INTERNATIONAL STANDARD

ISO 16642

Second edition
2017-11

Computer applications in terminology — Terminological markup framework

Applications informatiques en terminologie — Plate-forme pour le balisage de terminologies informatisées

Reference number
ISO 16642:2017(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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 on 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 the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 37, *Terminology and other language and content resources*, Subcommittee SC 3, *Computer applications for terminology*.

This second edition cancels and replaces the first edition (ISO 16642:2003), which has been technically revised.

The main changes compared to the previous version are as follows:

— The following formats are no longer actively used. Consequently, references to these formats have been removed (including Annex A, Annex B, and Annex C):

  — Martif with specified constraints (MSC);

  — Geneter;

  — Data category interchange format (DCIF);

  — Generic mapping tool (GMT).

— With the removal of Annex B and Annex C, this document no longer includes any comprehensive code examples of a TML. Examples of TMLs are now available in ISO 30042, TermBase eXchange, and also at the following Web site: www.tbxinfo.net.

— References to the former ISO/TC 37 Data Category Registry or ISOcat have been changed from normative to informative. In addition, the name has changed to DatCatInfo, now as an example of data category repositories.

— References to ISO 12620:1999 and ISO 12620:2009 have been removed. These previous standards have been withdrawn.

— The TypedValuedElement style has been added.

— Examples have been updated to reflect ISO 30042:2008 (TBX). TBX-Basic is mentioned as a TML.

— Some of the examples and tables have been moved to appropriate sections.

— As a consequence of the aforementioned changes, some historical, didactic, or duplicate information has been removed to adhere more closely to ISO editorial standards.

Introduction

Terminological data are collected, managed and stored in a wide variety of systems, typically various kinds of database management systems, ranging from personal computer applications for individual users to large terminological database systems operated by major companies and governmental agencies. Terminology databases are comprised of various types of information, called data categories, and can adopt different structural models. However, terminological data often need to be shared and reused in a number of applications, and this sharing is facilitated when the data adheres to a common model. To facilitate co-operation and to prevent duplicate work, it is important to develop standards and guidelines for creating and using terminological data collections (TDCs) as well as for sharing and exchanging data.

This document presents a modular approach for analysing existing TDCs and designing new ones. It also provides a framework for defining terminological markup languages (TMLs) that are interoperable.

This document makes reference to DatCatInfo, an example of an available data category repository. DatCatInfo is an online database of information about the types of data that can be included in terminological data collections and other language resources. It is available at www.datcatinfo.net.

Computer applications in terminology — Terminological markup framework

1 Scope

This document specifies a framework for representing data recorded in terminological data collections (TDCs). This framework includes a metamodel and methods for describing specific terminological markup languages (TMLs) expressed in XML. The mechanisms for implementing constraints in a TML are defined, but not the specific constraints for individual TMLs.

This document is designed to support the development and use of computer applications for terminological data and the exchange of such data between different applications. This document also defines the conditions that allow the data expressed in one TML to be mapped onto another TML.

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 704, *Terminology work — Principles and methods*

ISO 1087-1, *Terminology work — Vocabulary — Part 1: Theory and application*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 26162, *Systems to manage terminology, knowledge and content — Design, implementation and maintenance of terminology management systems*

ISO 30042:2008, *Systems to manage terminology, knowledge and content — TermBase eXchange (TBX)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 1087-1 and the following apply.

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

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

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

3.1
basic information unit
information unit (3.12) attached to a *component* (3.3) of the metamodel and that can be expressed by means of a single *data category* (3.6)

3.2
complementary information
CI
information supplementary to that described in *terminological entries* (3.22) and shared across the *terminological data collection* (3.21)

Note 1 to entry: Domain hierarchies, institution descriptions, bibliographic references and references to text corpora are typical examples of complementary information.

3.3
component
elementary description unit of a metamodel to which *data categories* (3.6) can be associated to form a data model

3.4
compound information unit
information unit (3.12) attached to a *component* (3.3) of the metamodel that is expressed by means of several grouped *data categories* (3.6), that, taken together, express a coherent unit of information

3.5
conceptual domain
set of valid value meanings associated with a *data category* (3.6)

Note 1 to entry: For example, the data category /part of speech/ could have the following conceptual domain: /noun/, /verb/, /adjective/, /adverb/, and so forth.

3.6
data category
elementary descriptor used in a linguistic description or annotation scheme

Note 1 to entry: In this document, data categories are indicated in between forward slashes (/), e.g. /definition/.

3.7
data category repository
DCR
electronic repository of *data category specifications* (3.9) to be used as a reference for the definition of linguistic annotation schemes or any other representation model for language resources

Note 1 to entry: A DCR for language resources is available at http://www.datcatinfo.net.

3.8
data category selection
DCS
set of *data categories* (3.6) selected from a *DCR* (3.7)

3.9
data category specification
set of attributes used to fully describe a given *data category* (3.6)

Note 1 to entry: The abbreviation "DCS" is associated with data category selection and is not used for data category specification.

3.10
expansion tree
structured group of XML elements that implement a level of the metamodel in a given *TML* (3.23)

3.11
global information
GI
technical and administrative information applying to the entire *terminological data collection* (3.21)

Note 1 to entry: For example, the title of the terminological data collection, revision history, owner or copyright information.

3.12
information unit
IU
elementary piece of information attached to a structural level of the metamodel

3.13
language section
LS
part of a *terminological entry* (3.22) containing information related to one language

Note 1 to entry: One terminological entry may contain information on one or more languages.

3.14
object language
language being described

3.15
persistent identifier
PID
unique Uniform Resource Identifier (URI) that assures permanent access for a digital object by providing access to it independently of its physical location or current ownership

3.16
structural node
instance of *component* (3.3) within the representation of a *terminological data collection* (3.21)

3.17
structural skeleton
abstract description of an instance of a *terminological data collection* (3.21) in conformity with the metamodel

3.18
style
specification for the implementation of a *data category* (3.6) in XML

3.19
term component section
TCS
part of a *term section* (3.20) giving linguistic information about the components of a term

3.20
term section
TS
part of a *language section* (3.13) giving information about a term

3.21
terminological data collection
TDC
resource consisting of *terminological entries* (3.22) with associated meta data and documentary information

3.22
terminological entry
TE
part of a *terminological data collection* (3.21) which contains the terminological data related to one concept

Note 1 to entry: Every element in the TE can be linked to complementary information, to other terminological entries and to other elements in the same terminological entry.

3.23
terminological markup language
TML
XML format for representing a *terminological data collection* (3.21) conforming to the constraints expressed in this document

3.24
Unified Modeling Language
UML
language for specifying, visualizing, constructing and documenting the artifacts of software systems

3.25
vocabulary
<data modeling> set of strings used to implement a *data category* (3.6) according to a *style* (3.18)

3.26
working language
language used to describe objects

3.27
XML outline
part of a *terminological data collection* (3.21) corresponding to the XML implementation of the metamodel

4 Modular approach

Terminological Markup Framework (TMF) consists of two levels of abstraction. The first (and most abstract) level is the metamodel level. The metamodel level supports analysis, design and exchange at a very general level, i.e. it is independent of any specific implementation or software. The metamodel shall be shared by all TDCs that are compliant with TMF. The second level is the data model level, which adds the necessary data categories for representing specific TDCs.

The implementation of a data model in XML is called a terminological markup language (TML). TMLs can be described on the basis of a limited number of characteristics, namely

— how the TML expresses the structural organization of the metamodel (i.e. the expansion trees of the TML);

— the specific data categories used by the TML and how they relate to the metamodel;

— the way in which these data categories can be expressed in XML and anchored on the expansion trees of the TML, i.e. the XML style of any given data category;

— the vocabularies used by the TML to express those various informational objects as XML elements and attributes according to the corresponding XML styles.

Figure 1 represents the information required to fully specify a TML.

— The metamodel describes the basic hierarchy of components to which any TML shall conform.

— A set of data category specifications from a data category repository, which can form the basis for defining a data category selection (DCS) for the TML

— The dialectal specification (dialect) includes the various elements needed to represent a given TML in an XML format. These elements comprise expansion trees and data category instantiation styles, together with their corresponding vocabularies.

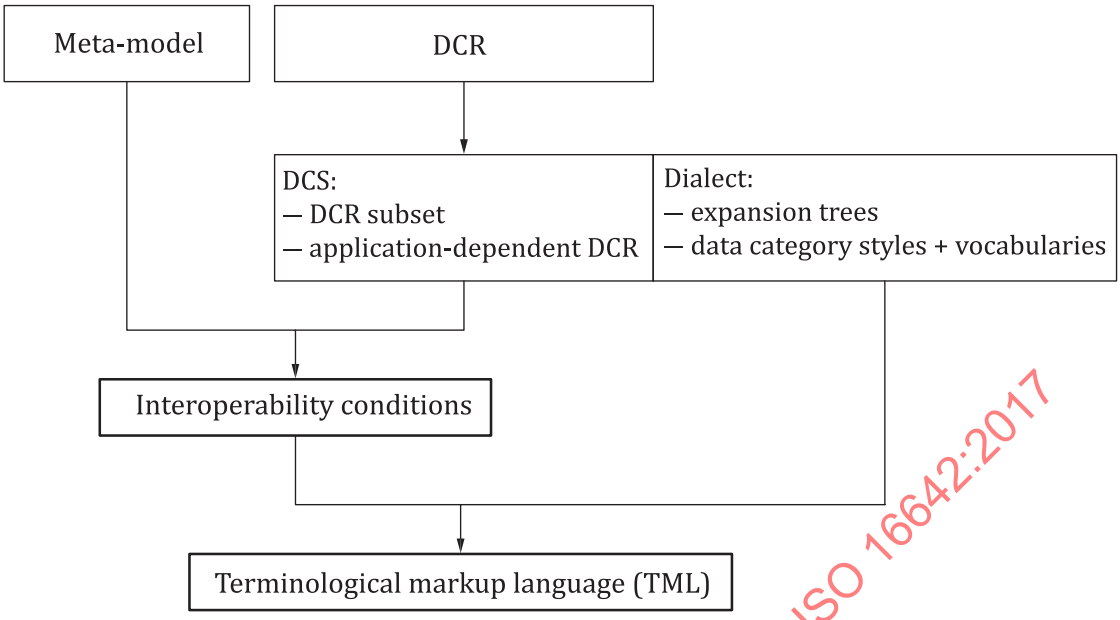

Figure 1 — Various knowledge sources involved in the description of a TML

A DCR providing sample data category specifications for language resources is available at www.datcatinfo.net. Where possible, data categories documented in this DCR should be used for a TML. If no suitable data category is available in this DCR, the implementers of the TML should propose the creation of the required data category specification within this DCR.

5 Generic model for describing terminological data

5.1 Principles

This clause describes a class of XML document structures which can be used to represent a wide range of terminological data formats, and provides a framework for representing these document structures in XML.

Each type of document structure is described by means of a three-tiered information structure that describes:

— a *metamodel*, which comprises a hierarchy of components;

— *information units*, which can be associated with each component of the metamodel;

— *annotations*, which can be used to qualify properties associated with a given information unit.

Information units can be basic or compound. A basic information unit encapsulates information that can be expressed by means of a single data category. A compound information unit encapsulates information that is expressed by means of several grouped data categories that, taken together, express a coherent unit of information. For instance, a compound information unit can be used to represent the fact that a transaction can be a combination of a transaction type (such as modification), the person who performed it, and the date when it was performed.

Basic information units, whether they are directly attached to a component or are placed within a compound information unit, can take two non-exclusive types of value:

— an atomic value corresponding either to a simple type (in the sense of XML schemas) such as a number, string, element of a picklist, etc., or to a mixed content type in the case of annotated text;

— a reference to a component in order to express a relation between it and the current component.

Information units can be abstractly represented as feature-value structures. For instance, the following markup sample

```
<Owner>UHB</Owner>
```

can be modelled as a basic information unit in the following feature-value structure:

```
[owner = UHB]
```

Similarly, the following TBX markup sample

```
<transacGrp>
    <transac>modification</transac>
    <transacNote type="responsiblePerson">YYY</transacNote>
    <date>1964-04-04</date>
</transacGrp>
```

can be modelled in a feature-value structure as shown in Figure 2.

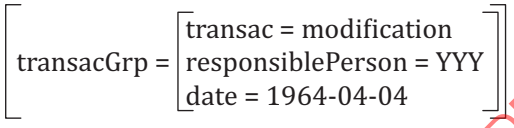

Figure 2 — Feature-value structure

There is also a need to associate semantic information with the content of a data category; this is achieved through *annotations*. A typical example is a definition in which the genus and/or differentia are explicitly marked, as in the following definition for lead pencil:

```
<definition>
    <broaderConcept>pencil</broaderConcept>whose
    <characteristic>casing</characteristic>is fixed around a central
    <characteristic>graphite</characteristic>medium which is
    <characteristic>used for writing or making marks</characteristic>
</definition>
```

Such information cannot be represented as a feature-value structure.

5.2 Generic representation of components and information units

Terminological data can be represented using a generic architecture that consists of a graph of elementary structural nodes to which one or more information units are attached. This architecture is shown in the UML diagram in Figure 3.

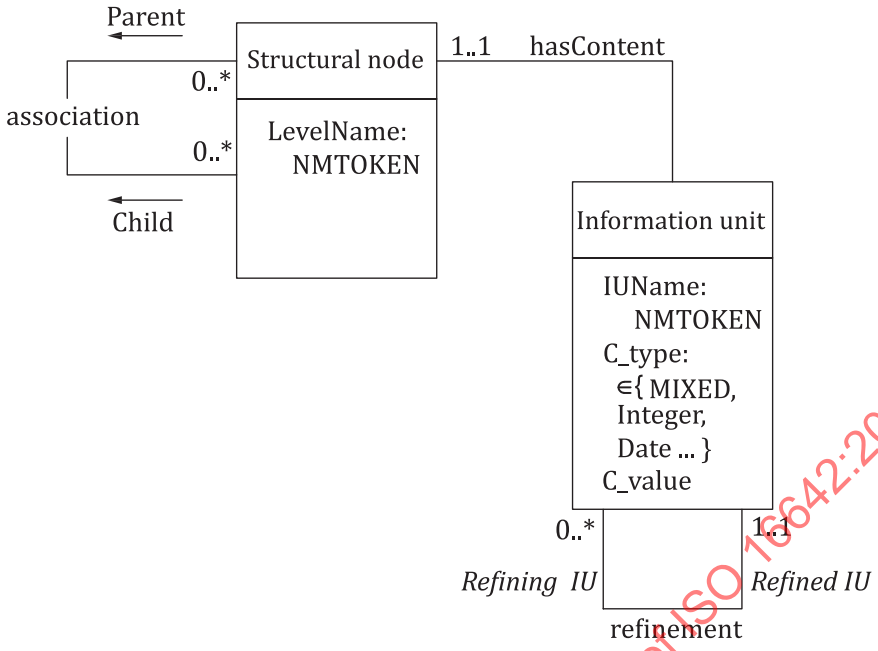

Figure 3 — UML diagram for structural nodes and information units

The diagram expresses the relationship between the following defined classes:

— structural node: a class containing one attribute (LevelName) which identifies objects of this type in the context of a given language resource (for example, TE/Terminological Entry for the representation of terminological data);

— information unit: a class containing three attributes that: a) identify objects of this type in relation to a given data category (IUName, e.g. /definition/, /partOfSpeech/, etc.); b) describe a type for its content (C_type); and c) provide the actual content value (C_value).

The value of C_type can either belong to the set of simple types as defined in *XML Schema Part 2: Datatypes* or be MIXED.

Objects of these two classes can be related in the following ways.

— association: Indicates that a structural node is related to another structural node by a hierarchical link. There is no constraint on the number of links or the structure of the network that those links create (tree, directed acyclic graph, etc.) (0..*);

— hasContent: Relates a structural node to information units (for instance, a /definition/ attached to a TE node (terminological entry)). An instance of an information unit is attached to one and only one structural node (1..1);

— refinement: Relates information units that provide additional information to another information unit (for example, a /note/ refining a /definition/). A refining information unit is related to one and only one refined information unit (1..1). Some TMLs allow more levels of refinement than others, and this affects the degree of interoperability.

The MIXED type is an ordered combination of textual content (strings) and information units, corresponding to any kind of annotated content. It can be represented in UML by means of the aggregation operator, as shown in Figure 4.

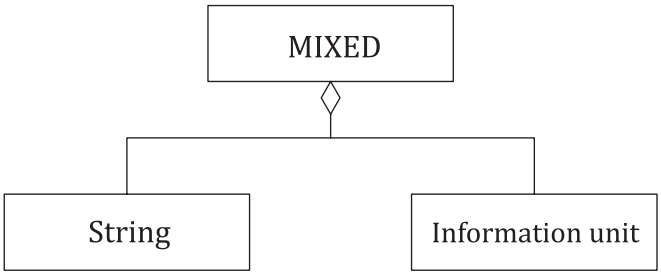

Figure 4 — MIXED object class

Adherence to this definition permits annotations to be refined by other information units (for instance to indicate when and by whom the annotation has been made).

5.3 The metamodel

The terminological metamodel is based on guidelines concerning the methods and principles of terminology management as described in ISO 704. One of the most important characteristics of a terminological entry, compared to a lexicographical entry, is its concept orientation. A terminological entry treats one concept in a given language and, in the case of multilingual terminological entries, one or more totally or partially equivalent concepts in one or more other languages, whereas a lexicographical entry contains one lemma (the base form of a lexical unit) and one or more definitions (representing different meanings) in one or more languages.

Note that some concepts are not universal in that they present slight differences in different languages or cultures. These differences may be significant enough to declare that they form different and distinct concepts. Depending on the degree of conceptual difference and similarity, it may be decided to describe these concepts in the same entry or in different entries.

A terminological data collection (TDC) comprises global information about the collection and a number of entries. Each entry performs three functions.

— It describes a single concept.

— It identifies the terms that designate the concept.

— It describes the terms themselves.

Each terminological entry can have multiple language sections, and each language section can have multiple term sections (terms and their accompanying information). Each data element in an entry can be associated with various kinds of descriptive and administrative information. In addition, there are various other resources that can be referenced by multiple entries. Such shared resources include bibliographic references, descriptions of ontologies, and binary data such as images that illustrate concepts.

The principles of terminology management as described in ISO 704, ISO 26162 and ISO 30042, shall be respected. These include:

— term autonomy;

— concept orientation;

— data elementarity;

— data granularity.

The terminological metamodel is described through seven instances from the structural node class, as shown in Figure 5.

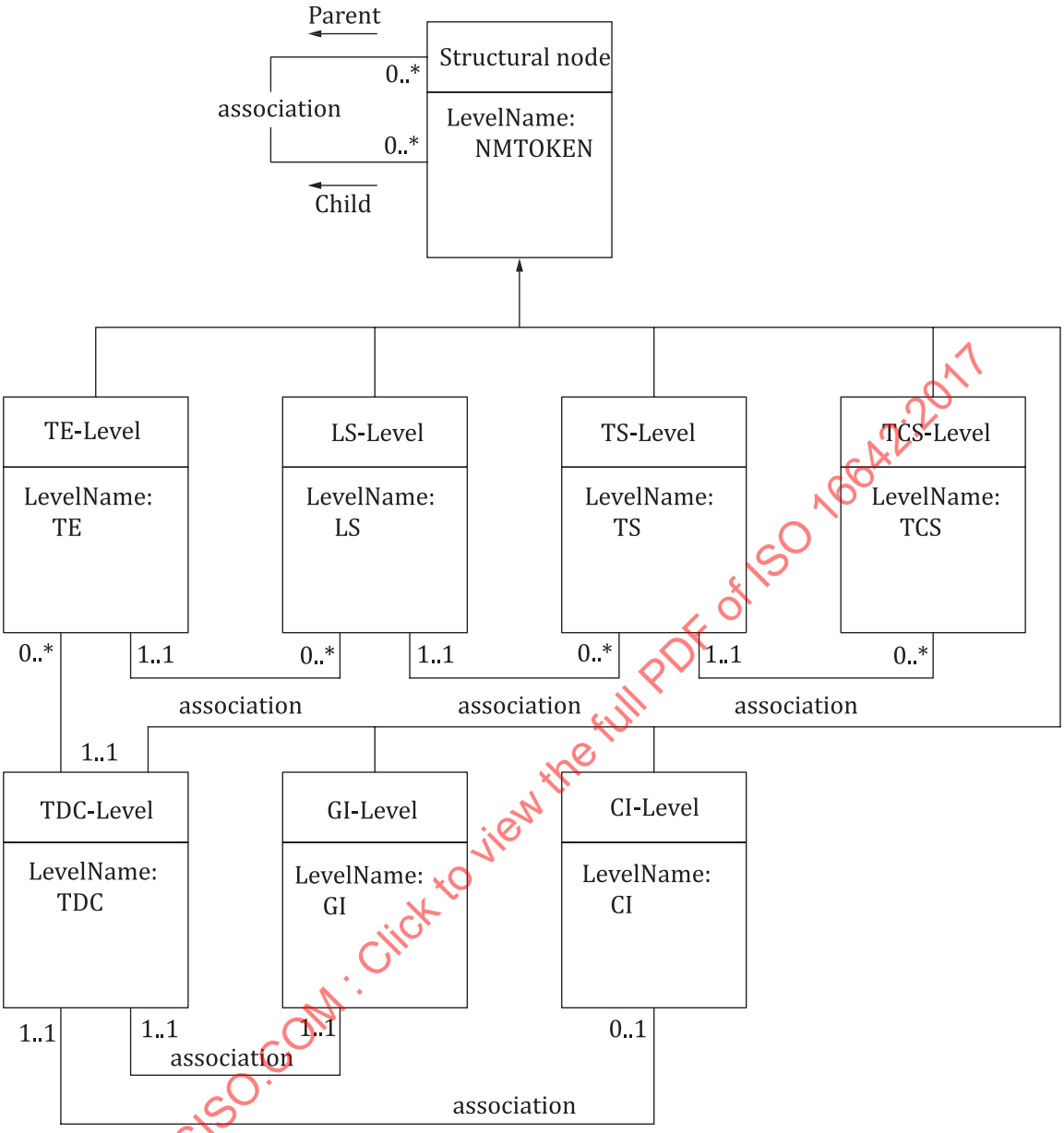

Figure 5 — Terminological metamodel — UML diagram

These seven instances of the structural node class are:

— TDC (terminological data collection): Top level container for all information contained in a terminological data collection.

— GI (global information): Information about the TDC as a whole. The GI section usually contains, for example, the title of the TDC, the institution or individual from which the file originated, address information, copyright information, update information, and so forth.

— TE (terminological entry): Information that pertains to a single concept, or two or more nearly equivalent concepts. The TE section contains descriptive information pertinent to a concept, such as a definition and subject field, and administrative information about the entry.

— LS (language section): The LS is a container for all the term sections of a terminological entry for a given language, as well as information pertaining to the concept in that language. For example, it may contain a definition in the given language, or a note about any specific characteristics of the concept or any specific properties of the object in that language.

— TS (term section): The TS contains exactly one term, and information about the term, such as the part of speech, term type and a context.

— TCS (term component section): Information about parts of a term such as morphemes, phonemes, syllables, or single words from a multiword term. In some languages, such as French or Spanish, it is sometimes necessary to associate information, such as gender, with the individual words used in a multiword term.

— CI (complementary information): The CI section usually contains, for example, bibliographic or administrative information, graphic images, video, audio, or any other kind of binary data. It can also include references to other terminological data collections or links to related text corpora, ontologies, or external URIs. These items are often designated as shared resources because they can be referenced from any terminological entry.

These instances of components implement the "association" relation with constraints on cardinality, which can also be schematized as shown in Figure 6.

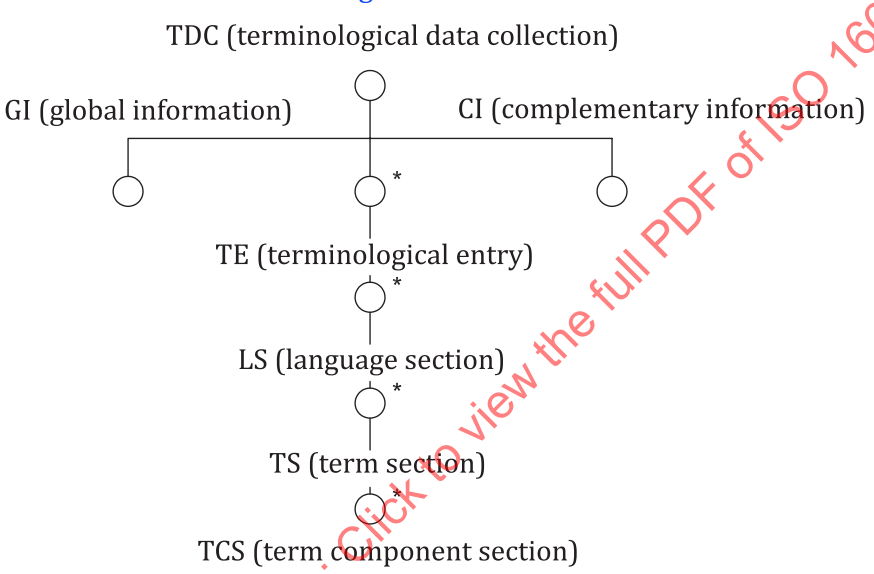

Figure 6 — Terminological metamodel — Schematic view

Cardinality:

— A TDC shall contain exactly one GI (1..1), at most one CI-Level (0..1) and any number of TEs (0..*).

— A TE can contain any number of LSs (0..*).

— An LS can contain any number of TSs (0..*).

— A TS can contain any number of TCSs (0..*).

Hierarchical organization is ensured by the 1..1 limitations expressed between nodes of the metamodel in Figure 5. For example, a term section (TS) is a child of the language section (LS).

5.4 Example

To illustrate how a TDC can be analysed as an abstract structure, consider a simple terminological entry expressed as an XML document conforming to TBX:

```
<termEntry id="eid-Oracle-67">
 <descrip type="subjectField">manufacturing</descrip>
 <descrip type="definition">A value between 0 and 1 used in...</descrip>
 <langSet xml:lang="en">
  <tig>
   <term id="tid-Oracle-67-en1">alpha smoothing factor</term>
```



```
    <termNote type="partOfSpeech">noun</termNote>
   </tig>
  </langSet>
  <langSet xml:lang="hu">
   <tig>
    <term id="tid-Oracle-67-hu1">Alfa simítási tényezo</term>
    <termNote type="partOfSpeech">noun</termNote>
   </tig>
  </langSet>
 </termEntry>
```

An abstract model of this terminological entry can be created by identifying a structural skeleton corresponding to the metamodel and by associating information units with each component in the structural skeleton, as shown in Figure 7.

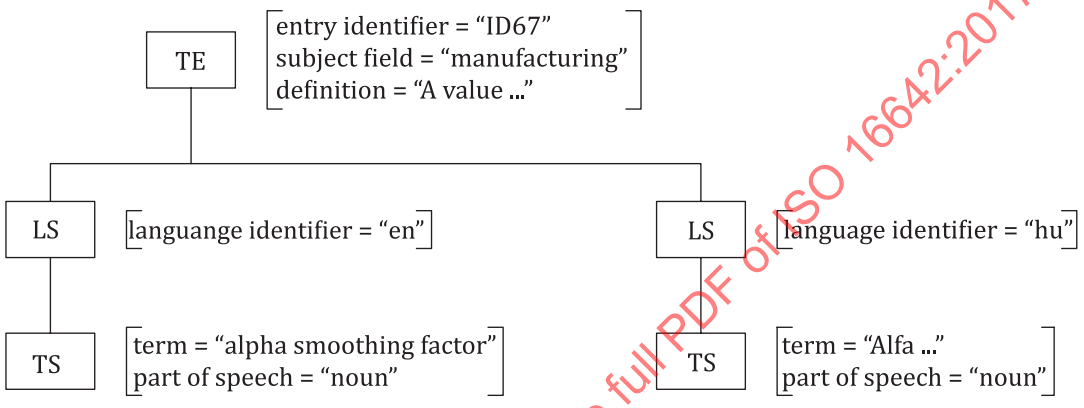

Figure 7 — Mapping an XML document to the abstract model

For describing the DCS of this TML, data categories can be mapped to the corresponding data categories specified in DatCatInfo.

Identifier	PID
entryIdentifier	http://www.datcatinfo.net/rest/dc/206
subjectField	http://www.datcatinfo.net/rest/dc/489
definition	http://www.datcatinfo.net/rest/dc/168
languageIdentifier	http://www.datcatinfo.net/rest/dc/279
term	http://www.datcatinfo.net/rest/dc/508
partOfSpeech	http://www.datcatinfo.net/rest/dc/396
noun	http://www.datcatinfo.net/rest/dc/1333

6 Requirements for compliance to TMF

All computer formats for TMLs shall be based on:

a) the metamodel;

b) a DCS, available from a DCR, such as DatCatInfo.

Each DCS adopted for a TML shall consist of a set of data categories from a DCR, such as DatCatInfo, with, optionally, some additional constraints on each of those data categories. Constraints include restrictions on the permissible values of a data category, such as "text with markup" for /context/ or "picklist" for /grammatical gender/, as well as specific XML datatypes such as for numbers and dates. Constraints also include restrictions on where a particular data category can appear at the different levels (components) of the metamodel. For example, /part of speech/ may be constrained to occur only at the TS level.

Any TDC conforming to TMF shall clearly distinguish between the working language and the object language, which are the two uses of language that can be associated with any level of the collection (see Clause 8).

Annex A provides an example scenario of conformance of terminological data to TMF.

7 Interchange and interoperability

Interchange involves a transfer of information between two computer systems and is often bidirectional. Interchange may require the use of an intermediate format.

The degree of interoperability between two TMLs can be determined by comparing their specifications (combination of the metamodel and their DCS). TMLs that are structured in compliance with the metamodel described in this document and that use the same DCS are said to be interoperable, that is, it will be possible to convert data from one TML to the other TML and back without loss of information. If the DCS of one TML is a subset of the DCS of the other TML, it will be possible to convert data from the former to the latter, but not the reverse.

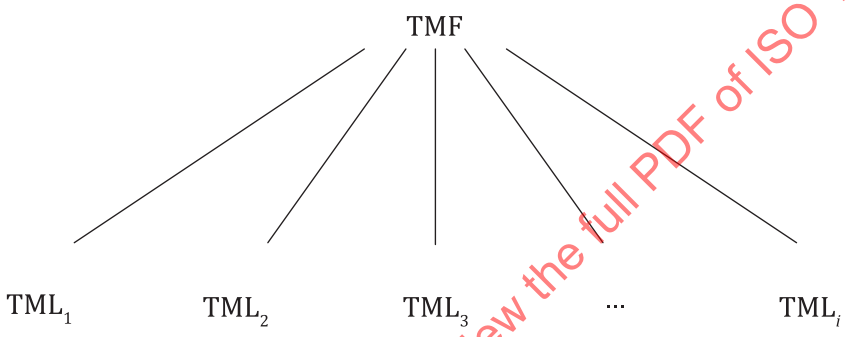

Figure 8 — TMF rendering TMLs more interoperable

8 Representing languages

The working language is the language used to express any given textual content in the TDC, such as a definition or a note. The scope of the working language is the whole sub-document starting from the element where it is declared, until it is superseded by another working language declaration.

The object language is the language that is being described in a language section .

Both the object language and the working language shall be represented using the xml:lang attribute as defined in the Extensible Markup Language (XML) recommendation of the W3C. This attribute, when used at the language section level, indicates both the object language and the working language.

The allowed values of the xml:lang attribute are found in IETF RFC 5646 or its successor, as identified in IETF BCP 47.

The following example shows how the two types of language declarations can be used within a LS expressed in TBX:

```
<langSet xml:lang="en">
 <descrip type="definition" xml:lang="fr">Une valeur entre 0 et 1 utilisée…
</descrip>
 <tig>
  <term>alpha smoothing factor</term>
  <termNote type="termType">fullForm</termNote>
 </tig>
</langSet>
```

9 Defining a TML

9.1 Steps

The following steps are required to define a TML.

a) Select data categories from a DCR, such as DatCatInfo.

b) Create expansion trees associated with the structural nodes in the terminological metamodel.

c) Identify the XML styles and vocabularies required for expressing the data categories assigned to the nodes in the trees.

d) Provide the minimal information for fully specifying the XML schemas controlling the valid instances of the TML.

e) Provide the interoperability conditions required for transforming the TML instance into an instance of another TML.

9.2 Defining interoperability conditions

Interoperability conditions shall be based on the specification of the DCS valid for the TML. The DCS includes data category specifications which express the properties of each data category. These properties include the following:

— the unique data category name;

— the persistent identifier (PID) for the data category specification;

— the structural nodes (level in the metamodel) where the data category may occur;

— the conceptual domain associated with the data category.

10 Implementing a TML

10.1 General

XML data structures for implementing a TML shall be specified in order to create conditions for interoperability. An XML outline shall be created by:

— mapping the structural skeleton of the TML instance to the metamodel, based on the appropriate set of XML elements;

— providing anchoring mechanisms for the various data categories included in the DCS.

10.2 Implementing the metamodel

The structural part of a TML shall be defined by associating an XML expansion tree with each structural node in the metamodel. For each structural node having a parent in the metamodel (i.e. for which there exists a higher level in the metamodel) an anchor shall also be defined which comprises a node in its parent's expansion tree and to which its own expansion tree can be attached.

The XML outline of an instance of a TML comprises all the expansion trees associated with its structural skeleton.

10.3 Anchoring data categories on the XML outline

10.3.1 General

The expansion tree associated with a structural node shall consist of a set of XML element nodes. Each of these nodes shall be a potential anchor for any of the data categories associated with that node. Each data category shall be expressed as a sub-structure of its anchor dependent upon

— its anchoring style, and

— the vocabulary used in the actual TML.

The description of anchors, style and vocabulary properties shall be stated for each data category in the full DCS for the TML.

10.3.2 Styles and vocabulary

Data categories attached to the TML structural skeleton shall be implemented as XML feature-value pairs using one of the following styles:

— Attribute;

— Element;

— TypedElement;

— ValuedElement;

— TypedValuedElement.

The Attribute style shall implement a data category as an attribute of its anchor. The value of the data category is realized as the content of the attribute. For example:

```
<termEntry id="ID67">...
</termEntry>
```

The Element style shall implement the data category as an XML element that is a child of its associated anchor. The value of the data category is realized as the content of the element. For example:

```
<tig>
      <term>alpha smoothing factor</term>
      <partOfSpeech>verb</partOfSpeech>
      <definition>A value between 0 and 1 used in...</definition>
</tig>
```

The TypedElement style shall implement a data category as an XML element that is a child of its associated anchor and that is specified as the value of an attribute *type*. The vocabulary comprises the name of this element and the value of its attribute *type*. The value of the data category is realized as the content of the element. For example:

```
<termEntry>
      <descrip type="subjectField">manufacturing</descrip>
      ....
</termEntry>
```

The ValuedElement style shall implement a data category as an XML element, which is itself a child of a given anchor. The element is further specified by an attribute *value*, which can itself specify another data category. The vocabulary comprises the name of this element, and the value of the attribute *value*. For example:

```
<termEntry>
      <subjectField value="manufacturing"/>
      ....
```



```
</termEntry>
```

The TypedValuedElement style shall implement a data category as an XML element, which is itself a child of a given anchor, and which is further specified by means of an attribute *type* and an attribute *value*. The vocabulary comprises the name of this element and the name of the attributes. The value associated with the data category is realized as the content of the attribute *value*. For example:

```
<termEntry>
      <prop type="subjectField" value="manufacturing"/>
      ...
</termEntry>
```

10.4 Constraints on datatypes

Data categories implemented using the Element or TypedElement style can take values containing additional markup (in particular markup resulting from the implementation of annotations). In contrast, data categories implemented using either the Attribute, ValuedElement, or TypedValuedElement style shall not contain any such markup.

10.5 Implementing annotations

Annotations shall be implemented in the same way (using XML styles and vocabularies) as data categories that are attached to the structural skeleton of a given TML. Only the Element and TypedElement styles support annotations.

10.6 Implementing brackets

Bracketed data categories shall be implemented using an element that provides the name of the main data category of the group and other elements that provide further related data categories such as for administrative information. For example:

```
<adminGrp>
 <admin type="terminologyManagementTransactions">modification</admin>
 <admin type="modificationDate">21-08-2001</admin>
</adminGrp>
```

Annex A
(informative)

Conformance of terminological data to TMF: example scenario

A.1 General

This annex discusses how XML-based terminological data can be made TMF-compliant by analysing the structure and content of the data and performing certain transformations on these data. Through this analysis and transformation, the TML is clearly specified in a way that represents the terminological data without loss of information and renders the TML interoperable with other TMLs.

A.2 Example

Consider the following example of a terminological entry from an automotive engineering TDC. This example does not fully conform to TMF, but with some modifications it can be mapped to TMF.

```
<termBank>
 <tbid>00aa</tbid>
 <tbDescription>Automotive Engineering</tbDescription>
 <conceptEntry>
  <domainOfConcept>ABS</domainOfConcept>
  <conceptLastModified>21-08-2001</conceptLastModified>
  <termGroup>
   <languageCode>Deutsch</languageCode>
   <termDefinition>Bauteile, die die elektronischen Steuer- und Regelvorgänge für die
Blockierregelung und die Antriebsschlupfregelung übernehmen.</termDefinition>
   <termString>ABS/ASR-Steuerung</termString>
   <usageDescriptors>
    <usedIn>Germany</usedIn>
    <usedIn>Switzerland</usedIn>
   </usageDescriptors>
   <wordClass>n</wordClass>
   <wordGender>f</wordGender>
   <termLastModified>21-08-2001</termLastModified>
  </termGroup>
  <termGroup>
   <languageCode>English</languageCode>
   <termString>ABS/ASR control</termString>
   <usageDescriptors>
    <usedIn>Britain</usedIn>
   </usageDescriptors>
   <wordClass>n</wordClass>
   <termLastModified>20-08-2001</termLastModified>
  </termGroup>
 </conceptEntry>
</termBank>
```

Table A.1 describes the information contained in this example.

Table A.1 — Description of content of elements

XML element	Description	Description of content
<tbid>	Unique identifier of this TDC	Alphanumeric code
<tbDescription>	Text describing this TDC	Text
<domainOfConcept>	Subject field of this concept entry	Selected value related to concept