INTERNATIONAL STANDARD

ISO 24611

First edition
2012-11-01

Language resource management — Morpho-syntactic annotation framework (MAF)

Gestion des ressources langagières — Cadre d'annotation morphosyntaxique (MAF)

Reference number
ISO 24611:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web 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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

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

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.

ISO 24611 was prepared by Technical Committee ISO/TC 37, *Terminology and other language and content resources*, Subcommittee SC 4, *Language resource management*.

Introduction

ISO/TC 37/SC 4 focuses on the definition of models and formats for the representation of annotated language resources. To this end, it has generalised the modelling strategy initiated by its sister committee, SC 3, for the representation of terminological data [Romary, 2001], through which linguistic data models are seen as the combination of a generic data pattern (a meta-model), which is further refined through a selection of data categories that provide the descriptors for this specific annotation level. Such models are defined independently of any specific formats, and ensure that an implementer has the necessary conceptual instrument with which to design and compare formats with regard to their degrees of interoperability.

One important aspect of representing any kind of annotation is the capacity to provide a clear and reliable semantics for the various descriptors used, either in the form of formal features and feature values, or directly as objects in a representation that is expressed, for instance, in XML. In order to be shared across various annotation schemas and encoding applications, such a semantics should be implemented as a centralised registry of concepts: we will henceforth refer to these as data categories. As such, data categories should bear the following constraints.

— From a technical point of view, they must provide unique, stable references (implemented as persistent identifiers, in the sense of ISO 24619) such that the designer of a specific encoding schema can refer to them in his or her specification. By doing so, two annotations will be deemed to be equivalent when they are in fact defined in relation to the same data categories (as feature and feature value).

— From a descriptive point of view, each unique semantic reference should be associated with precise documentation combining a full text elicitation of the meaning of the descriptor with the expression of specific constraints that bear upon the category.

In recent years, ISO has developed a general framework for representing and maintaining such a registry of data categories, encompassing all domains of language resources. This initiative, described in ISO 12620, has led to the implementation of an online environment providing access to all data categories that have been standardized in the context of the various language resource-related activities within ISO, or specifically as part of the maintenance of the data category registry. It also provides access to the various data categories that individual language technology practitioners have defined in the course of their own work and decided to share with the community.

The ISO data category registry, as available through the ISOCat (www.isocat.org) implementation, is intended as a 'flat' marketplace of semantic objects, providing only a limited set of ontological constraints. The objective there is to facilitate the maintenance of a comprehensive descriptive environment where new categories are easily inserted and reused without the need for any strong consistency check with the registry at large. Indeed, the following basic constraints are part of the data category model, as defined in ISO 12620:

— simple generic-specific relations, when these are useful for the proper identification of interoperability descriptors between data categories. For instance, the fact that /properNoun/ is a sub-category of /noun/ makes it possible to compare morpho-syntactic annotations based on different descriptive levels of granularity;

— the description of conceptual domains, in the sense of ISO 11179, to identify, when known or applicable, the possible value of so-called complex data categories For instance, it can be used to record that possible values of /grammaticalGender/ (limited to a small group of languages [Romary 2011]), could be a subset of {/masculine/, /feminine/ and /neutral/};

— language-specific constraints, either in the form of specific application notes or as explicit restrictions bearing upon the conceptual domains of complex data categories. For instance, it is possible to express explicitly that /grammaticalGender/ in French can only take the two values: {/masculine/ and /feminine/}.

This International Standard provides a comprehensive framework for the representation of morpho-syntactic (also referred to as part-of-speech) annotations. Such an annotation level corresponds to a first lexical abstraction level over language data (textual or spoken) and, depending on the language to be annotated, together with the characteristics of the annotation tool or annotation scheme that is being used, can vary enormously in structure and complexity.

In order to deal with such complex issues as ambiguity and determinism in morpho-syntactic annotation, this International Standard introduces a meta-model that draws a clear distinction between the two levels of tokens (representing the surface segmentation of the source) and word-forms (identifying lexical abstractions associated with groups of tokens). These two levels share the following specificities: on the one hand, they can be represented as simple sequences and as local graphs such as multiple segmentations and ambiguous compounds; on the other hand, any n-to-n combination can stand between word forms and tokens.

As linguistic segments (sometimes called 'markables' in the literature [see, for instance, Carletta et al. 1997]), *tokens* may be embedded in the source document as inline mark-up, or they may point remotely to it by means of so-called stand-off annotations.

As linguistic abstractions, *word-forms* can be qualified by various linguistic features characterising the morpho-syntactic properties that are instantiated in the realisation of the lexical entry within the annotated text. Such properties may range from the simple indication of a lemma up to an explicit reference to a lexical entry in a dictionary. In most existing applications of morpho-syntactic annotation, linguistic properties are expressed by means of so-called tags; these codes refer to basic feature structures (see early examples in Monachini and Calzolari, 1994). Such codes may also provide morphological information, including its part of speech (e.g. noun, adjective or verb), and features such as number, gender, person, mood and verbal tense.

In keeping with the general modelling strategy of ISO/TC 37, this International Standard/MAF provides means of relating morpho-syntactic tags expressed as feature structures (compliant with ISO 24610) to the data categories available in ISOCat. A normative annex of this International Standard elicits a core set of data categories that can be used as reference for most current morpho-syntactic annotation tasks in a multilingual context. However, when implementers of this International Standard find these categories inappropriate in either coverage, scope or semantics, they are encouraged to use ISOCat to define their own categories in compliance with ISO/TC 37 principles.

Associated to the meta-model, MAF also provides a default XML syntax that may be used to serialise MAF-compliant annotation models. Since many existing projects are based on the text encoding initiative (TEI) guidelines (www.tei-c.org) — particularly in digital humanities, where a proper encoding of textual sources is essential — this International Standard will also provide clues about how to articulate the MAF model with TEI-compliant encodings. Indeed, the TEI guidelines already offer a variety of constructs and mechanisms to cope with many issues relevant to spoken corpora and their annotations (Romary and Witt, 2012).

Finally, it should be noted here that this International Standard forms the conceptual basis for the development of the ISO 24614 series on word segmentation, whereby all general principles and rules defined in ISO 24614-1 as well as the constraints expressed in additional parts for specific languages, are to be understood according to the token–word-form dichotomy.

Language resource management — Morpho-syntactic annotation framework (MAF)

1 Scope

This International Standard provides a framework for the representation of annotations of word-forms in texts; such annotations concern tokens, their relationship with lexical units, and their morpho-syntactic properties.

It describes a metamodel for morpho-syntactic annotation that relates to a reference to the data categories contained in the ISOCat data category registry (DCR, as defined in ISO 12620). It also describes an XML serialization for morpho-syntactic annotations, with equivalences to the guidelines of the TEI (text encoding initiative).

2 Normative references

The following referenced documents are indispensable for the application 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 24610-1, *Language resource management — Feature structures — Part 1: Feature structure representation*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 24610-1 and the following apply.

3.1
DAG
directed acyclic graph
graph with directed edges and no cycles

Note 1 to entry: DAGs are a subset of *finite state automata* (3.4).

3.3
feature structure
set of feature specifications, used in the morpho-syntactic annotation framework (MAF) to express morpho-syntactic content

Note 1 to entry: Feature structures are described in ISO 24610-1.

3.4
FSA
finite state automata
graphs made up of states with an initial state and a final state, and a finite set of transitions from state to state

Note 1 to entry: See also *DAG* (3.1).

3.5
grapheme
minimal unit in a written language

EXAMPLE Letter, pictogram, ideogram, numeral, punctuation.

3.6
inflection
modification or marking of a lexeme that reflects its morpho-syntactic properties

3.7
inflected form
form that a word can take when used in a sentence or a phrase

Note 1 to entry: An inflected form of a word is associated with a combination of morphological features, such as grammatical number and case.

3.8
lemma
lemmatised form
conventional form chosen to represent a lexeme

Note 1 to entry: In European languages, the lemma is usually the *singular* if there is a variation in number, the *masculine* form if there is a variation in *gender*, and the *infinitive* for all verbs. In some languages, certain nouns are defective in the singular form; in these cases, the plural is chosen. For verbs in Arabic, the lemma is usually deemed to be the third person singular with the accomplished aspect.

3.9
lexeme
morpheme generally associated with a set of word-forms sharing a common meaning

3.10
lexical entry
container for managing a set of word-forms and possibly one or more meanings to describe a lexeme

3.11
lexicon
resource comprising a collection of lexical entries for a language

3.12
morpheme
smallest linguistic unit that carries a meaning in a discourse, but which cannot be divided into smaller meaningful units

Note 1 to entry: A morpheme is either grammatical (grammeme) or lexical (lexeme).

3.13
morphological feature
morpho-syntactic feature
feature induced from the inflected form of a word

Note 1 to entry: The ISOCat data category registry provides a comprehensive list of values for European languages.

EXAMPLE “grammaticalGender”.

3.14
morphology
description of the structure and formation of word-forms

3.15
morpho-syntactic tag
tag
feature structure used systematically to qualify a word-form

3.16
tagset
comprehensive set of tags used for the morpho-syntactic description of a language

Note 1 to entry: The ISOCat data category registry is to be used as the reference for describing a tagset.

3.17
part of speech
grammatical category
category assigned to a word based on its grammatical and semantic properties

EXAMPLE Noun, verb.

Note 1 to entry: The ISOCat data category registry provides a comprehensive list of values for parts of speech.

3.18
phoneme
minimal unit in the sound system of a language

3.19
script
set of graphic characters used for the written form of one or more languages

3.20
syntagmatic relation
relation by which linguistic units in a discourse are associated

3.21
token
non-empty contiguous sequence of graphemes or phonemes in a document

Note 1 to entry: For editorial reasons, some annotation scheme may extend the notion of token to an empty sequence. See the section on token attachment (6.2).

3.22
tokenization
process identifying tokens

3.23
transcription
form resulting from a coherent method of writing down speech sounds

3.24
transliteration
form resulting from the conversion of one script into another, usually through a one-to-one correspondence between characters

3.25
word-form
morpho-syntactic unit
contiguous or non-contiguous linguistic unit identified as corresponding to a lexical entity in a language

Note 1 to entry: Word-forms may have no acoustic or graphic realization, or may correspond to one or more tokens.

3.26
word lattice
set of possible alternative decompositions of a text or speech segment into word-forms

Note 1 to entry: A word lattice has the algebraic properties of a directed acyclic graph with an initial node and a final node.

Note 2 to entry: See also *DAG* (3.1) and *FSA* (3.4).

4 The MAF meta-model

4.1 Overview

Morpho-syntactic annotations provide an important layer of linguistic information in a document. This International Standard is based on a meta-model that draws a clear distinction between the two levels of tokens (representing the surface segmentation of the source) and word-forms (identifying lexical abstractions associated to groups of tokens). These two levels share the following specificities: on the one hand, they can be represented as simple sequences and as local graphs (e.g. multiple segmentations and ambiguous compounds); on the other hand, any n-to-n combination can stand between word-forms and tokens. This International Standard delimits minimal and maximal sequences in documents (either text or speech) that can be identified as word-forms and seeks to categorise the linguistic and distributional criteria that may be used to mark these word-forms within some larger syntagmatic context. Minimal units cannot be further decomposed using similar criteria, but may however be divided into smaller units using morphological or phonological properties. Word-forms can be aggregated to form maximal units (such as compound words or multi-word units) that act as elementary units for other levels of linguistic analysis, particularly syntax. In particular, word-forms correspond to the non-terminal level defined in ISO 24615.

4.2 MAF Meta-model

Figure 1 presents a simplified view of the proposed meta-model for morpho-syntactic annotations, whereas Figure 2 presents a more formal view based on UML (Unified Modeling Language).

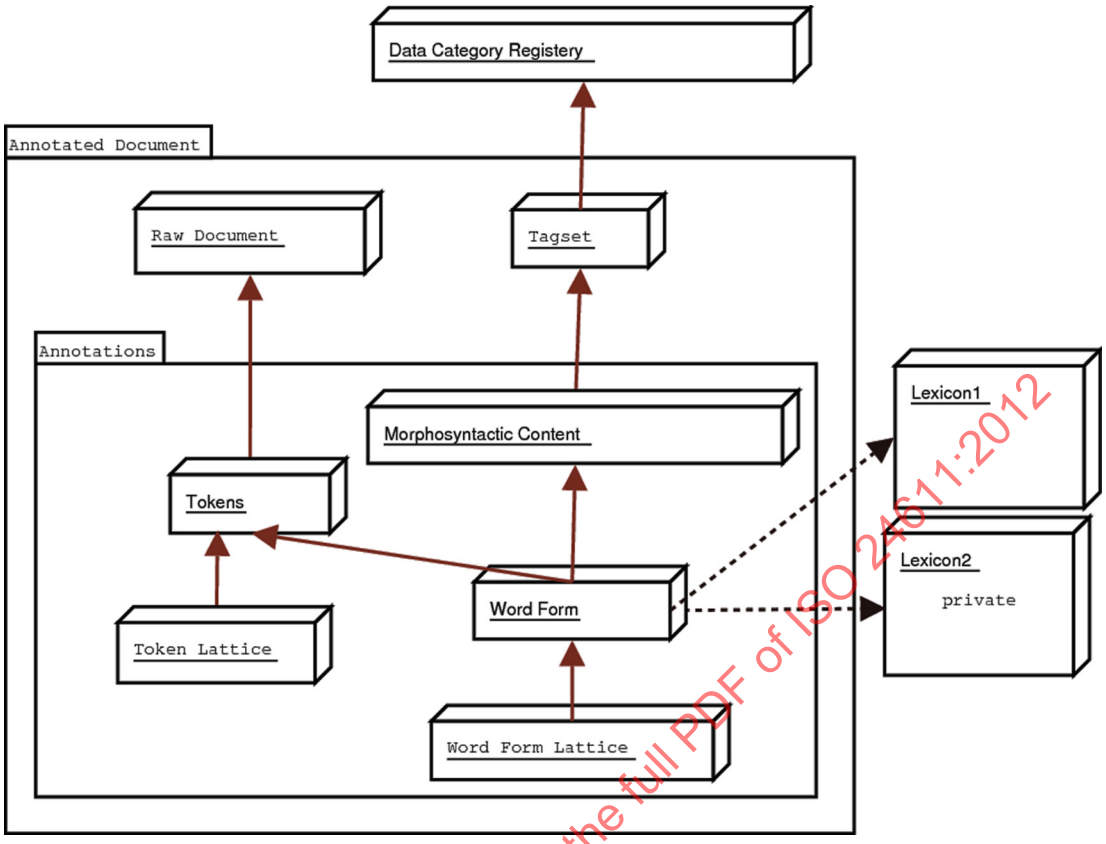

Figure 1 — Simplified view of MAF meta-model

An annotated document comprises an original document and a set of annotations. Annotations are associated with word-forms corresponding to zero or more tokens in the original document. A word-form may also be associated with a lexical entry providing information about its underlying lemma and inflected form. The morpho-syntactic annotation associated with a word-form is represented by a tag, the significance of which may be expressed as a feature structure. A set of such tags used by a particular annotation scheme is referred to as a tagset, and corresponds to what is defined in the ISO 24610-2-specified feature structures representation (FSR) as a feature structure library. Each discrete category within such a tagset should be describable in terms of registered data categories as described in ISO 12620 and implemented in ISOCat. Because annotation may be applied both to tokens and to word-forms, structural ambiguity is likely. Hence annotation is typically conceptualised as one or more streams, each represented as a word lattice or more formally as a directed acyclic graph (DAG).

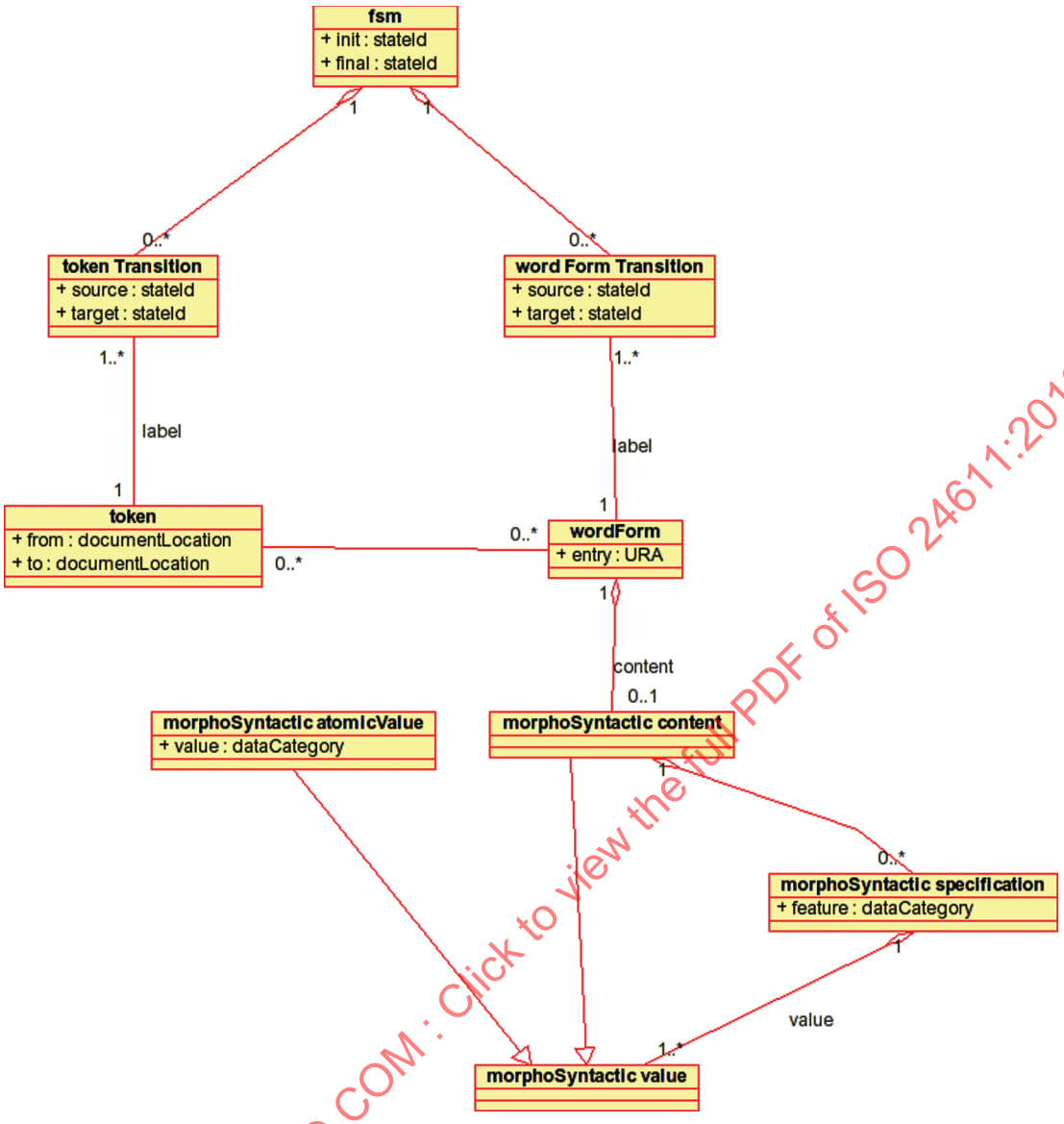

Figure 2 — UML view of MAF meta-model

5 Segmenting with tokens

5.1 General

Morpho-syntactic annotations are carried by segments, called tokens, that are present in the document flow, but this does not imply that the resulting segmentation corresponds to a sequence of adjacent segments partitioning the original document. It is particularly important to distinguish the word-forms from their realisations. Some parts of a document may carry no annotations (e.g. typographic marks, stage directions and markup elements) while other parts may not correspond exactly to their segmented form (e.g. abbreviations, brachygraphies, orthographic errors and variations, and typographic and morphological contractions). A word-form may not correspond exactly to a segment identified by orthographic marks such as white spaces or hyphens (e.g. for German compound words, speech transcription and Sanskrit writing).

The following list shows typical examples of tokenised inputs in two languages, with the original linguistic segment followed by the representation of tokens as vertical bar-separated strings:

La petite fille

La|petite|fille

白菜和猪肉

白|菜|和|猪|肉

The element <token> is used to represent those segments of the original document which, in approximate terms, follow orthographic, morphological, or phonological boundaries. This International Standard does not define the linguistic properties of tokens. In different languages, a token may be identified through its typographic properties (white-space, hyphens or characters), its phonological properties (e.g. linking phenomena, hiatus, elision and final-obstruent devoicing such as the "Auslautverhärtung" in German), its morphological properties (radical, affix, morpheme etc.), or by all of them. The description of the orthographic, morphological, phonological and lexical structures that may define a token is not covered by this International Standard.

Also not covered by this International Standard are those aspects of a writing system that are used to format pages or to separate words and paragraphs, and provide similar encoding information, since these do not constitute morpho-syntactic annotation.

5.2 Formal description: <token>

The token level in MAF is implemented by means of the <token> element. This is formally defined as follows.

— <token> element used to mark tokens as defined in 3.21:

@from	Left span boundary
@to	Right span boundary
@join	Relationship with neighbouring tokens

— att.token.information attributes used to provide additional information about the content of a token:

@form	normalised form of the token
@phonetic	phonetic transcription
@transcription	general transcription
@transliteration	transliteration to some other script

5.3 Embedding notation

It is not always necessary to separate the original document from its annotations. In simple cases, textual content may be directly embedded within <token> elements in the form of an inline annotation. An example is shown in Figure 3.

```
<token xml:id="t1">The</token>
<token xml:id="t2">victim</token>
<token xml:id="t3">'s</token>
<token xml:id="t4">friends</token>
<token xml:id="t5">told</token>
<token xml:id="t6">police</token>
<token xml:id="t7">that</token>
<token xml:id="t8">Krueger</token>
<token xml:id="t9">drove</token>
<token xml:id="t10">into</token>
<token xml:id="t11">the</token>
<token xml:id="t12">quarry</token>
<token xml:id="t13">and</token>
<token xml:id="t14">never</token>
<token xml:id="t15">surfaced</token>
<token xml:id="t16">.</token>
```

Figure 3 — Inline annotation of tokens for the sentence 'The victim's friends told the police that Krueger drove into the quarry and never surfaced.' (en)

Although this inline notation is used for most of the examples provided for MAF, it may pose problems in certain circumstances, for example where the treatment of white space characters in XML has not been properly taken into account, or more significantly in the presence of other conflicting hierarchies. In such circumstances it may be preferable to define the content of a token using stand-off notation.

5.4 Alternate representation for TEI based documents

For representations that are based on texts or transcriptions encoded according to the TEI guidelines, the <w> element has to be used within the TEI namespace (http://www.tei-c.org/ns/1.0) in order to implement the token level of the MAF meta-model. For punctuation marks, the <pc> element has to be used. This is illustrated by the example shown in Figure 4.

```
<p>
 <w xml:id="t1">The</w>
 <w xml:id="t2">victim</w>
 <w xml:id="t3">'s</w>
 <w xml:id="t4">friends</w>
 <w xml:id="t5">told</w>
 <w xml:id="t6">police</w>
 <w xml:id="t7">that</w>
 <w xml:id="t8">Krueger</w>
 <w xml:id="t9">drove</w>
 <w xml:id="t10">into</w>
 <w xml:id="t11">the</w>
 <w xml:id="t12">quarry</w>
 <w xml:id="t13">and</w>
 <w xml:id="t14">never</w>
 <w xml:id="t15">surfaced</w>
 <pc xml:id="t16">.</pc>
</p>
```

Figure 4 — Tokenised text encoded in compliance with the TEI guidelines

5.5 Stand-off notation

The content of a <token> element may also be defined independently of the original document by referencing an interval within the stream of characters constituting the document. The @*from* and @*to* attributes are used to define such intervals. The value of these attributes depends both on a chosen addressing schema to denote non-ambiguous document positions and on the nature of the original document. In the simplest cases, character offsets may be sufficient. ISO 24612 makes use of anchors to refer to locations in between the base units of the data representation. Assuming a document beginning 'The victim's friends...' we might therefore represent the first four tokens as shown in Figure 5.

```
 Locations in the document:
 |T|h|e| |v|i|c|t|i|m|'|s| |f|r|i|e|n|d|s|
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
                     1                   2

<token from="0" to="3"/>
<!-- The -->
<token from="4" to="10"/>
<!-- victim -->
<token from="10" to="12"/>
<!-- 's -->
<token from="13" to="20"/>
<!-- friends -->
```

Figure 5 — Serialisation of a stand-off tokenisation

When the tokenisation is to be implemented within a TEI encoded document, the syntax shown in Figure 6 has to be applied.

```
<s xml:id="s1">The victim's friends</s>
<s>
 <w corresp="#string-range(s1,0,3)"/>
<!-- The -->
 <w corresp="#string-range(s1,4,6)"/>
<!-- victim -->
 <w corresp="#string-range(s1,10,2)"/>
<!-- 's -->
 <w corresp="#string-range(s1,13,7)"/>
<!-- friends -->
</s>
```

Figure 6 — Stand-off annotation in compliance to the TEI guidelines

5.6 Informative attributes

Tokens identify segments of the original document, either by enclosing them or by pointing to them in a stand-off manner, through the specification of a start and end point within the character stream that the document consists of. It is also often useful to associate with each token some additional abstract information, for instance concerning graphical or phonological variations, even if it is not linguistically relevant. The non-mandatory attributes @*form*, @*phonetic*, and @*transliteration* may be used to perform such abstraction.

These attributes are typically used to provide a preferred normalisation in the presence, for example, of variations, a phonetic transcription, or a transliteration into some other writing system. See Figure 7.

```
<token form="etcetera">etc.</token>
<token
  form="etcetera"
  phonetic="/etsettrə/"
  from="1251"
  to="1253"/>
<token form="tzar">csar</token>
<token form="tzar">tsar</token>
<token transliteration="tsar">цар</token>
<token form="2003-02-23">February 23rd 2003</token>
```

Figure 7 — Usage examples of the *@form* attribute

The abstraction provided by the *@form* attribute is also appropriate for dealing with the phenomena of contraction and agglutination, where two tokens may cover the same segment of the original document while having distinct values (see 5.7.2, on overlapping tokens).

5.7 Completing the inline token notation

As mentioned above, inline token notation is less precise than stand-off notation, in particular because it does not make the contiguity and overlapping of tokens fully explicit. In the absence of strict conventions for the interpretation of the embedded token notation, more precise identification of segments is possible using document positions, as in the case of stand-off notation.

5.7.1 Joining tokens in embedded mode

By convention, two sibling tokens are considered to be separated by whatever separator is standard for the document language, for instance, a space. This convention may be modified by using the *@join* attribute to specify how a token is joined to its sibling tokens. The attribute *@join* takes the value 'left' or 'right' to indicate that a token is contiguous with its preceding and following siblings respectively, rather than separated from them.

```
<token>L'</token>
<token join="left">on</token>
<token>dit</token>
```

Figure 8 — Tokenisation of 'L'on dit' (fr – 'it is said ...')

The value 'both' is used to show that a token is contiguous with both left and right siblings, as may be the case if it encloses material usually considered as separator, such as spaces, newline, dash or apostrophe (see Figure 9).

```
<token>L</token>
<token join="both">'</token>
<token>on</token>
<token>dit</token>
```

Figure 9 — Tokenisation of 'L'on dit' (fr – 'it is said ...')

Another example, in Modern Greek, is provided by the idiomatic expression 'καλοκαγαθός' (el – 'good and brave'). This expression may be regarded as three agglutinated segments καλός, και, and αγαθός and represented by Figure 10.

```
<token form="καλὸς">καλο</token>
<token form="και" join="left">κ</token>
<token form="αγαθὸς" join="left">αγαθὸς</token>
```

Figure 10 — Tokenisation of 'καλοκαγαθὸς' (el – 'good and brave') with the *@join* attribute

5.7.2 Overlapping tokens

Two tokens may overlap, for instance to denote an agglutinated or contracted form. In such cases, a <token> may not only mark the realisation of a typographical or vocal sequence, but also express a deeper linguistic reality relevant for segmenting a document. However, it is more usual to disregard the overlap at the token level of analysis and to defer consideration of the issue to the level of linguistic analysis ('word-forms') as discussed in the next section.

The value overlap for the *@join* attribute may be used to denote overlapping at the level of inline tokens. For instance, in 'etc.', the dot may be seen as a token that overlaps: it is used both within the abbreviation and as a punctuation mark. This may be encoded as shown in Figures 11 and 12.

```
<token form="etcetera" from="0" to="3"/>
<token form="." from="3" to="3"/>
```

Figure 11 — Stand-off notation

```
<token form="etcetera">etc.</token>
<token form="." join="overlap"/>
```

Figure 12 — Inline notation

6 Word-forms as linguistic units

The segments identified by <token> elements correspond to word-forms. A word-form may be associated with a lexical entry in a lexicon using the *@entry* attribute (see 6.3). It may also be characterised by qualifying part-of-speech information that expresses morphological and grammatical properties as a feature structure (see 7.2). Information about the lemma and inflected forms may also be provided, using the *@lemma* and *@form* attributes. In particular, the *@form* attribute on the <wordForm> element is useful when the inflected form attached to the word-form does not coincide with the content of the <token> element, for instance because of spelling corrections.

A token may be associated with more than one word-form and, conversely, a word-form may represent more than one token. The *@tokens* attribute is used to associate a <wordForm> element with one or more <token> elements. The association is represented by means of a pointer, typically supplying the value of the *@xml:id* attribute on one or more <token> elements.

For instance, in French, the morphological agglutination of 'auquel' ('to which') may have several representations, depending on the granularity of the tokenisation.

When the priority is put on coarse granularity, the character sequence 'auquel' is not decomposed and corresponds to a single token, but there are two different word-forms associated with this single token. It is encoded as shown in Figure 13.

```
<token xml:id="t0">auquel</token>
<wordForm lemma="à" tag="#pos.prep" tokens="#t0"/>
<wordForm lemma="lequel" tag="#pos.pronrel" tokens="#t0"/>
```

Figure 13 — Encoding of 'auquel' – coarse granularity

With a fine granularity perspective, the tokeniser identifies two agglutinated parts represented by two tokens, each of them associated with a word-form, as shown in Figure 14.

```
<token form="a" xml:id="t0">auquel</token>
<token form="lequel" xml:id="t1" join="overlap"/>
<wordForm lemma="à" tag="#pos.prep" tokens="#t0"/>
<wordForm lemma="lequel" tag="#pos.pronrel" tokens="#t1"/>
```

Figure 14 — 'auquel'

The level of granularity chosen may be motivated by the intended application or by the tools available for a given language.

As mentioned before, there are no mandatory linguistic properties for defining the tokens; for example, they can be automatically recognised by regular languages. On the other hand, a word-form that may cover zero or more tokens should represent a linguistic unit carrying morpho-syntactic information.

This International Standard does not address the linguistic choices that define these linguistic units but it provides enough flexibility to annotate them. The choice may be motivated by lexical or morphological properties based on context and language, depending on the nature and function of the words.

6.1 Formal description: <wordForm>

— <wordForm> linguistic units built upon <tokens>:

@tokens	List of tokens for the corresponding <wordForm> element
@tag	Reference to one or more morpho-syntactic descriptions expressed as feature structures
@lemma	Lemma attached to a wordForm
@form	Form attached to a wordForm
@entry	Lexical entry attached to a wordForm

6.2 Token attachment

6.2.1 One token; one word-form

The simplest case of a relationship between tokens and word-forms is when one word-form corresponds to a single token, as shown in Figure 15.

```
<token xml:id="t10">apple</token>
<wordForm lemma="apple" tokens="#t10"/>
```

Figure 15 — 'apple'

6.2.2 Several contiguous tokens; one word-form

In a slightly more complex case, a single word-form (a compound word) corresponds to more than one adjacent token, as shown in Figure 16.

```
<token xml:id="t20">prime</token>
<token xml:id="t21">minister</token>
<wordForm lemma="prime_minister" tokens="#t20 #t21"/>
```

Figure 16 — 'Prime Minister'

6.2.3 Several discontinuous tokens; one word-form

A sequence of non-contiguous tokens may also be associated with a single word-form, for instance to handle cases where some material is inserted inside the components of a word-form, as shown in Figure 17.

```
<token xml:id="t31">afin</token>
<token xml:id="t32">justement</token>
<token xml:id="t33">de</token>
<wordForm lemma="afin_de" tokens="#t31 t33"/>
<wordForm lemma="justement" tokens="#t32"/>
```

Figure 17 — Discontinuous tokens

This kind of phenomenon may also occur with verbs with detached particles, for instance in English or German. The English infinitive verbal form (to + <verb>) may also fall into this scheme (Figure 18)

```
<token xml:id="t41">to</token>
<token xml:id="t42">boldly</token>
<token xml:id="t43">go</token>
<wordForm lemma="to_go" tokens="#t41 #t43"/>
<wordForm lemma="boldly" tokens="#t42"/>
```

Figure 18 — Discontinuous tokens

In order to identify discontinuous word-forms while preserving some information about the position of each component in the flow of word-forms, word-forms may be used covering the same sequence tokens and referring to the same entry (and possibly sub-entries), as in Figure 19.

```
<token xml:id="t41">to</token>
<token xml:id="t42">boldly</token>
<token xml:id="t43">go</token>
<wordForm entry="urn:lexicon:en:go:to" tokens="#t41 #t43"/>
<wordForm entry="urn:lexicon:en:boldly" tokens="#t42"/>
<wordForm entry="urn:lexicon:en:go:main" tokens="#t41 #t43"/>
```

Figure 19 — Discontinuous tokens with lexical references

6.2.4 Zero token; one word-form

Another case that may arise is when it is desired to insert a word-form that is not realised in the original document and that is therefore associated with an empty sequence of tokens. Use cases include some pronouns in Spanish, and the hypothesis of traces. This kind of annotation practice is highly dependent on the linguistic assumption of the corresponding project. The relative position of the pronoun is determined here (Figure 20) by its order in the sequence of word-forms.

```
<token xml:id="t51">Jean</token>
<token xml:id="t52">propose</token>
<token xml:id="t53">de</token>
<token xml:id="t55">partir</token>
<wordForm lemma="Jean" tokens="#t51"/>
<wordForm lemma="proposer" tokens="#t52"/>
<wordForm lemma="de" tokens="#t53"/>
<wordForm lemma="PRO"/>
<wordForm lemma="partir" tokens="#t55"/>
```

Figure 20 — Word-form with no surface realization

6.2.5 One token; several word-forms

Finally, several word-forms may be continued into the same token, as illustrated by the examples shown in Figures 21 and 22.

```
<token form="dammelo" xml:id="t61">Dammelo</token>
<wordForm lemma="dare" tokens="#t61"/>
<wordForm lemma="mi" tokens="#t61"/>
<wordForm lemma="lo" tokens="#t61"/>
```

Figure 21 — Multiple word-forms in ‘dammelo’ (it – ‘Give it to me’)

```
<token xml:id="t70">auquel</token>
<wordForm lemma="à" tag="#pos.prep" tokens="#t70"/>
<wordForm lemma="lequel" tag="#pos.pronrel" tokens="#t70"/>
```

Figure 22 — Multiple word-forms in ‘auquel’ (fr – ‘to which’)

6.3 Referring to lexical entries

A word-form is a linguistic unit carrying morpho-syntactic properties. Generally, a linguistic unit may be characterised by a label corresponding to an entry in a lexicon. As shown earlier, the *@entry* attribute may be used to supply a reference to a lexical entry in the form of a URN (uniform reference name). See Figure 23.

```
<token xml:id="t21">Prime</token>
<token xml:id="t22">minister</token>
<wordForm entry="urn:lexicon:en:prime_minister" tokens="#t21 #t22"/>
```

Figure 23 — Simple reference to a lexical entry

Depending on the editorial environment of the corresponding digital lexicon, entries may be referred to by means of a URN (as exemplified above) or, when applicable, by a full URI that directly points to the corresponding entry (as exemplified in TEI based representations). It should be noted that it is also possible to refer to lexical ‘sub-entries’ for polysemous entries or for compound forms. See Figure 24.

```
<token xml:id="t71">to</token>
<token xml:id="t72">eventually</token>
<token xml:id="t73">decide</token>
<wordForm entry="urn:lexicon:en:decide:to" tokens="#t71 #t73"/>
<wordForm entry="urn:lexicon:en:eventually" tokens="#t72"/>
<wordForm entry="urn:lexicon:en:decide:main" tokens="#t71 #t73"/>
```

Figure 24 — Complex reference to multiple lexical entries

A token or a sequence of tokens may sometimes be identified as forming a word-form because of various properties but cannot be associated with any lexical entry, perhaps because no lexicon is available, or because the word-form corresponds to a named entity (e.g. a proper name, date or address) or because the word-form is a neologism. In that case, the *@entry* attribute may be omitted. The other informative attributes *@lemma* and *@form* may still be used to provide information about the word-form. See Figure 25.

```
<token xml:id="t80">October</token>
<token xml:id="t81">,</token>
<token xml:id="t82">23rd</token>
<token xml:id="t83">2005</token>
<wordForm lemma="DATE" form="2005/10/23" tokens="#t80 #t81 #t82 #t83"/>
```

Figure 25 — Lexicalised entity

In a production system, such unknown words may be collected in an additional document-specific lexicon, which may then be referenced in the same way.

6.4 Compound word-forms

The structure of compound forms (including multi-word expressions) may be expressed using nested word-forms, thus providing information about the subparts even when no information is available for the whole, for instance for neologisms (Figure 26).

```
<token form="Geburtstag" xml:id="t91" join="right">Geburtstags</token>
<token form="Geschenk" xml:id="t92" join="right">geschenk</token>
<token form="Papier" xml:id="t93">papier</token>
<wordForm tokens="#t91 #t92 #t93">
 <wordForm entry="urn:lexicon:de:geburstag" lemma="geburtstag" tokens="#t91"/>
 <wordForm entry="urn:lexicon:de:geschenk" lemma="geschenk" tokens="#t92"/>
 <wordForm entry="urn:lexicon:de:papier" lemma="papier" tokens="#t93"/>
</wordForm>
```

Figure 26 — Representing compounds - ‘Geburtstagsgeschenkpapier’ (de - ‘Birthday gift wrapping paper’)

The definition of compounding or derivational morphology is outside the scope of MAF.

6.5 Identification of word-forms within a TEI-compliant document

When morpho-syntactic annotation is inserted within a TEI document, the word-form component of the MAF meta-model should be implemented by means of a element, subject to the following constraints:

— elements may be grouped together within one or more <spanGrp> elements;

— the element must have a @type attribute with value wordForm, with this @type attribute able to be moved to the encompassing <spanGrp> element when applicable;

— the @target attribute should contain the list of URIs which refer to the tokens that the word-form is associated with;

— the @corresp attribute can be used to point to a lexical entry (encoded as an <entry> element);

— the @ana attribute can be used to point to a feature structure (or a feature-value library entry, see below) that provides the morpho-syntactic content of the word-form.

See Figure 27.

```
<p>
 <w xml:id="w1">I</w>
 <w xml:id="w2">wanna</w>
 <w xml:id="w3">put</w>
 <w xml:id="w4">up</w>
 <w xml:id="w5">new</w>
 <w xml:id="w6">wallpaper</w>
 <pc>.</pc>
</p>
<spanGrp type="wordForm">
 <span target="#w1" ana="#fs1" corresp="#entry1"/>
 <span target="#w2" ana="#fs2" corresp="#entry2"/>
 <span target="#w2" ana="#fs3" corresp="#entry3"/>
 <span target="#w3 #w4" ana="#fs4" corresp="#entry4"/>
 <span target="#w5" ana="#fs5" corresp="#entry5"/>
 <span target="#w6" ana="#fs6" corresp="#entry6"/>
</spanGrp>
<fs xml:id="fs1" corresp="#entry1">
 <f name="lemma">
  <string>I</string>
 </f>
 <f name="pos">
  <symbol value="PP"/>
 </f>
</fs>
<fs xml:id="fs2">
 <f name="lemma">
  <string>want</string>
 </f>
 <f name="pos">
  <symbol value="VBP"/>
 </f>
</fs>
<fs xml:id="fs3">
 <f name="lemma">
  <string>to</string>
 </f>
 <f name="pos">
  <symbol value="TO"/>
 </f>
</fs>
<fs xml:id="fs4">
 <f name="lemma">
  <string>put up</string>
 </f>
 <f name="pos">
  <symbol value="VB"/>
 </f>
</fs>
```

```
<fs xml:id="fs5">
 <f name="lemma">
  <string>new</string>
 </f>
 <f name="pos">
  <symbol value="JJ"/>
 </f>
</fs>
<fs xml:id="fs6">
 <f name="lemma">
  <string>wallpaper</string>
 </f>
 <f name="pos">
  <symbol value="NN"/>
 </f>
</fs>
<entry xml:id="entry1">
 <form type="lemma">
  <orth>I</orth>
 </form>
</entry>
<entry xml:id="entry2">
 <form type="lemma">
  <orth>want</orth>
 </form>
</entry>
<entry xml:id="entry3">
 <form type="lemma">
  <orth>to</orth>
 </form>
</entry>
<entry xml:id="entry4">
 <form type="lemma">
  <orth>put up</orth>
 </form>
</entry>
<entry xml:id="entry5">
 <form type="lemma">
  <orth>new</orth>
 </form>
</entry>
<entry xml:id="entry6">
 <form type="lemma">
  <orth>wallpaper</orth>
 </form>
</entry>
```


Figure 27 — Full annotation of the sentence 'I wanna put up new wallpaper' (en), in compliance with the TEI guidelines (with simplified lexical entries, for the sake of illustration)

7 Morpho-syntactic content

7.1 General

This clause explains how to attach morpho-syntactic content to word-forms, how to define reusable tagsets in order to provide compact notations by means of tags, and how to control the validity of such tags.

The previous section explained how to enrich a document with morpho-syntactic annotations, but did not define the content of these annotations. What set of features and feature values should be used to express such content, and how should it be interpreted?

Such a set, which is usually referred to as a tagset, specifies the range of possible annotations. The diversity of approaches and languages makes the definition of any single universally applicable tagset almost impossible. In this context, this International Standard provides mechanisms for defining tagsets by using the ISO data category registry (DCR) and FSR.

An annotated document should include or reference a tagset defining the content of its annotations.

7.2 Using feature structures

A <wordForm> element may be provided with morpho-syntactic content that defines its linguistic nature and its grammatical function in the current context. Such content may be expressed using feature structures, following the ISO 24610-1. A feature structure may attach one or more (possibly complex) values to linguistic properties (for instance, noun to part of speech, present to tense, and indicative to mood). See Figure 28.

```
<token xml:id="t01">belle</token>
<wordForm entry="urn:lexicon:fr:beau" lemma="beau" tokens="#t01">
 <fs>
  <f name="pos">
   <symbol value="adjective"/>
  </f>
  <f name="adj_type">
   <symbol value="qualifier"/>
  </f>
  <f name="gender">
   <symbol value="feminine"/>
  </f>
  <f name="number">
   <symbol value="singular"/>
  </f>
 </fs>
</wordForm>
```

Figure 28 — Feature structure-based annotation for ‘belle’ (fr - ‘beautiful’)

The feature structure content attached to a word-form may provide a wide range of additional information of interest about a word-form.

7.3 Compact morpho-syntactic tags

ISO 24610-1 addresses the compact representation of feature structures based on libraries that assign names to feature values and feature specifications (feature-value pairs). These names may be used by the *@tag* attribute on the <wordForm> element to simplify the tagging, following standard practice in the natural language processing community (Figure 29).

```
<token xml:id="t0">belle</token>
<wordForm
  tokens="t0"
  entry="urn:lexicon:fr:beau"
  tag="#pos.adj #adj_type.qual #gender.fem #num.sg"/>
```

Figure 29 — Morpho-syntactic annotation for 'belle' (fr - 'pretty') using the *@tag* attribute

The value of the *@tag* attribute is syntactically identical to the value of the *@feats* attribute defined in ISO 24610-2, namely a space-separated sequence of feature specification URIs.

The libraries where recurrent values and feature specifications are defined constitute the tagset(s) supplied with, or referenced by, the annotated document.

7.4 FSR libraries

The generic library mechanism provided by FSR is illustrated by the example, using the *@feats* attribute of element <fs>, shown in Figures 30 and 31.

```
<fvLib n="French morpho values">
 <symbol xml:id="noun" value="noun"/>
 <symbol xml:id="sing" value="singular"/>
 <symbol xml:id="plu" value="plural"/>
 <symbol xml:id="masc" value="masculine"/>
 <symbol xml:id="fem" value="feminine"/>
</fvLib>
```

Figure 30 — A feature-value library

```
<fLib>
 <f xml:id="pos.n" name="pos" fVal="#noun"/>
 <f xml:id="num.sg" name="number" fVal="#sing"/>
 <f xml:id="num.p" name="number" fVal="#plu"/>
 <f xml:id="gen.f" name="gender" fVal="#fem"/>
 <f xml:id="gen.m" name="gender" fVal="#masc"/>
</fLib>
```

Figure 31 — A feature specification library

As stated in ISO 12620, it is recommended that the *@dcr:datcat* and *@dcr:valueDatcat* attributes on the corresponding <f> elements link features declared in the FSR library, or a FSD (feature system declaration) to the appropriate data category references in ISOCat, be used.

The two libraries can be the basis for morpho-syntactic annotations as exemplified in Figure 32.

```
<wordForm
  tokens="#t1 #t2"
  lemma="prime_minister"
  tag="#pos.n #num.sg #gen.f"/>
```

Figure 32 — Morpho-syntactic encoding of 'Prime Minister' (en - 'Premier ministre') using feature-value libraries

Disjunctive values are allowed by FSR and may also be simplified, following the same mechanism (Figure 33).

```
<tagset>
 <fvLib>
  <vAlt xml:id="firstOrThird">
   <symbol value="first"/>
   <symbol value="third"/>
  </vAlt>
  <symbol xml:id="verb" value="verb"/>
  <symbol xml:id="sg" value="singular"/>
 </fvLib>
 <comment>A feature specification library</comment>
 <fLib>
  <f xml:id="pers.13" name="pers" fVal="#firstOrThird"/>
  <f xml:id="pos.v" name="pos" fVal="#verb"/>
  <f xml:id="num.sg" name="number" fVal="#sg"/>
 </fLib>
</tagset>
<comment>Annotated document</comment>
<token xml:id="t100">porte</token>
<wordForm
  tokens="#t100"
  entry="urn:lexicon:fr:porter"
  tag="#pos.v #pers.13 #num.sg"/>
```

Figure 33 — A feature-value library

7.5 Designing tagsets

The features, values and, possibly, feature types used to specify morpho-syntactic content are not simply labels; they also carry linguistic meanings or, to put it another way, semantic content. To avoid misinterpretation or incoherence, the semantic content attached to a feature, value or type should be clearly defined. The combination of features, values and types should also be controlled in order to avoid linguistically invalid combinations, such as using 'neuter' as a value for 'gender' in French, or using a 'tense' feature for nouns in most languages.

This International Standard does not provide the semantic content for all such features, values and types. It would be almost impossible, given the diversity of languages, and it would be equally impossible to assign to each component a meaning that the whole community agreed on.

Instead, it is proposed that an annotated document should be completed by including or referencing one or more tagsets.

The first objective of a tagset is to list the terminology used when annotating a document as a set of data categories, the meanings of which are precisely defined in a data category registry in line with ISO 12620 and the definition of a *data category registry* (as implemented in ISOCat). The process may be seen as selecting a subset of morpho-syntactic data categories (data category selection, or DCS), as shown in Figure 34.

```
<tagset>
 <dcs
   local="genre"
   registered="http://www.isocat.org/datcat/DC-1297"
   rel="eq"/>
 <dcs
   local="fem"
   registered="http://www.isocat.org/datcat/DC-1880"
   rel="eq"/>
</tagset>
```

Figure 34 — Tagset declaration in relation to the DCR

The correspondence to a registered data category may not be perfect. The *@rel* attribute may be used to specify the kind of relationship that exists between the local and registered data categories. For instance, it is possible to introduce a local data category 'advneg' as being subsumed by a more general registered data category 'adverb'. See Figure 35.

```
<dcs local="advneg" registered="dcs:morphosyntax:pos:adverb" rel="subs"/>
<dcs local="strange" rel="none"/>
```

Figure 35 — Approximate reference to the DCR

It is also possible to introduce a local data category that bears no relationship to any registered data category (Figure 36), although this is not generally recommended.

```
<dcs local="title"/>
```

Figure 36 — Local data category declaration

When the correspondence is not perfect or missing, a description should be added to define the meaning of a local data category (Figure 37).

```
<dcs local="title">
 <description> A part of speech used to denote honorific titles like
   Pr. or S.A.S. </description>
</dcs>
```

Figure 37 — Local data category declaration with description

The second objective of a tagset is to specify the set of valid feature structures based on the selected data categories. This is achieved by means of the ISO 24610-2-specified FSD.

The third objective of a tagset is to name the most common morpho-syntactic structures through the use of FSR libraries (7.4).

7.6 Formal description: <tagset>

— <tagset> tagset to be used to check and interpret the annotations

— <dcs/> DCS: the selection of data categories used to express the annotations

@local	local name of the category
@registered	registered name of the category in the ISO data category registry
@rel	(relationship) relationship between the local meaning of a category and the registered one
@desc	(description) informal description of a data category

8 Handling ambiguities

Ambiguities naturally arise when handling natural language, especially in the case of automatically produced annotations. They may also occur at various levels, and MAF therefore proposes several alternatives to cope with ambiguities as simply as possible.

8.1 Word-form content ambiguities

ISO 24610-1 provides several ways of representing ambiguities, for instance at the level of feature values. These mechanisms may be used to handle ambiguities occurring within the morpho-syntactic content of a word-form.

For example, the French inflected verb form ‘mange’ (eat) is ambiguous between the first and third persons. This ambiguity can be captured by the <vAlt> element provided by FSR:

```
<token xml:id="t0">mange</token>
<wordForm tokens="t0" entry="urn:lexicon:fr:manger">
 <fs>
  <f name="pos">
   <symbol value="verb"/>
  </f>
  <f name="aux">
   <symbol value="avoir"/>
  </f>
  <f name="mood">
   <symbol value="indicative"/>
  </f>
  <f name="tense">
   <symbol value="present"/>
  </f>
  <f name="person">
   <vAlt>
    <symbol value="first"/>
    <symbol value="third"/>
   </vAlt>
  </f>
  <f name="number">
   <symbol value="singular"/>
  </f>
 </fs>
</wordForm>
```

Figure 38 — Person ambiguity in the description of a word-form

A compact tag notation can still be used by registering the most frequent cases of ambiguities in FSR libraries (7.4). See Figure 39.

```
<token xml:id="t101">mange</token>
<wordForm
  tokens="#t101"
  entry="urn:lexicon:fr:manger"
  tag="#pos.v #aux.avoir #mood.i #tense.p #pers.13 #num.sg"/>
```

Figure 39 — 'mange'

8.2 Lexical Ambiguities

Ambiguities between different lexical entries corresponding to a single sequence of tokens can be handled by the element <wfAlt>, as shown in Figure 40.

```
<token xml:id="t102">porte</token>
<wfAlt>
 <wordForm tokens="#t102" entry="lexicon:porte" tag="#pos.n ..."/>
 <wordForm tokens="#t102" entry="lexicon:porter" tag="#pos.v ..."/>
</wfAlt>
```

Figure 40 — Lexical ambiguity on a word-form for 'porte' [fr, which can be a noun (en 'door') or a verb ('porter' fr for 'to carry')]

8.3 Structural ambiguities

8.3.1 Structural ambiguities with word-forms

A generic answer is to describe the possible readings as paths through a DAG whose edges are labelled by a word-form. Such DAGs form a subcase of FSA, and also correspond to the notion of word lattice used in the parsing and speech recognition communities. They are powerful enough to facilitate ambiguities between several decompositions into compound forms. They can also be used to denote simpler cases of lexical ambiguity.

MAF introduces the element <fsm> (for finite state machine) to represent such lattices. An <fsm> element contains a sequence of <transition> elements each of which represents an edge of the DAG.

For instance, the French textual sequence 'fer à cheval' (fr, 'horse shoe') can still be read in several ways, for example, ('[horse shoe]', '[iron] [on horse]' or '[iron] [of] [horse]', giving the DAG shown in Figure 41; see also Figure 42.

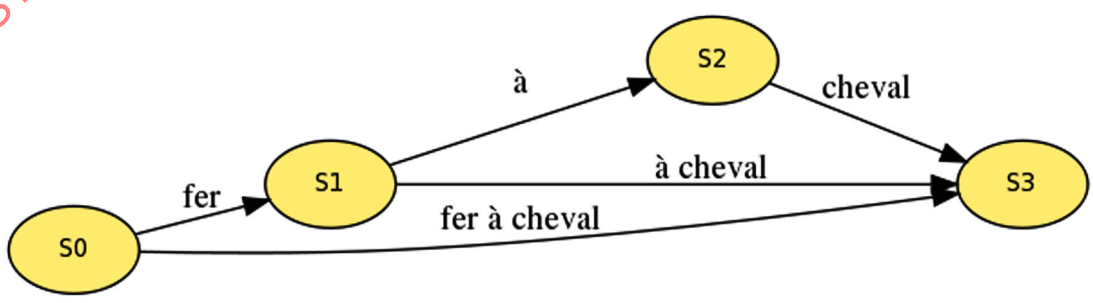

Figure 41 — DAG for fer à cheval

```
<token xml:id="t301">fer</token>
<token xml:id="t302">à</token>
<token xml:id="t303">cheval</token>
<fsm init="S0" final="S3">
 <transition source="S0" target="S3">
  <wordForm
    tokens="#t301 #t302 #t303"
    entry="urn:lex:fr:fer_%E0_cheval"
    lemma="fer_à_cheval"/>
 </transition>
 <transition source="S0" target="S1">
  <wordForm entry="urn:lex:fr:fer" tokens="#t301"/>
 </transition>
 <transition source="S1" target="S2">
  <wordForm tokens="#t302" entry="urn:lex:fr:%E0" lemma="à"/>
 </transition>
 <transition source="S2" target="S3">
  <wordForm tokens="#t303" entry="urn:lex:fr:cheval"/>
 </transition>
 <transition source="S1" target="S3">
  <wordForm tokens="#t302 #t303" entry="urn:lex:fr:%E0_cheval" lemma="à_cheval"/>
 </transition>
</fsm>
```

Figure 42 — Structural ambiguity in word-forms

The linguistic units ‘fer à cheval’, ‘fer’, ‘à’, ‘cheval’ and ‘à cheval’ correspond to minimal syntagmatic units that can be annotated.

Additional information, such as probabilities, can be added to edges.

8.3.2 Structural ambiguities with tokens

Structural ambiguities may also arise with sequences of tokens, resulting from ambiguities in the tokenisation of the annotated document (e.g. speech documents).

Structural ambiguities with tokens are represented by transitions labelled by tokens. The *@tinit* and *@tfinal* attributes on the <fsm> element are used to state the initial and final states for the token paths.

The two levels of structural ambiguities are represented by two lattices that form a kind of chart. It is not mandatory but recommended that the two lattices share their states whenever possible.

A validity condition has to be expressed between the two levels of structural ambiguity, namely that the tokens covered by word-forms along a word-form path belong to some token path.

8.4 Simplified structuring variants

8.4.1 Non-ambiguous linear representation

When there is no ambiguity, the generic lattice notation proposed by MAF must be replaced by a much simpler linear notation in which the <token>, <wordForm> and <wfAlt> elements are implicitly chained according to the order in which they appear, as illustrated in the example shown in Figure 43.

```
<token xml:id="t311">fer</token>
<token xml:id="t312">à</token>
<token xml:id="t313">cheval</token>
<wordForm entry="urn:lex:fr:fer" tokens="#t311"/>
<wordForm entry="urn:lex:fr:à" tokens="#t312"/>
<wordForm entry="urn:lex:fr:cheval" tokens="#t313"/>
```

Figure 43 — Simplified unambiguous representation

8.4.2 Mixed linear and lattice representation

Ambiguities are generally localised, and it is therefore appropriate to place the lattice notation only where it is needed. In this way, MAF allows the <fsm> element to appear at any point within a linear sequence of <token>, <wordForm> and <wfAlt> elements. See Figure 44.

```
<token xml:id="t400">afin</token>
<token xml:id="t401">de</token>
<fsm init="s0" final="s2">
 <transition source="s0" target="s2">
  <wordForm tokens="#t400 #t401" entry="urn:lex:fr:afin_de" tag="#pos.prep"/>
 </transition>
 <transition source="s0" target="s1">
  <wordForm tokens="#t400" entry="urn:lex:fr:afin" tag="#pos.prep"/>
 </transition>
 <transition source="s1" target="s2">
  <wordForm tokens="#t401" entry="urn:lex:fr:de" tag="#pos.prep"/>
 </transition>
</fsm>
<token xml:id="t52">grandir</token>
<wordForm entry="urn:lex:fr:grandir" tag="#pos.verb ..." tokens="#t52"/>
<token xml:id="t53">,</token>
<wordForm entry="lexicon:,\" tag="#pos.ponct" tokens="#t53"/>
<token xml:id="t54">il</token>
<wordForm entry="urn:lex:fr:il" tag="#pos.pronoun ..." tokens="#t54"/>
<token xml:id="t55">mange</token>
<wordForm tokens="#t55" entry="urn:lex:fr:manger" tag="#pos.verb ..."/>
<token xml:id="t56">des</token>
<wordForm
  tokens="#t56"
  entry="urn:lex:fr:une"
  form="des"
  tag="#pos.det #num.pl ..."/>
<token xml:id="t57">pommes</token>
<token xml:id="t58">de</token>
<token xml:id="t59">terre</token>
<fsm init="s8" final="s11">
 <transition source="s8" target="s11">
  <wordForm
    tokens="#t57 #t58 #t59"
    entry="urn:lex:fr:pomme_de_terre"
    tag="#pos.noun ..."/>
 </transition>
 <transition source="s8" target="s9">
  <wordForm tokens="#t57" entry="urn:lex:fr:pomme" tag="#pos.noun ..."/>
 </transition>
 <transition source="s9" target="s10">
  <wordForm tokens="#t58" entry="urn:lex:fr:de" tag="#pos.prep"/>
 </transition>
```

```
 <transition source="s10" target="s11">
  <wordForm tokens="#t59" entry="urn:lex:fr:terre" tag="#pos.noun ..."/>
 </transition>
</fsm>
```

Figure 44 — Mixed representation

8.5 Expanding the simplified variants

Simplified variants are allowed because they may always be expanded into a global lattice by applying the steps outlined in the following subclauses.

8.5.1 Separating tokens and word-forms

All tokens embedded within a word-form may be extracted and moved just before the word-form (and before an enclosing <wfAlt>) without changing the relative order of the <token> elements, as shown in Figure 45, which may also be represented as in Figure 46.

```
<wordForm entry="urn:lex:fr:manger" tag="#pos.verb ...">
 <token xml:id="t60">mange</token>
</wordForm>
```

Figure 45 — Embedded representation

```
<token xml:id="t60">mange</token>
<wordForm entry="urn:lex:fr:manger" tag="#pos.verb ..." tokens="#t60"/>
```

Figure 46 — Disembedded representation

Tokens embedded in word-forms, themselves embedded in transitions, should not be handled, because no clear semantics can be attached to this case.

8.5.2 Wrapping into local lattices

Tokens and word-forms outside transitions are embedded into local lattices as <wfAlt> elements are considered to be word-forms, the representation shown in Figure 47, which becomes that of Figure 48.

```
<token xml:id="t64">il</token>
<wordForm entry="urn:lex:fr:il" tag="#pos.pronoun ..." tokens="#t64"/>
<token xml:id="t65">mange</token>
<wordForm entry="urn:lex:fr:manger" tag="#pos.verb ..." tokens="#t65"/>
<token xml:id="t66">des</token>
```

Figure 47 — Linear representation

```
<fsm
  tinit="s0"
  tfinal="s1"
  init="s0"
  final="s0">
 <transition source="s0" target="s1">
  <token xml:id="t64">il</token>
 </transition>
</fsm>
<fsm
  init="s0"
  final="s1"
  tinit="s0"
  tfinal="s0">
 <transition source="s0" target="s1">
  <wordForm entry="urn:lex:fr:il" tag="#pos.pronoun ..." tokens="#t64"/>
 </transition>
</fsm>
<fsm
  tinit="s0"
  tfinal="s1"
  init="s0"
  final="s0">
 <transition source="s0" target="s1">
  <token xml:id="t65">mange</token>
 </transition>
</fsm>
<fsm
  init="s0"
  final="s1"
  tinit="s0"
  tfinal="s0">
 <transition source="s0" target="s1">
  <wordForm entry="urn:lex:fr:manger" tag="#pos.verb ..." tokens="#t65"/>
 </transition>
</fsm>
```

Figure 48 — Local lattice representation

Lattice state names are local to the narrowest containing lattice.

8.5.3 Merging local lattices

Two adjacent lattices may be merged by renaming the intermediary states in order to avoid name clashes, and in such a way that the word-form (or token) final state of the first lattice equals the word-form (or token) initial state of the second lattice. When merging, it is recommended that the lattice states are renamed in such a way that, wherever possible, the initial and final states for tokens and word-form coincide.

The example of Figure 48 then becomes that shown in Figure 49 and in turn Figure 50.

```
<fsm
  tinit="s0"
  tfinal="s1"
  init="s0"
  final="s1">
 <transition source="s0" target="s1">
  <token xml:id="t64">il</token>
 </transition>
 <transition source="s0" target="s1">
  <wordForm entry="urn:lex:fr:il" tag="#pos.pronoun ..." tokens="#t64"/>
 </transition>
</fsm>
<fsm
  tinit="s0"
  tfinal="s1"
  init="s0"
  final="s1">
 <transition source="s0" target="s1">
  <token xml:id="t5">mange</token>
 </transition>
 <transition source="s0" target="s1">
  <wordForm entry="urn:lex:fr:manger" tag="#pos.verb ..." tokens="#t65"/>
 </transition>
</fsm>
```

Figure 49 — Source description before merge

```
<fsm
  tinit="s0"
  tfinal="s2"
  init="s0"
  final="s2">
 <transition source="s0" target="s1">
  <token xml:id="t64">il</token>
 </transition>
 <transition source="s0" target="s1">
  <wordForm entry="urn:lex:fr:il" tag="#pos.pronoun ..." tokens="#t64"/>
 </transition>
 <transition source="s1" target="s2">
  <token xml:id="#t65">mange</token>
 </transition>
 <transition source="s1" target="s2">
  <wordForm entry="urn:lex:fr:manger" tag="#pos.verb ..." tokens="#t65"/>
 </transition>
</fsm>
```

Figure 50 — Merged representation

8.5.4 Removing <wfAlt>

A transition concerning a lexical ambiguity encoded by a <wfAlt> element may be expanded into two equivalent simpler transitions: that shown in Figure 51 becomes Figure 52.

```
<transition source="s0" target="s1">
 <wfAlt>
  <wordForm tokens="#t0" entry="lexicon:porte" tag="#pos.noun ..."/>
  <wordForm tokens="#t0" entry="lexicon:porter" tag="#pos.verb ..."/>
 </wfAlt>
</transition>
```

Figure 51 — Source-alternate representation

```
<fsm init="s0" final="s1">
 <transition source="s0" target="s1">
  <wordForm tokens="#t0" entry="urn:lex:fr:porte" tag="#pos.noun ..."/>
 </transition>
 <transition source="s0" target="s1">
  <wordForm tokens="#t0" entry="urn:lex:fr:porter" tag="#pos.verb ..."/>
 </transition>
</fsm>
```

Figure 52 — Resulting removal of <wfAlt>

The ordering of transitions inside lattices is not significant, unlike the ordering of word-forms and tokens outside lattices. The relative ordering of local lattices is also significant.

8.6 Formal description: <wfAlt> and <fsm>

— <fsm> (finite state machine) used to describe an ambiguous flow of <token> and/or <wordForm> elements:

@init	initial state of the FSM wrt wordForms
@final	final state of the FSM wrt wordForms
@tinit	initial state of the FSM wrt tokens
@tfinal	final state of the FSM wrt tokens

— <transition> FSM transition in a flow of tokens and/or wordForms:

@source	source state of a transition
@target	target state of a transition

— <wfAlt> (WordForm alternative) simplified form used to represent an alternative between several word-forms.

Annex A
(informative)

Encoded example using the MAF serialization

```
<maf xmlns="http://www.iso.org/ns/MAF">
 <token xml:id="t1">I</token>
 <token xml:id="t2" join="right">wan</token>
 <token xml:id="t3" join="left">na</token>
 <token xml:id="t4">put</token>
 <token xml:id="t5">up</token>
 <token xml:id="t6">new</token>
 <token xml:id="t7">wall</token>
 <token xml:id="t8">paper</token>
 <token xml:id="t9">.</token>
 <wordForm lemma="I" tokens="#t1">
  <fs>
   <f name="pos">
    <symbol value="PP"/>
   </f>
  </fs>
 </wordForm>
 <wordForm lemma="want" tokens="#t2">
  <fs>
   <f name="pos">
    <symbol value="VBP"/>
   </f>
  </fs>
 </wordForm>
 <wordForm lemma="to" tokens="#t3">
  <fs>
   <f name="pos">
    <symbol value="TO"/>
   </f>
  </fs>
 </wordForm>
 <wordForm tokens="#t2 #t3"/>
 <wordForm lemma="put" tokens="#t4"/>
 <wordForm lemma="up" tokens="#t5"/>
 <wordForm lemma="put_up" tokens="#t4 #t5">
  <fs>
   <f name="pos">
    <symbol value="VB"/>
   </f>
  </fs>
 </wordForm>
 <wordForm lemma="new" tokens="#t6">
  <fs>
   <f name="pos">
    <symbol value="JJ"/>
   </f>
  </fs>
 </wordForm>
 <wordForm lemma="wallpaper" tokens="#t7 #t8">
  <fs>
   <f name="pos">
```

```
      <symbol value="NN"/>
     </f>
    </fs>
   </wordForm>
  </maf>
```

Figure A.1 — Inline encoding

```
<maf xmlns="http://www.iso.org/ns/MAF" document="sample.txt" addressing="char_offset">
 <token
   xml:id="t1"
   form="I"
   from="0"
   to="1"/>
 <token
   xml:id="t2"
   join="right"
   form="wan"
   from="2"
   to="5"/>
 <token
   xml:id="t3"
   join="left"
   form="na"
   from="5"
   to="7"/>
 <token
   xml:id="t4"
   form="put"
   from="8"
   to="11"/>
 <token
   xml:id="t5"
   form="up"
   from="12"
   to="14"/>
 <token
   xml:id="t6"
   form="new"
   from="15"
   to="18"/>
 <token
   xml:id="t7"
   form="wall"
   from="19"
   to="23"/>
 <token
   xml:id="t8"
   form="paper"
   from="23"
   to="28"/>
 <token
   xml:id="t9"
   form="."
   from="28"
   to="29">.</token>
 <wordForm lemma="I" tokens="#t1">
  <fs>
   <f name="pos">
```

```
    <symbol value="PP"/>
   </f>
  </fs>
 </wordForm>
 <wordForm lemma="want" tokens="#t2">
  <fs>
   <f name="pos">
    <symbol value="VBP"/>
   </f>
  </fs>
 </wordForm>
 <wordForm lemma="to" tokens="#t3">
  <fs>
   <f name="pos">
    <symbol value="TO"/>
   </f>
  </fs>
 </wordForm>
 <wordForm tokens="t2 t3"/>
 <wordForm lemma="put" tokens="#t4"/>
 <wordForm lemma="up" tokens="#t5"/>
 <wordForm lemma="put_up" tokens="#t4 #t5">
  <fs>
   <f name="pos">
    <symbol value="VB"/>
   </f>
  </fs>
 </wordForm>
 <wordForm lemma="new" tokens="#t6">
  <fs>
   <f name="pos">
    <symbol value="JJ"/>
   </f>
  </fs>
 </wordForm>
 <wordForm lemma="wallpaper" tokens="#t7 #t8">
  <fs>
   <f name="pos">
    <symbol value="NN"/>
   </f>
  </fs>
 </wordForm>
</maf>
```

Figure A.2 — Stand-off encoding

Annex B
(normative)

MAF specification

B.1 Elements

B.1.1 <dcs/>

<dcs/> (data category selection) Selection of data categories used to express the annotations	
Module	MAF
Attributes	local — local name of the category **Status** Optional **Datatype** `xsd:NCName` registered — registered name of the category in the ISO Data Category Registry **Status** Optional **Datatype** `xsd:anyURI` rel — (relationship) relationship between the local meaning of a category and the registered one **Status** Optional **Datatype** `text` **Suggested values include:** eq [Default] subs gen desc — (description) informal description of a Data Category **Status** Optional **Datatype** `text`
Used by	
Contained by	**MAF:** <tagset>
May contain	Empty element
Declaration	`element dcs { attribute local { xsd:NCName }?, attribute registered { xsd:anyURI }?, attribute rel { "eq" | "subs" | "gen" | text }?, attribute desc { text }?, empty }`

B.1.2 <fsm>

<fsm> (finite state machine) Used to describe an ambiguous flow of <token> and/or <wordForm> elements	
Module	MAF
Attributes	init initial state of the FSM wrt wordForms **Status** Optional **Datatype** `xsd:Name` final final state of the FSM wrt wordForms **Status** Optional **Datatype** `xsd:Name` tinit initial state of the FSM wrt tokens **Status** Optional **Datatype** `xsd:Name` tfinal final state of the FSM wrt tokens **Status** Optional **Datatype** `xsd:Name`
Used by	
Contained by	**MAF:** <maf>
May contain	**MAF:** <transition>
Declaration	`element fsm { attribute init { xsd:Name }?, attribute final { xsd:Name }?, attribute tinit { xsd:Name }?, attribute tfinal { xsd:Name }?, transition+ }`

B.1.3 <maf>

<maf> MAF start element	
Module	MAF
Used by	
Contained by	Empty element
May contain	**MAF:** <fsm> <tagset> <token> <wfAlt> <wordForm>
Declaration	`element maf { tagset?, ( token | wordForm | wfAlt | fsm )+ }`

B.1.4 <tagset>

<tagset>	Tagset to be used to check and interpret the annotations
Module	MAF
Attributes	ref **Status** Optional **Datatype** `xsd:anyURI`
Used by	
Contained by	**MAF:** <maf>
May contain	**MAF:** <dcs/>
Declaration	`element tagset { attribute ref { xsd:anyURI }?, dcs*, fsd*, ( fvLib, fLib )* }`

B.1.5 <token>

<token>	Element used to mark tokens as defined in 3.21
Module	MAF
Attributes	att.token.information (@form, @phonetic, @transcription, @transliteration) att.token.span (@from, @to) att.token.join (@join) xml:id (identifier) provides a unique identifier for the element bearing the attribute. **Status** Optional **Datatype** `xsd:ID` **Values** any valid XML identifier.
Used by	
Contained by	**MAF:** <maf> <transition> <wordForm>
May contain	Character data only
Declaration	`element token { att.token.information.attributes, att.token.span.attributes, att.token.join.attributes, attribute xml:id { xsd:ID }?, text }`

B.1.6 <transition>

<transition> FSM transition in a flow of tokens and/or wordForms	
Module	MAF
Attributes	source source state of a transition **Status** Optional **Datatype** `xsd:Name` target target state of a transition **Status** Optional **Datatype** `xsd:Name`
Used by	
Contained by	**MAF:** <fsm>
May contain	**MAF:** <token> <wfAlt> <wordForm>
Declaration	`element transition { attribute source { xsd:Name }?, attribute target { xsd:Name }?, ( token | wordForm | wfAlt ) }`

B.1.7 <wfAlt>

<wfAlt> (WordForm alternative) Simplified form to represent an alternative between several word-forms	
Module	MAF
Used by	
Contained by	**MAF:** <maf> <transition>
May contain	**MAF:** <wordForm>
Declaration	`element wfAlt { wordForm+ }`

B.1.8 <wordForm>

<wordForm> Linguistic units built upon <tokens>	
Module	MAF
Attributes	att.wordForm.tokens (@tokens) att.wordForm.content (@tag) xml:id (identifier) provides a unique identifier for the element bearing the attribute. **Status** Optional **Datatype** `xsd:ID` **Values** any valid XML identifier. lemma lemma attached to a wordForm **Status** Optional **Datatype** string form form attached to a wordForm **Status** Optional **Datatype** `string` entry lexical entry attached to a wordForm **Status** Optional **Datatype** `xsd:anyURI`
Used by	
Contained by	**MAF:** <maf> <transition> <wfAlt> <wordForm>
May contain	**MAF:** <token> <wordForm>
Declaration	`element wordForm { att.wordForm.tokens.attributes, att.wordForm.content.attributes, attribute xml:id { xsd:ID }?, attribute lemma { string }?, attribute form { string }?, attribute entry { xsd:anyURI }?, token*, wordForm*, fs? }`

B.2 Model classes

model.common	Groups common chunk- and inter-level elements
Module	tei
Used by	
Members	model.divPart model.inter
Note	This class defines the set of chunk- and inter-level elements; it is used in many content models, including those for textual divisions.

B.3 Attribute classes

B.3.1 att.token.information

att.token.information	Attributes used to provide additional information about the content of a token
Module	MAF
Members	<token>
Attributes	form — normalised form of the token **Status** Optional **Datatype** string **Note** This attribute is provided as a facility to provide a normalized string corresponding to the surface for, for instance to facilitate search phonetic — phonetic transcription **Status** Optional **Datatype** string transcription — general transcription **Status** Optional **Datatype** string transliteration — transliteration to some other script **Status** Optional **Datatype** string

B.3.2 att.token.join

att.token.join	
Module	MAF
Members	<token>
Attributes	join Relationship with neighbouring tokens **Status** Optional **Datatype** `text` **Legal values are:** no [Default] left right both overlap

B.3.3 att.token.span

att.token.span	Attributes to denote a span in the annotated document
Module	MAF
Members	<token>
Attributes	from Left span boundary **Status** Optional **Datatype** `NMTOKEN` to Right span boundary **Status** Optional **Datatype** `NMTOKEN`

B.3.4 att.wordForm.content

att.wordForm.content	Groups descriptive attributes for a <wordForm> element
Module	MAF
Members	<wordForm>
Attributes	tag Reference to one or more morpho-syntactic descriptions expressed as feature structures **Status** Optional **Datatype** 1–∞ occurrences of `xsd:anyURI` separated by whitespace

B.3.5 att.wordForm.tokens

att.wordForm.tokens Class of attributes related to the identification of the <token> elements associated to a <wordForm> element	
Module	MAF
Members	<wordForm>
Attributes	tokens List of tokens for the corresponding <wordForm> element **Status** Optional **Datatype** 1–∞ occurrences of `xsd:anyURI` separated by whitespace

B.4 Macros

B.4.1 data.certainty

data.certainty Defines the range of attribute values expressing a degree of certainty	
Module	tei
Used by	
Declaration	`data.certainty = "high" | "medium" | "low" | "unknown"`
Note	Certainty may be expressed by one of the predefined symbolic values high, medium, or low. The value unknown should be used in cases where the encoder does not wish to assert an opinion about the matter. For more precise indication, `data.probability` may be used instead or in addition.

B.4.2 data.code

data.code Defines the range of attribute values expressing a coded value by means of a pointer to some other element which contains a definition for it	
Module	tei
Used by	
Declaration	`data.code = xsd:anyURI`
Note	It will usually be the case that the item pointed to is to be found somewhere else in the current TEI document, typically in the header, but this is not mandatory.

B.4.3 data.count

data.count	Defines the range of attribute values used for a non-negative integer value used as a count
Module	tei
Used by	
Declaration	`data.count = xsd:nonNegativeInteger`
Note	Only positive integer values (including zero) are permitted

B.4.4 data.duration.w3c

data.duration.w3c Defines the range of attribute values available for representation of a duration in time using W3C datatypes	
Module	tei
Used by	
Declaration	`data.duration.w3c = xsd:duration`
Example	`<time dur="PT45M">forty-five minutes</time>`
Example	`<date dur="P1DT12H">a day and a half</date>`
Example	`<date dur="P7D">a week</date>`
Example	`<time dur="PT0.02S">20 ms</time>`
Note	A duration is expressed as a sequence of number-letter pairs, preceded by the letter P; the letter gives the unit and may be Y (year), M (month), D (day), H (hour), M (minute), or S (second), in that order. The numbers are all unsigned integers, except for the `S` number, which may have a decimal component (using `.` as the decimal point). If any number is 0, then that number-letter pair may be omitted. If any of the H (hour), M (minute), or S (second) number-letter pairs are present, then the separator `T` must precede the first 'time' number-letter pair. For complete details, see the W3C specification.

B.4.5 data.enumerated

data.enumerated Defines the range of attribute values expressed as a single XML name taken from a list of documented possibilities	
Module	tei
Used by	
Declaration	`data.enumerated = data.name`
Note	Attributes using this datatype must contain a word which follows the rules defining a legal XML name (see http://www.w3.org/TR/REC-xml/#dt-name): for example, they cannot include whitespace or begin with digits. Typically, the list of documented possibilities will be provided (or exemplified) by a value list in the associated attribute specification, expressed with a <valList> element.

B.4.6 data.key

data.key Defines the range of attribute values expressing a coded value by means of an arbitrary identifier, typically taken from a set of externally-defined possibilities.	
Module	tei
Used by	
Declaration	`data.key = string`
Note	Information about the set of possible values for an attribute using this datatype may (but need not) be documented in the document header. Externally defined constraints, for example that values should be legal keys in an external database system, cannot usually be enforced by a TEI system. Similarly, because the key is externally defined, no constraint other than a requirement that it consist of Unicode characters is possible.

B.4.7 data.language

data.language writing system	Defines the range of attribute values used to identify a particular combination of human language and
Module	tei
Used by	
Declaration	`data.language = xsd:language`
Note	The values for this attribute are language 'tags' as defined in BCP 47. Currently BCP 47 comprises RFC 4646 and RFC 4647; over time, other IETF documents may succeed these as the best current practice. A 'language tag', per BCP 47, is assembled from a sequence of components or ***subtags*** separated by the hyphen character (-, U+002D). The tag is made of the following subtags, in the following order. Every subtag except the first is optional. If present, each occurs only once, except the fourth and fifth components (variant and extension), which are repeatable. — **language:** The IANA-registered code for the language. This is almost always the same as the ISO 639 2-letter language code if there is one. The list of available registered language subtags can be found at http://www.iana.org/assignments/language-subtag-registry. It is recommended that this code be written in lower case. — **script:** The ISO 15924 code for the script. These codes consist of 4 letters, and it is recommended they be written with an initial capital, the other three letters in lower case. The canonical list of codes is maintained by the Unicode Consortium, and is available at http://unicode.org/iso15924/iso15924-codes.html. The IETF recommends this code be omitted unless it is necessary to make a distinction you need. — **region:** Either an ISO 3166 country code or a UN M.49 region code that is registered with IANA (not all such codes are registered, e.g. UN codes for economic groupings or codes for countries for which there is already an ISO 3166 2-letter code are not registered). The former consist of 2 letters, and it is recommended they be written in upper case. The latter consist of 3 digits. — **variant:** An IANA-registered variation. These codes are used to indicate additional, well-recognized variations that define a language or its dialects that are not covered by other available subtags. — **extension:** An extension has the format of a single letter followed by a hyphen followed by additional subtags. These exist to allow for future extension to BCP 47, but as of this writing no such extensions are in use. — **private use:** An extension that uses the initial subtag of the single letter x (i.e., starts with `x-`) has no meaning except as negotiated among the parties involved. These should be used with great care, since they interfere with the interoperability that use of RFC 4646 is intended to promote. In order for a document that makes use of these subtags to be TEI conformant, a corresponding <language> element must be present in the TEI header. There are two exceptions to the above format. First, there are language tags in the IANA registry that do not match the above syntax, but are present because they have been 'grandfathered' from previous specifications. Second, an entire language tag can consist of only a private use subtag. These tags start with `x-` and do not need to follow any further rules established by the IETF and endorsed by these Guidelines. Like all language tags that make use of private use subtags, the language in question must be documented in a corresponding <language> element in the TEI header. Examples include sn — Shona zh-TW — Taiwanese zh-Hant-HK — Chinese written in traditional script as used in Hong Kong en-SL — English as spoken in Sierra Leone pl — Polish es-MX — Spanish as spoken in Mexico es-419 — Spanish as spoken in Latin America The W3C internationalization activity has published a useful introduction to BCP 47, Language tags in HTML and XML.

B.4.8 data.name

data.name	Defines the range of attribute values expressed as an XML Name.
Module	tei
Used by	data.enumerated
Declaration	`data.name = xsd:Name`
Note	Attributes using this datatype must contain a single word which follows the rules defining a legal XML name (see http://www.w3.org/TR/REC-xml/#dt-name): for example, they cannot include whitespace or begin with digits.

B.4.9 data.numeric

data.numeric	Defines the range of attribute values used for numeric values.
Module	tei
Used by	
Declaration	`data.numeric = xsd:double | token { pattern = "(\-?[\d]+/\-?[\d]+)" } | xsd:decimal`
Note	Any numeric value, represented as a decimal number, in floating point format, or as a ratio. To represent a floating point number, expressed in scientific notation, 'E notation', a variant of 'exponential notation', may be used. In this format, the value is expressed as two numbers separated by the letter E. The first number, the significand (sometimes called the mantissa) is given in decimal format, while the second is an integer. The value is obtained by multiplying the mantissa by 10 the number of times indicated by the integer. Thus the value represented in decimal notation as 1000.0 might be represented in scientific notation as 10E3.A value expressed as a ratio is represented by two integer values separated by a solidus (/) character. Thus, the value represented in decimal notation as 0.5 might be represented as a ratio by the string 1/2.

B.4.10 data.pointer

data.pointer	Defines the range of attribute values used to provide a single URI pointer to any other resource, either within the current document or elsewhere
Module	tei
Used by	
Declaration	`data.pointer = xsd:anyURI`
Note	The range of syntactically valid values is defined by RFC 3986 *Uniform Resource Identifier (URI): Generic Syntax*. Note that the values themselves are encoded using RFC 3987*Internationalized Resource Identifiers* (IRIs) mapping to URIs. For example, `https://secure.wikimedia.org/wikipedia/en/wiki/%` is encoded as `https://secure.wikimedia.org/wikipedia/en/wiki/%25` while `http://وزارة-موقع.مصر-الاتصالات/` is encoded as `http://xn--4gbrim.xn----rmckbbajlc6dj7bxne2c.xn--wgbh1c/`

B.4.11 data.probability

data.probability	Defines the range of attribute values expressing a probability
Module	tei
Used by	
Declaration	`data.probability = xsd:double { minInclusive = "0" maxInclusive = "1" }`
Note	Probability is expressed as a real number between 0 and 1; 0 representing certainly false and 1 representing certainly true.

B.4.12 data.temporal.w3c

data.temporal.w3c	Defines the range of attribute values expressing a temporal expression such as a date, a time, or a combination of them, that conform to the W3C *XML Schema Part 2: Datatypes* specification
Module	tei
Used by	

Declaration	`data.temporal.w3c =`
	` xsd:date`
	` | xsd:gYear`
	` | xsd:gMonth`
	` | xsd:gDay`
	` | xsd:gYearMonth`
	` | xsd:gMonthDay`
	` | xsd:time`
	` | xsd:dateTime`

Note	If it is likely that the value used is to be compared with another, then a time zone indicator should always be included, and only the dateTime representation should be used.

B.4.13 data.truthValue

data.truthValue	Defines the range of attribute values used to express a truth value
Module	tei
Used by	
Declaration	`data.truthValue = xsd:boolean`
Note	The possible values of this datatype are 1 or true, or 0 or false.
Note	This datatype applies only for cases where uncertainty is inappropriate; if the attribute concerned may have a value other than true or false, e.g. unknown, or inapplicable, it should have the extended version of this datatype: `data.xTruthValue`.

B.4.14 data.word

data.word	Defines the range of attribute values expressed as a single word or token
Module	tei
Used by	
Declaration	`data.word = token { pattern = "(\p{L}|\p{N}|\p{P}|\p{S})+" }`
Note	Attributes using this datatype must contain a single 'word' which contains only letters, digits, punctuation characters, or symbols: thus it cannot include whitespace.

B.4.15 data.xTruthValue

data.xTruthValue (extended truth value) Defines the range of attribute values used to express a truth value which may be unknown	
Module	tei
Used by	
Declaration	`data.xTruthValue = xsd:boolean | "unknown" | "inapplicable"`
Note	In cases where uncertainty is inappropriate, use the datatype `data.TruthValue`.

Annex C
(normative)

Morpho-syntactic data categories

This annex lists and documents the morpho-syntactic standardized data categories from which applications that are compliant with ISO 24611 can create their tagsets.

A repository of data categories, including morpho-syntactic data categories, may be found at www.isocat.org

All the data categories in this annex have been grouped together as a specific DCS within ISOCat: http://www.isocat.org/rest/dcs/568. This DCS is associated with a discussion forum where interested parties, when registered in ISOcat, can actually contribute to help improving the description and coverage of the standardized data categories associated to ISO 24611. This forum can be found under http://www.isocat.org/forum/viewforum.php?f=14, with all posts are publicly visible.

abbreviation	Designation formed by omitting words or letters from a longer form and referring to the same concept.
abessiveCase	Case that expresses the lack or absence of the referent of the noun it marks.
ablativeCase	Case used to typically indicate locative or instrumental function.
absolutiveCase	Case for nouns in ergative-absolute languages that would generally be the subjects of intransitive verbs or the objects of transitive verbs in the translational equivalents of nominative-accusative languages such as English.
accusativeCase	Case used to indicate direct object.
activeVoice	Value that expresses the situation where the grammatical subject is also the semantic actor of the verb.
adessiveCase	Case which expresses the meaning of presence ‘at’ or ‘near’ a place.
additiveCase	Case expressing “to” in Basque studies.
adjective	Part of speech related to attributes of noun.
adposition	Part of speech that occurs before/inside/after a complement composed of a noun phrase, noun, pronoun or clause that functions as a noun phrase and form a single structure with the complement to express its grammatical and semantic relation to another unit.
adverb	Part of speech to refer to a heterogeneous group of words whose most frequent function is to specify the mode of action of the verb.
affirmativeParticle	Particle used to express affirmation.
affix	Letter or group of letters which are added to a word to make a new word.
affixedPersonalPronoun	Personal pronoun that is affixed.
affixRank	Rank of an affix
allativeCase	Case which expresses the meaning of motion ‘to’ or ‘towards’ the referent it marks.
allomorph	One of two or more complementary morphs which manifest a morpheme in its different phonological or morphological environments.
allusivePronoun	Allusive pronoun.