall include:

— the creation of subclasses based on UML modelling principles;

— the addition of new classes;

— the selection of a subset of available core classes that are meaningful for a particular extension;

— constraints on the cardinality and type of associations;

— specification of different anchor points for associations;

— DCSs.

An LMF-conformant lexicon is defined as the combination of an LMF core package, zero to many lexical extensions and a set of data categories. The combination of all these elements is described in the UML activity diagram in Figure 4.

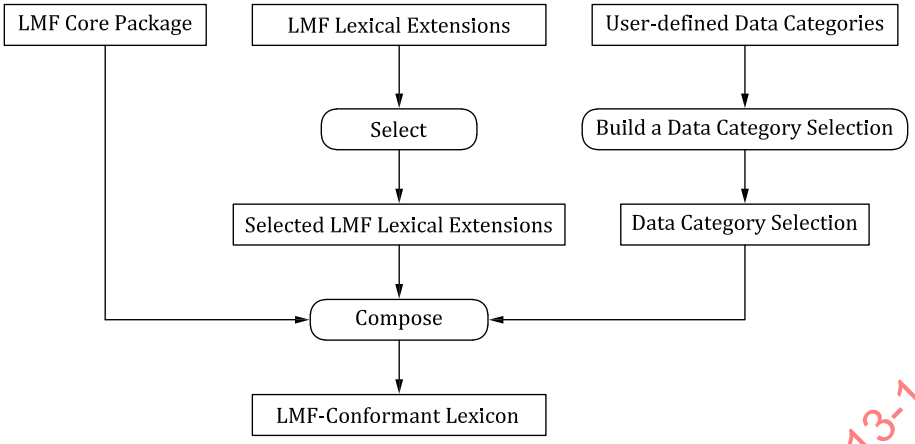

Figure 4 — LMF process

5.6.2 Lexicon comparison

When comparing two LMF-conformant lexicons to determine what information will be retained or lost during a data exchange, the following factors can be used:

— the catalogue of classes and subclasses;

— the associations among the classes;

— the links (cross-references) and link mechanisms;

— the catalogue of data categories and the DCSs;

— the inclusion of extensions.

A lossless exchange of information is possible when two lexicons have different structures (classes and associations). The lexicon developer shall determine the degree of possible loss through a comparison of the structures and the methods of object instantiation in order to identify equivalences and gaps. When two lexicons have different data category catalogues and selections, a lack of metadata in one lexicon can prevent the exchange of both content and metadata from the other lexicon. Two lexicons can be LMF-conformant and allow a lossless exchange of information without being identical.

Annex A
(informative)

Data category examples

This annex provides examples of data categories, including attributes, values and associated classes. The catalogue in Table A.1 is not intended to be exhaustive.

Table A.1 — Examples of data category selection

Attributes	Values	Allocated class	Comments
lexiconTitle		Lexicon	Open category.
lexiconID		Lexicon	
languageCoding	ISO 639	GlobalInformation LexiconInformation	A descriptive set of language coding values.
scriptCoding	ISO 15924	GlobalInformation LexiconInformation	
characterCoding	UNICODE type	GlobalInformation LexiconInformation	A descriptive set of UNICODE character coding values, such as UTF8. IRIs require UTF-8.
xml:lang	ISO 639	OrthographicRepresentation	Consult IETF BCP 47.[8]
script	ISO 15924	OrthographicRepresentation	
notation		GlobalInformation LexiconInformation OrthographicRepresentation	User defined, e.g. various US Board of Geographic Names (BGN) transliterations. GlobalInformation and LexiconInformation classes usually contain a list of notations and a description of their usage.
representationType	canonicalForm phoneticForm transliteration transcription romanization syllabification hyphenation	OrthographicRepresentation	
geographicalVariant			Open category.
formCategory	lemma wordForm relForm wordPart stem verbRoot		
formType	fullForm abbreviation acronym inflection	Form	