INTERNATIONAL STANDARD

ISO 24613-2

First edition
2020-07

Language resource management — Lexical markup framework (LMF) —

Part 2: Machine-readable dictionary (MRD) model

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

Partie 2: Modèle de dictionnaire lisible par ordinateur (MRD)

Reference number
ISO 24613-2:2020(E)

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 documents 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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 first edition of ISO 24613-2, together with ISO 24613-1:2019, ISO 24613-3[1], ISO 24613-4[1], ISO 24613-5[1], ISO 24613-6[2] and ISO 24613-7[2], cancels and replaces ISO 24613:2008, which has been divided into several parts and technically revised.

The main changes compared to the previous edition are as follows.

This edition merges two normative annexes from the previous edition, Annex A, Morphology extension, and Annex C, Machine-readable dictionary extension, providing a more cohesive description of the key structures (classes and associations) found in that edition. The cross-reference (CrossREF) model introduced in Part 1, Core model, of this edition, provides a new capability for correlating lexical features across different form and sense classes. In addition, the CrossREF model has replaced the ListOfComponents and Component classes, enabling a more extensible and flexible capability for managing multiword expressions. The metamodel of generalization by typing introduced in Part 1 provides a more rigorous and unambiguous framework for applying LMF modelling mechanisms in ways that enable greater editorial freedom and support the comparison of different LMF conformant designs. This edition has kept most of the informative examples found in the previous edition (deleting only a few redundant examples) and has added new examples to illustrate new modelling features. There have been some class name changes (e.g. OrthographicRepresentation for Representation and Translation for Equivalent), but no changes in the underlying concepts of the previously existing classes.

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.

[1] Under preparation.

[2] Planned.

Introduction

The ISO 24613 series is based upon the definition of an implementation-independent metamodel combining a core model and additional models that onomasiological (form-oriented) and semasiological (concept-oriented) lexical content can take.

It provides guidelines for various implementation use cases, and where appropriate describes LMF compliant serializations that fit various application contexts.

This document extends ISO 24613-1, the LMF core model, through the use of the processes and mechanisms described in ISO 24613-1. The objective is to enable flexible design methods to support the development of machine-readable dictionaries for different purposes while enabling cross-comparisons of different designs and a basis for developing assessments of standards conformance. The scope of supported design goals ranges from simple to complex human-oriented MRDs, both monolingual and bilingual, lexicons that support conceptual-lexical systems through links with ontological resources, rigorously constrained lexicons for supporting machine processes, and lexicons that provide an extensional description of the morphology of lexical entries. Since this document is based on ISO 24613-1, the LMF core model, it is designed to interchange data with other parts of the ISO 24613 series where applicable.

Language resource management — Lexical markup framework (LMF) —

Part 2: Machine-readable dictionary (MRD) model

IMPORTANT — The electronic file of this document contains colours which are considered to be useful for the correct understanding of the document. Users should therefore consider printing this document using a colour printer.

1 Scope

This document describes the machine-readable dictionary (MRD) model, a metamodel for representing data stored in a variety of electronic dictionary subtypes, ranging from direct support for human translators to support for machine processing.

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 24613-1, *Language resource management — Lexical markup framework (LMF) — Part 1: Core model*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 24613-1 apply.

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

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

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

4 Key standards used by LMF

The key standards applicable to this document are described in ISO 24613-1, the LMF core model.

5 The machine-readable dictionary (MRD) model

5.1 General

The MRD model is represented by UML classes, associations among the classes (the structure), sets of data categories (attribute-value pairs), and links (cross-references). Subclauses 5.2 through 5.12 describe each of these features, their interdependencies, and their implementation.

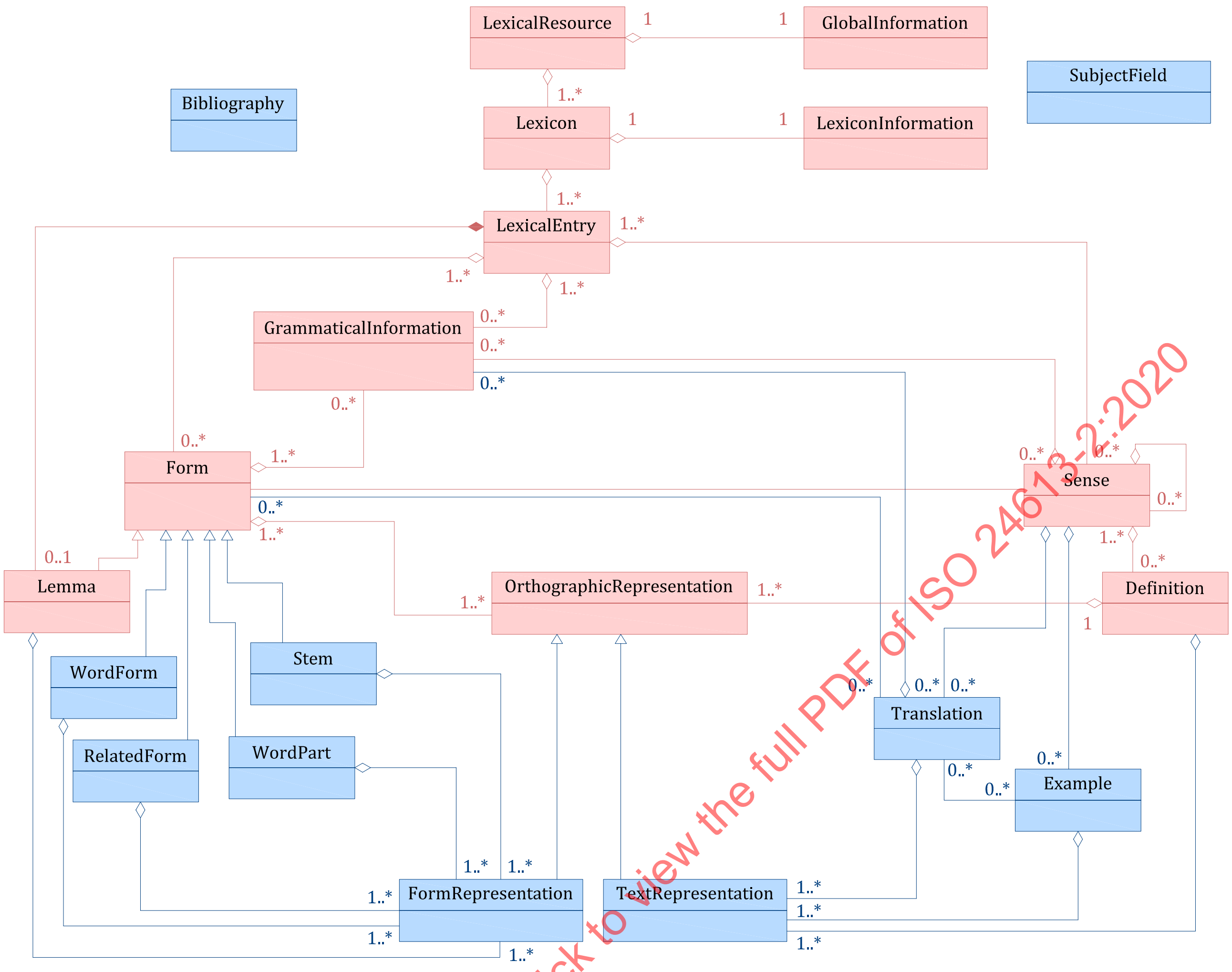

Figure 1 — MRD class model

5.2 MRD class model

5.2.1 Set of classes

The classes defined in ISO 24613-1, the LMF core model, that are used in the MRD extension include LexicalResource, GlobalInformation, Lexicon, LexiconInformation, GrammaticalInformation, LexicalEntry, Lemma, Form, Sense, Definition, OrthographicRepresentation, and principles for applying the CrossREF class. These classes, together with the associations and constraints described in ISO 24613-1, are applicable to the design of MRD. New classes introduced in this document include WordForm, Stem, WordPart, RelatedForm, Translation, Example, FormRepresentation, TextRepresentation, Bibliography and SubjectField.

5.2.2 Class selection and multiplicity

The sets of classes shown in the model in Figure 1 can support a wide range of design objectives. A specific design objective can require all or only some of the classes shown in the above model and can require as well the creation of new subclasses. The recommended first step in the creation of a model for a specific design objective (e.g. a bilingual dictionary) should be the selection and possible exclusion of classes contained in the class model and the application of desired multiplicities to the class associations as required by the model and the design goals (the optional classes in the model have a minimum cardinality of zero). The developer can create new subclasses, as needed, using the mechanisms described in ISO 24613-1, the LMF core model. The selected classes and their associations provide the structure and nodes (classes) appropriate for the intended lexical design. The classes and subclasses are described in detail below (see 5.5 to 5.11).

EXAMPLE

- Certain classes of MRD, such as monolingual and bilingual dictionaries, generally require a Sense class instantiation.
- Certain classes of MRD, such as concept hierarchies, do not necessarily require a Form class instantiation.
- Certain classes of MRD, such as orthographic dictionaries and extensional morphologies do not necessarily require a Sense class instantiation.
- Certain classes of MRD, such as extensional morphologies, can provide constraints on the attributes managed by the RelatedForm class.

NOTE The purpose of the MRD morphology extension is to provide the mechanisms to support the development of lexicons that have an extensional description of the morphology of lexical entries in which all relevant inflections or derivations of a lemma are included.

5.2.3 Generalization

Figure 1 illustrates the use of generalization (typing) through the Form class (superclass) and its subclasses, Lemma, WordForm, Stem, and WordPart, and OrthographicRepresentation (superclass) and its subclasses, FormRepresentation and TextRepresentation. The typing mechanism describes how to allocate specific sets of data categories, associations, multiplicities, and cross-references to subclasses (e.g. Lemma) in order to redefine the superclass. ISO 24613-1 provides a more complete description of typing.

NOTE The subclasses shown in Figure 1 are available for use in LMF compliant designs, but are not exhaustive, since LMF allows the creation of additional subclasses. The lexicon designer specifies what sets of features are available in form features.

5.2.4 Object realization

LMF provides examples of object models (see Annex B), but does not provide an in-depth description of the overall methodologies for developing the object models, since those processes are heavily dependent on the choice of model serialization (e.g. XML, JSON). Different serializations can require different design approaches and impose limitations on how the object can be modelled.

EXAMPLE XML provides a number of structural models for implementing XML schemas. Within the framework of these models, a lexicon designer could implement UML classes as XML elements or a combination of an XML element and attributes. For example, a designer could instantiate the Lemma class as a <Lemma> element or a <form type="lemma"> element-attribute combination. These object modelling choices use selective class and data category allocations to implement object designs that are strongly dependent on the structures and methods of the chosen serialization.

5.3 Data category selection and class population

Data category selection can include all or a subset of data categories used by a given domain. Examples of data categories and their allocations are listed in Annex A. Where needed, the lexicon developer can create new data categories that are not listed in the annex.

5.4 CrossREF allocation

Figure 1 shows links (cross-references) between the Form and Sense and the Form and Translation classes. The principles for modelling cross-references are described in ISO 24613-1, the LMF core model. The CrossREF class is specifically allowed for the LexicalEntry class, the Lemma class, the WordForm class, the WordPart class, the Sense class, and the Sense class children. The lexicon designer should consider using cross-references with the RelatedForm class. The use of data categories to provide information about the CrossREF features (e.g. internal reference, external reference, type of ID, lexical type, syntactic type, or semantic type) is a best practice.

EXAMPLE A WordPart that contains the suffix component of a Lemma can be cross-referenced with the LexicalEntry that contains that suffix as the Lemma, or a Sense can be cross-referenced with a broader Sense contained in a different LexicalEntry, or an authentic Quote can be cross-referenced with a document that contains the Quote.

NOTE The range of data categories describing CrossREF features is potentially quite broad and could be used to support references to audio, video, and other types of metadata relevant for lexical resources.

5.5 Form subclasses

5.5.1 WordForm class

WordForm is a Form subclass containing a word form, such as an inflected form, that a lexeme can take when used in a sentence or a phrase. The WordForm class is in a zero-to-many aggregate association with the LexicalEntry class (inheriting the Form multiplicity). The WordForm class can manage simple lexemes, compounds, multi-word expressions, and sub-lexemes such as affixes and roots.

5.5.2 Lemma class

Lemma is a Form subclass representing a lexeme or sub-lexeme used to designate the LexicalEntry (part of the Form-Sense paradigm). The Lemma class is in a zero-to-one aggregate association with the LexicalEntry class that overrides the multiplicity inherited from the Form class (see ISO 24613-1 for a more complete description of the Lemma).

5.5.3 Stem class

Stem is a Form subclass containing a stem or root. The Stem class can be typed as a specific type of stem or root (e.g. type=”arabicRoot”). The Stem class is in a zero-to-one aggregate association with the LexicalEntry class (overriding the multiplicity inherited from the Form class).

5.5.4 WordPart class

WordPart is a Form subclass representing sub-lexeme parts other than the stem or root (e.g. affix, prefix, suffix). The WordPart class is in a zero-to-many aggregate association with the LexicalEntry class.

5.5.5 RelatedForm class

RelatedForm is a Form subclass containing a word form or a morph that is typical of run-on entries in print dictionaries. The RelatedForm has a different Sense than the Lemma and can be considered a candidate for eventual inclusion in a different LexicalEntry object when realized in a lexical database. The RelatedForm can be related to the Lemma in a variety of ways (e.g. synonym, cross-reference, multi-word expression, idiom). The RelatedForm class is in a zero-to-many aggregate association with the LexicalEntry class and can contain a recursive cross-reference to the LexicalEntry class, which would be realized as a link to a different LexicalEntry object when instantiated in a lexical database. The RelatedForm class can be typed (generalization) using data categories.

EXAMPLE A developer possibly wants to use the RelatedForm class for a multi-word expression (e.g. *United States*) that contains a component form of a Lemma (e.g. *united*). The design goal could be to preserve the format of the original source material, or to provide immediate user support while developing an improved lexicon that includes /united/ and /United States/ as separate entries.

5.6 FormRepresentation class

FormRepresentation is an OrthographicRepresentation subclass that contains the text literals and metadata (e.g. pronunciation, hyphenation, xml:lang, script) for a Lemma, WordForm, or other subclass of the Form class. The FormRepresentation class is in a one-to-many aggregate association with a Form subclass. The FormRepresentation class allows subclasses (typing).

NOTE Data categories, such as xml:lang, script, and notation, are associated with the OrthographicRepresentation class and inherited by subclasses.

EXAMPLE Because searching for WordPart data (e.g. suffix components of a form) is generally not a high user priority, a lexicon developer might want to create a PartRep subclass of the FormRepresentation class in order to support application designs that use object (class) names as part of their query strategy. Creating different search criteria for FormRepresentation objects and PartRep objects is one way to increase search and display efficiency.

5.7 TextRepresentation class

TextRepresentation is an OrthographicRepresentation subclass that manages the text literals and metadata (e.g. xml:lang, script) for classes associated with the Sense class and its child classes. The TextRepresentation class enables improved modelling of the Form-Sense paradigm by distinguishing the management of descriptive text literals in the Sense related classes from text literals that represent a form (e.g. word form, sub-lexeme). The TextRepresentation class is in a one-to-many aggregate association with the Definition, Translation, or Example class. The TextRepresentation class can be typed (allows subclasses).

NOTE In practice, a TextRepresentation class associated with a Definition class will most likely be in a one-to-one association.

EXAMPLE A Quote subclass could be created for a TextRepresentation class associated with an Example class that provides authentic context.

5.8 Translation class

In a bilingual MRD, the Translation class represents the translation equivalent of the word form contained by the Lemma or WordForm class. The Translation class is in a zero-to-many aggregate association with the Sense class, which allows the lexicon developer to omit the Translation class from a monolingual dictionary.

5.9 Example class

The Example class contains a text string that illustrates the usage of the Lemma, WordForm, or Translation in authentic or constructed context. The Example class can be typed (subclass) in order to further define the context. The Example class is in a zero-to-many aggregate association with the Sense class. The Example class can also be in a zero-to-many aggregate association with the Translation class when there is a need to differentiate among multiple Example objects contained in a Sense, not all of which have to be directly associated with the Translation.

5.10 SubjectField class

SubjectField is a class managing subject domain information for a LexicalEntry or an object (class instantiation) of the LexicalEntry. When there are multiple class instantiations, for example Sense class instantiations, each instantiation can have a different SubjectField.

5.11 Bibliography class

Bibliography is a class containing source information for a LexicalEntry or an object (class instantiation) of the LexicalEntry. The Bibliography class can be associated with more than one class in a LexicalEntry, depending on design goals.

NOTE The Bibliography class can more typically be associated with a Form subclass, the Sense class, or one of the child classes of the Sense class.

5.12 Multiword Expression (MWE) Analysis

LMF enables an analysis of MWE using cross-references between an instantiation of a Lemma subclass containing the MWE and two or more different entries, each of which contains a component form of the MWE. The CrossREF class is the main mechanism for enabling MWE analysis. Possible targets can include the Lemma objects or the Sense objects in the other entries. A design that targets the Sense objects would be the optimum design choice for reducing semantic ambiguity. The LexicalEntry objects can be in the same lexicon, a different lexicon, or even a different resource. The PartOfSpeech and other grammatical features describing the component word forms can be different from the grammatical features describing the MWE in the source Lemma.

NOTE In the most common use cases for MWE analysis, the MWE is contained in the Lemma, but it is possible that other Form subclasses are allowed.

Annex A
(informative)

Data category examples

A.1 Introduction

This annex provides examples of data categories, including attributes, values, and class allocations. Some data categories introduced in ISO 24613-1, the LMF core model, are presented here when there are new allocation examples related to subclasses. The data categories in this document and in ISO 24613-1 should be referenced together.

A.2 Data category examples

Table A.1 gives open data categories.

Table A.1 — Open data categories

Attribute	Allocation	Comments
lexiconTitle	/Lexicon/	
lexiconID	/Lexicon/	
notation	/LexiconInformation/ /FormRepresentation/ /TextRepresentation/	For example, notation describes a unique orthographic system, such as a language transliteration approved by the U.S. Board of Geographic Names (BGN).
note		A comment that provides information about the values, associations, usage, or other aspects of LMF classes or class content.

A.3 Examples of closed data category types and picklists

The closed data categories in Table A.2 are some examples used in existing or planned designs.

Table A.2 — Data category types and picklists

Attribute	Values	Allocation	Comments
partType	affix prefix suffix circumfix enclitic	/Lemma/ /WordPart/	When associated with a /Lemma/, partType describes a single instance of a sub-lexeme form that has been selected as the Lemma; when associated with a /WordPart/, partType specifies the type of a sub-lexeme form that is associated with the Lemma and managed by the /WordPart/.

Table A.2 *(continued)*

Attribute	Values	Allocation	Comments
relationshipType	synonym antonym crossRef variant	/RelatedForm/ /CrossREF/	
MWEType	MWEcompound auxilliaryVerb-Noun	/Lemma/ /WordForm/	
actionType	inchoative dynamic	/GrammaticalInformation/	Used to describe features in languages that distinguish dynamic and inchoative or stative verb categories.
verbPerfective	regular short	/GrammaticalInformation/	Used to describe a class of perfective verbs that has specific inflection or agglutinative features.
stativePerfective	CaCa regular	/GrammaticalInformation/	Used to describe a class of perfective verbs that apply to stative or inchoative modes.
rootType	basic fund	/GrammaticalInformation/	Used to describe a class of roots, such as roots in agglutinative languages.
representationType	canonicalForm phoneticForm transliteration transcription romanization syllabification hyphenation	/Lemma/ /WordForm/	
xml:lang	ISO 639 codes	/FormRepresentation/ /Orth/ /Pron/ /Quote/	Consult IETF BCP 47.
script	ISO 15924 codes	/FormRepresentation/ /Orth/ /Pron/ /Quote/	
notation	open	/FormRepresentation/ /Orth/ /Pron/ /Quote/	

Annex B
(informative)

Machine-readable dictionary examples

B.1 Introduction

This annex provides examples of MRD object models that illustrate approaches for dealing with a number of important design objectives, including the treatment of regional variants, multilingual scripts and orthographies, Arabic root management, MWE analysis, design for agglutinative languages, and language-variant object association patterns in multilingual lexicons. Examples in English, Arabic, Chinese, French, German, and Zulu provide coverage of different language families, scripts, and morphological typologies. There are also several examples that illustrate approaches for model simplification. See ISO 24613-1:2019, 5.5 (Methods for data category selection and subclass creation), and in particular, 5.5.6 (Principles for model simplification), for a normative description of the simplification process.

B.2 Example of simple FormRepresentation

In the following example, the developer is designing an English language monolingual lexicon. The lexical entry is associated with a lemma *clergyman* and two inflected forms *clergyman* and *clergymen*. The developer references the MRD metamodel to support an LMF conformant design that best meets the developer's objectives. Figure B.1 shows a section of a design that instantiates the conceptual features of the MRD metamodel at a very detailed level for the grammatical information and orthographic representations.

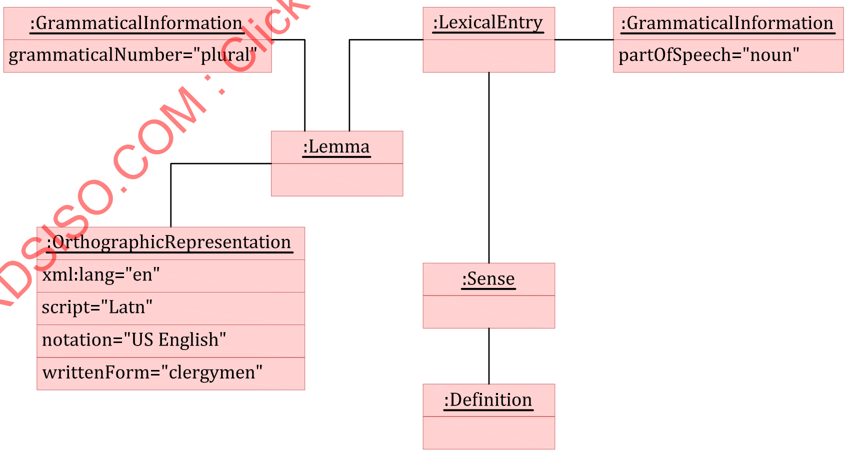

Figure B.1 — Simplified design, first step

While conformant to the MRD metamodel, the instantiation in Figure B.1 is overly complex if used for a monolingual lexicon or a collection of monolingual lexicons. The manner in which design choices are implemented depends in large part on what serialization is used. In an XML serialization, for example, choices include whether to use elements or attributes for expressing a metamodel object, implementing a metamodel object as a container, or entailing a concept directly within a data category when that can be achieved without impacting understanding of the conceptual design. When data category values are consistent across metadata objects, the GlobalInformation or LexiconInformation classes can often be used to manage that information. Figure B.2 shows a simplified approach for implementing MRD designs for monolingual dictionaries. Data categories, such as xml:lang, which have common object allocations across all entries can be managed in the GlobalInformation or LexiconInformation class, reducing the complexity of the Lemma, WordForm, and/or OrthographicRepresentation instantiations.

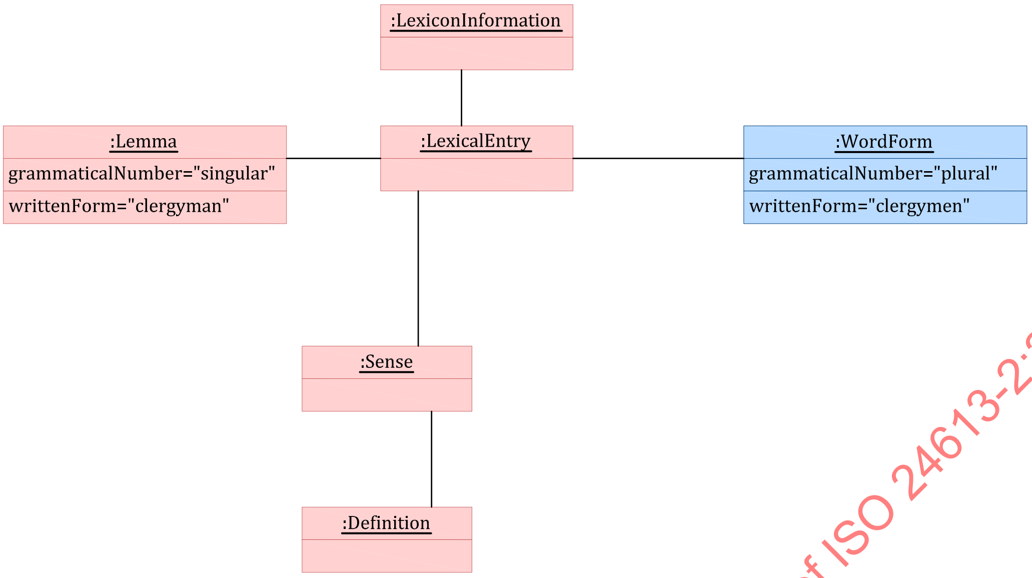

Figure B.2 — Simplified design, second step

B.3 Example of regional variants

Regional variants can be modelled using the FormRepresentation class, with variant /phoneticForm/ attributes, as shown in Figure B.3.

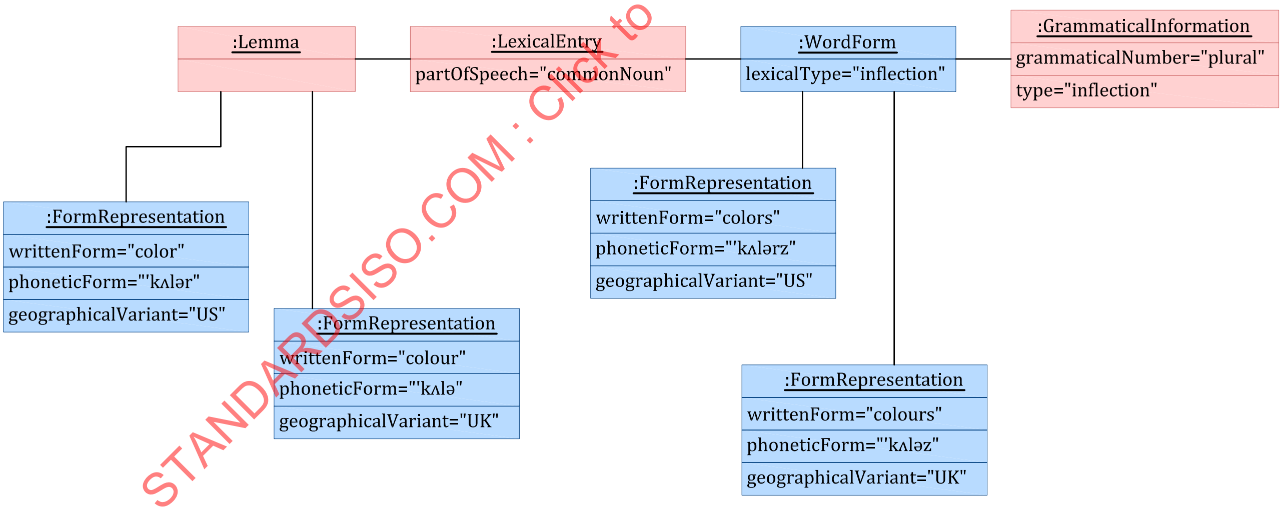

Figure B.3 — Example of regional variants using FormRepresentation

B.4 Example of different styles in realization

Figure B.4 shows how LMF modelling principles can be used to instantiate two different object models that are equivalent, but not identical. In the XML realizations, the Lemma element and the Form type=”lemma” element-attribute combination instantiate the same sets of features and thus represent the same concepts. The element (object) names are not necessarily the same as the class names, since the instantiation methods follow XML design practices. A different realization (e.g. JSON) might use different instantiation methods and strategies. The two examples in Figure B.4 use different approaches when instantiating the FormRepresentation class. The example on the left reuses the class name and explicit features (data categories) to specify the object characteristics; the example on the right uses a descriptive type, not the class name. In this latter case, the full characteristics of the object can be described in GlobalInformation or LexiconInformation, as appropriate.

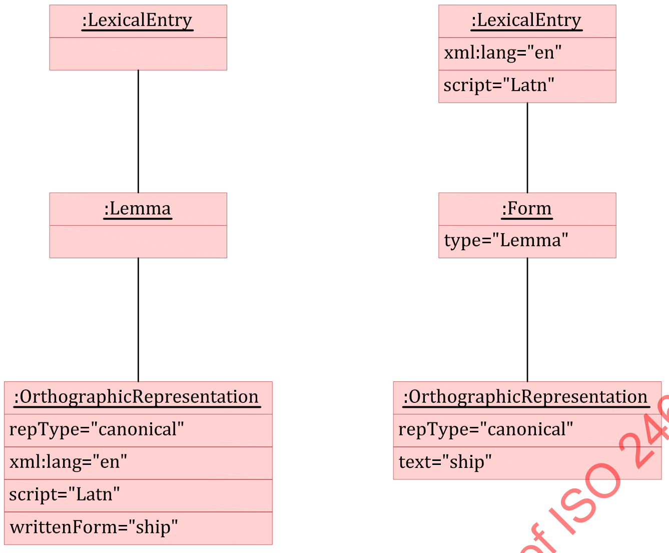

Figure B.4 — Object instantiation - Lemma

B.5 Example of multiple scripts and orthographies

In the example in Figure B.5, the Lemma and WordForm are both represented as inflected forms with data categories describing their grammatical features. The Lemma and WordForm have three FormRepresentation class instances that describe variant orthographical representations of the form.

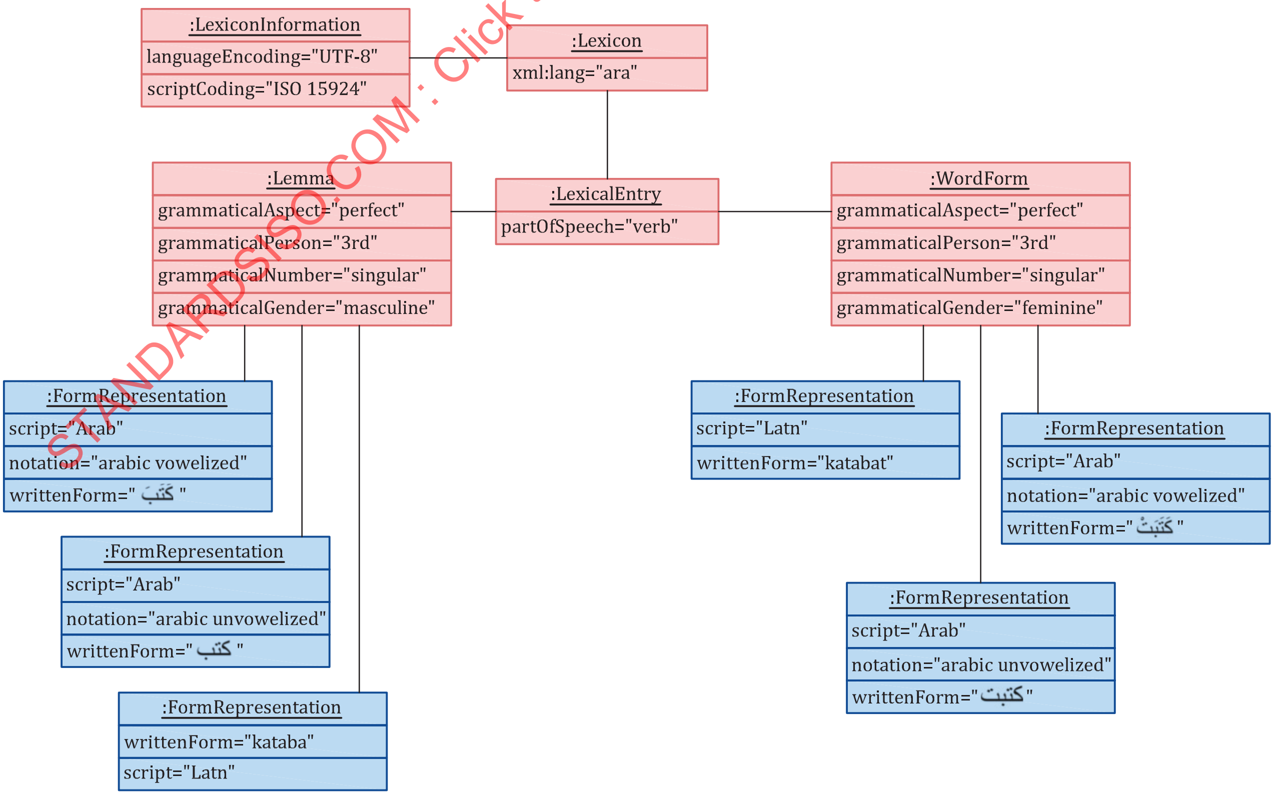

Figure B.5 — Example of multiple scripts and orthographies

It is worth noting that this strategy is not the only possible option in Arabic. Another strategy would be to describe the Arabic vowelized script forms in the lexicon and to provide an external mechanism to compute automatically the Arabic unvowelized script forms and transliterations. In this case, FormRepresentation instances are not needed.

B.6 Example of a bilingual MRD with multiple representations

The example of a bilingual MRD in Figure B.6 shows an entry containing the Arabic word “kitaab” and two equivalents in English, “book” (the most common meaning) and “credentials”. The transcriptions provide users with more information about the pronunciation of the words and their context than can be derived from the Arabic script. In this example, the WordForm class provides information about the form and pronunciation of the Arabic broken plural, which is an irregular inflection. The decision to include the FormRepresentation class is an editorial choice determined by the goals of the lexicon developer. If the goal were to produce an Arabic-English MRD that contained only Arabic script for the Arabic word forms, the inclusion of the FormRepresentation class would not be necessary.

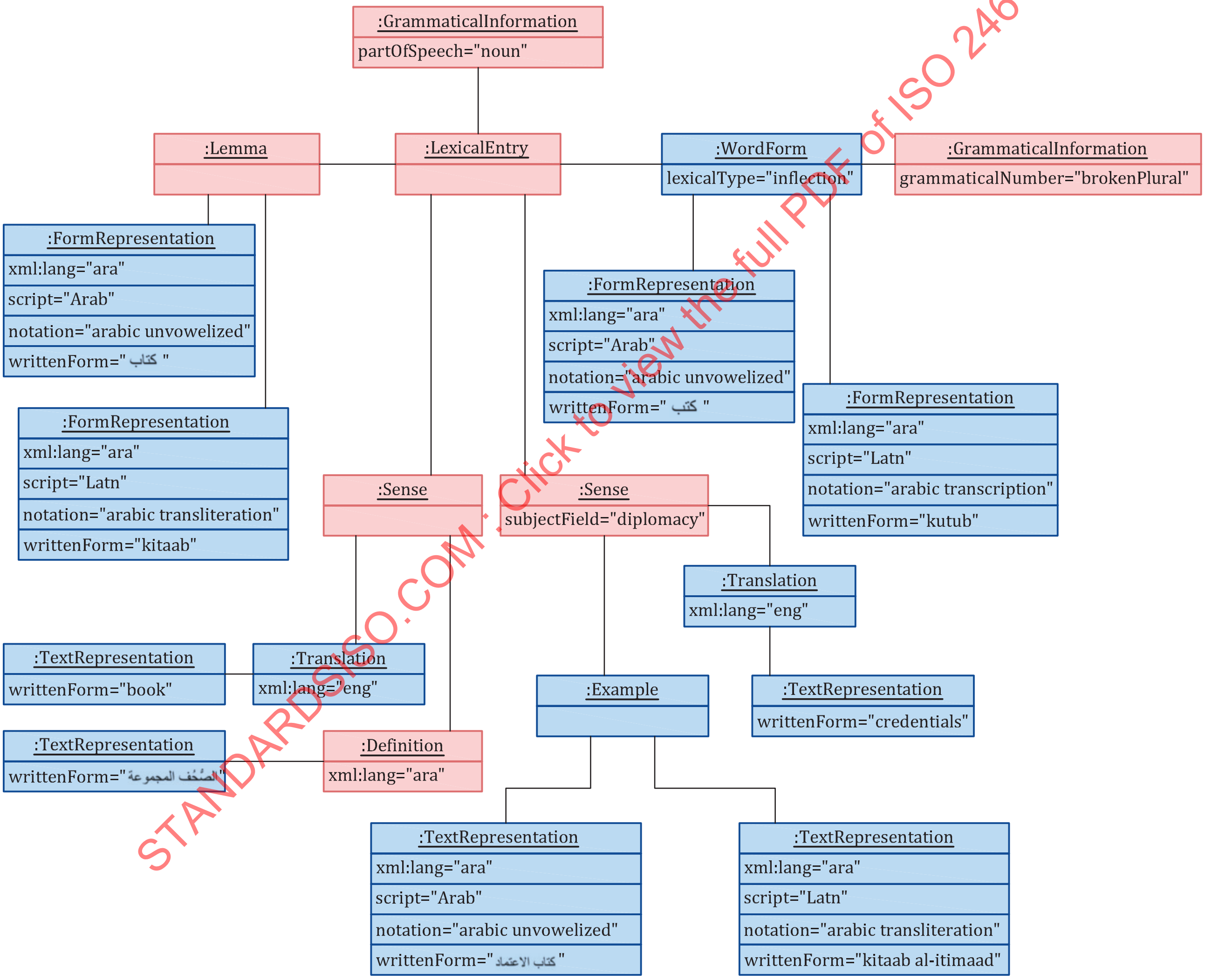

Figure B.6 — Instantiation example for a bilingual MRD

B.7 Examples of multiple scripts and orthographies in Chinese

Chinese is an isolating language. Over the centuries, the Chinese writing system has changed. The latest movement to simplify the Chinese writing system originated in the 1890s and was extended in the 1950s. The strategy of simplification involves a reduction in the number of strokes of commonly used characters. And at the moment, two variants are in use. According to ISO 15924, the script code is *Hans* for the simplified variant and *Hant* for the traditional variant.

The following example shows a situation where a relatively simple traditional character was borrowed to replace a more complex traditional character. In this case, the borrowed character is still used for its original sense. The language and script information are global to all lexical entries, thus these orthographical attributes are located on the Lexicon or LexiconInformation instances, see Figure B.7.

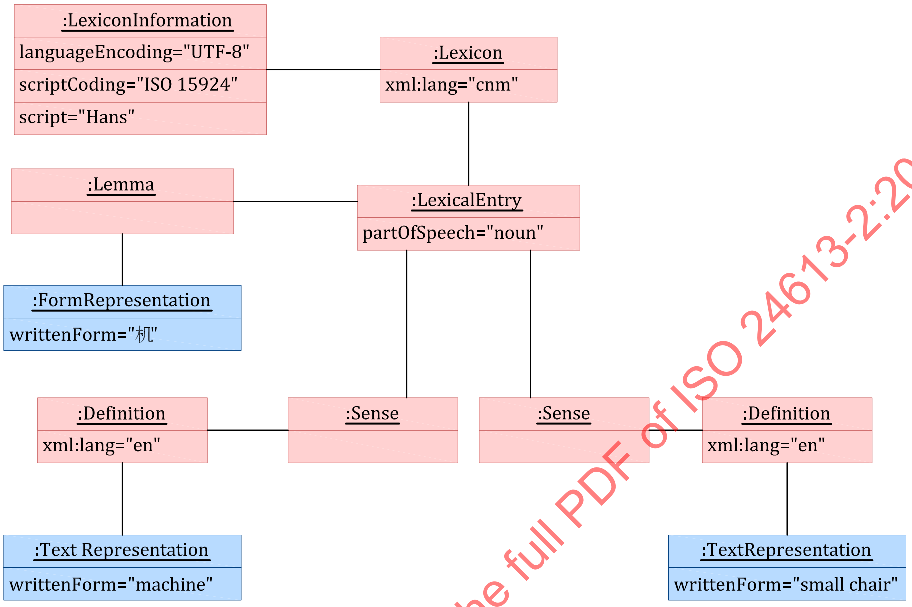

Figure B.7 — Example in simplified Chinese writing system

If the user wants to describe traditional forms, two LexicalEntry instances are required because there are two distinct traditional forms and the meaning of each of these lexemes is different, see Figure B.8.

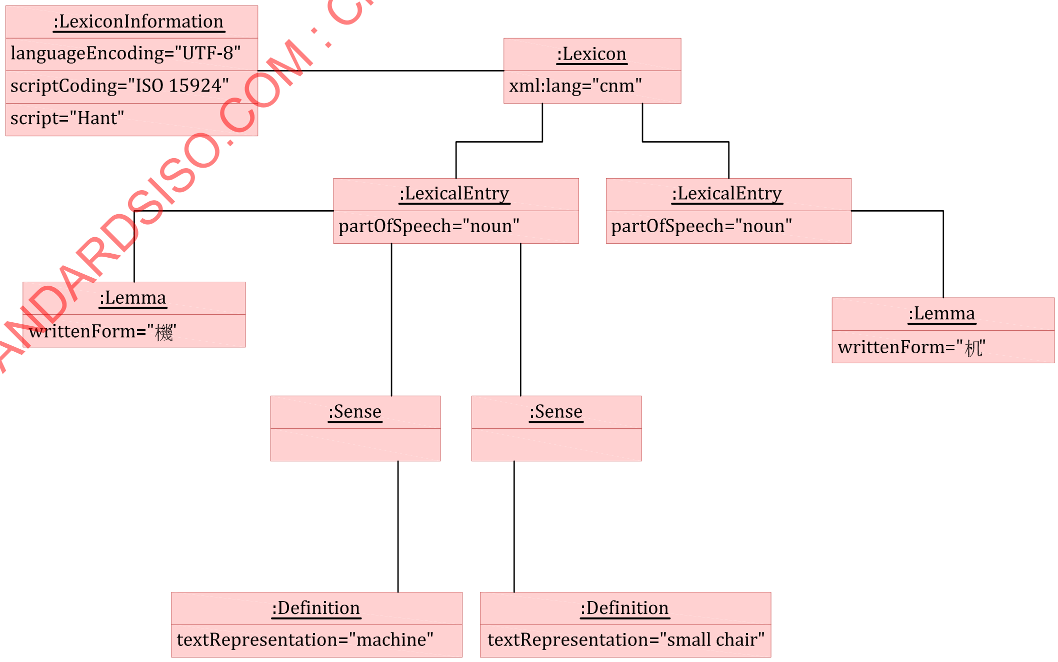

Figure B.8 — Example in traditional Chinese writing system

It is worth noting that if the user wants to mix simplified and traditional forms in the same lexicon, the script attribute cannot be set to the Lexicon instance but shall be set to each FormRepresentation instance, since the orthographic values are not globally applicable. See Figure B.9.

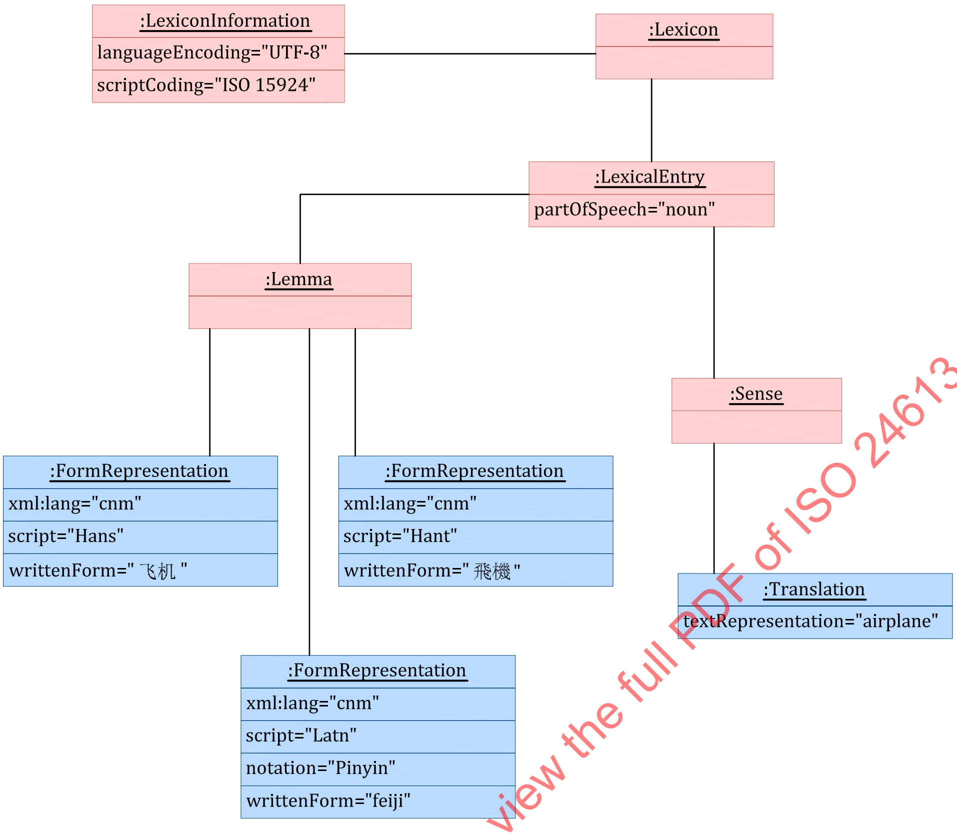

Figure B.9 — Example of mixed simplified and traditional Chinese

B.8 Example of Arabic root management

In the example in Figure B.10, the Arabic root is represented by a Stem class typed as an /arabicRoot/. The verbs *yadrus* and *yudarris* both share the same root *drs*. When this schema is instantiated in a dictionary application, a user can search separately for verbs, or search on the root, *drs,* and get back all derived forms sharing the root, including verbs, nouns, and other parts of speech.

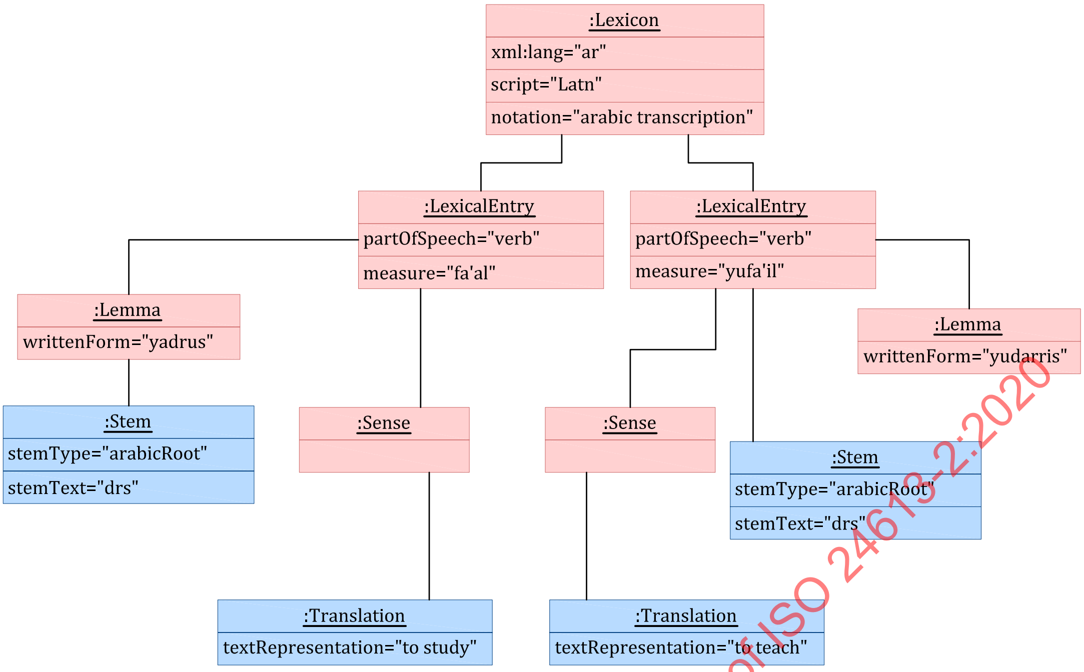

Figure B.10 — Example of Arabic root management

B.9 Examples of Zulu lexical entries (management of agglutinative features)

Zulu is a member of the Bantu language family spoken mainly in South Africa. Zulu nouns are divided into 16 morphological classes (or genders), with different noun prefixes for singular and plural forms. Other parts of speech generally have concord prefixes which shall be in agreement with the noun classes. Like other Bantu languages, Zulu has an agglutinative morphology with very complex verb forms. This section provides examples of different approaches for managing noun entries and verb entries. Both the following examples (see B.9.1 and B.9.2) use complex orthographic representation models on the assumption that the Zulu-English dictionary is part of a collection of bilingual lexicons in multiple source languages.

B.9.1 Example of a Zulu noun entry

In Zulu, a noun stem can be used in one or more morphological classes. For example, *umuntu*, *person*, and *ubuntu*, *human kindness*, share the noun stem -*ntu*. Some Zulu-English dictionaries are organized by noun stem, but this approach is problematical for language learners, since phonetic shifts can make it difficult to determine the root from the word in context. In the below example, the Lemma and Word Form contain the full word, *umuntu* and *abantu*, respectively. The noun root, -*ntu* is managed by the Stem class. In an electronic dictionary, it is possible for a user to search by the root, -*ntu* and retrieve all examples that have that root. The GrammaticalInformation class associated with the Lemma and Word Form classes manages information about the morphological class and the grammatical number. A GrammaticalInformation class instance associated with the LexicalEntry class contains a /grammaticalClassGroup/ which aggregates the morphological classes of the entry in order to support the extraction of metadata for building specialized lexicons (non-LMF) for morphological analyzers. This example also shows how the Definition class can be used in a bilingual dictionary to explain Zulu words that do not have an exact equivalent in English. See Figure B.11.

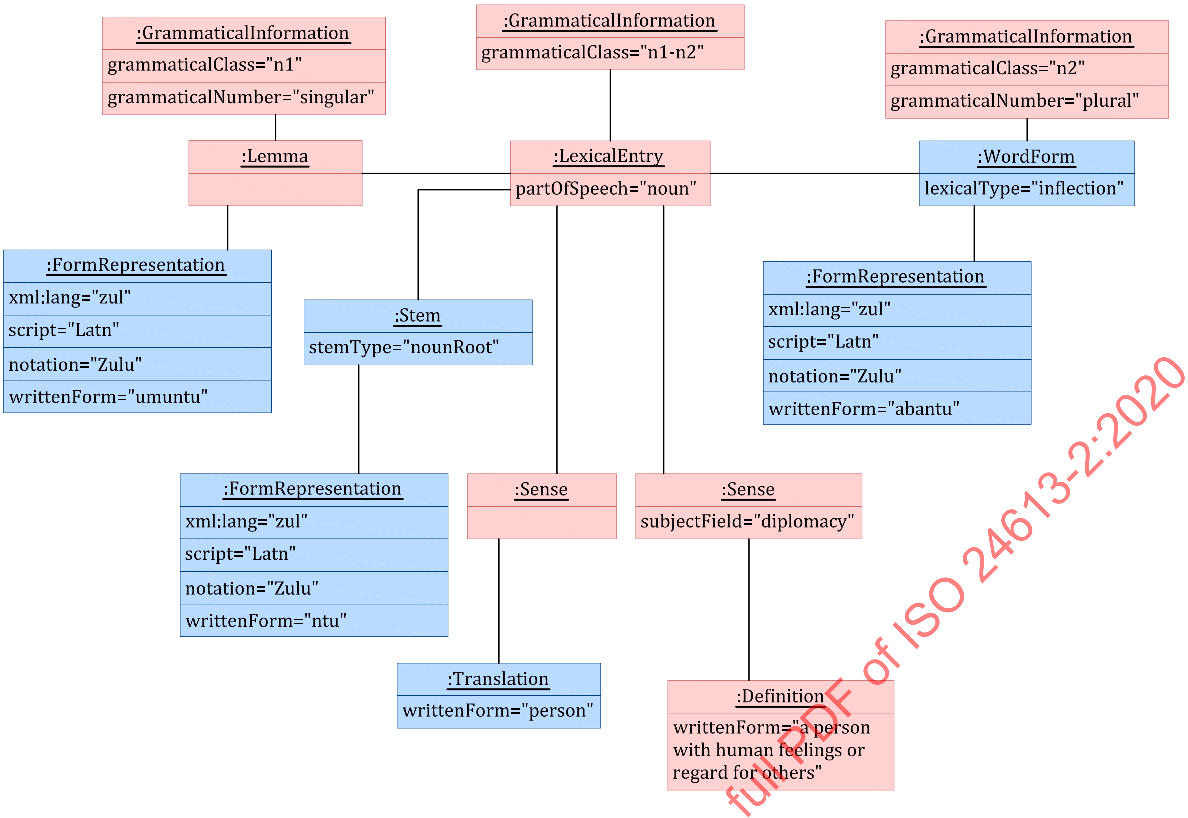

Figure B.11 — Example of Zulu noun entry

B.9.2 Example of a Zulu verb entry

In contrast to the noun entry (see B.9.1), the verb entry here is organized by the verb root, since the Zulu verb consists of a number of different morpheme types that are built on the verb root. The verb root is managed by the Lemma, which is typed as a “verbRoot”. The GrammaticalInformation class manages the data categories relevant for determining the conjugation of the verb managed by the entry. The verb in the example, *lala*, is an inchoative verb with the meanings *start to rest*, *start to lie down*, *go to sleep* in the present and regular perfective tenses. There is another perfective form in which the verb has the meaning *to rest*, *to lie down*, *to sleep*. In this example, these forms and meanings are shown through the Example class as an editorial choice. See Figure B.12.

NOTE Gloss represents a type of note rather than a class.