ISO TC 37/SC 4 N119 Rev. 3

Date: 2005-08-22

ISO/CD 24611

Committee identification: ISO/TC 37/SC 4/WG 2

Secretariat: KATS

Language Resource Management – Morpho-syntactic Annotation Framework (MAF)

Warning

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

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

Document type: International Standard
Document subtype: Not applicable
Document language: en

Copyright notice

This ISO document is a draft international standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to ISO at the address below or ISO's member body in the country of the requester:

Copyright Manager
ISO Central Secretariat
1 rue de Varembé
1211 Geneva 20 Switzerland
tel. +41 22 749 0111
fax +41 22 734 0179
e-mail central@iso.ch

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

1 Scope **7**

2 Normative references **7**

3 Terms and definitions **7**

4 Key standards used by MAF **12**
4.1 ISO 12620 Data Category Registry (DCR) . . . 12
4.2 ISO 24610 Feature Structures (FSR and FSD) . . . 12
4.3 OLAC Metadata . . . 12
4.4 Unified Modeling Language (UML) . . . 12

5 General characteristics of MAF **13**
5.1 Overview . . . 13
5.2 MAF Meta-Model . . . 13

6 Segmenting with tokens **13**
6.1 Standoff notation . . . 14
6.2 Embedding notation . . . 16
6.3 Informative attributes . . . 16
6.4 Completing the embedding token notation . . . 17
6.4.1 Joining tokens . . . 17
6.4.2 Overlapping tokens . . . 17
6.5 Formal description: `token` . . . 18

7 Word Forms as linguistic units **18**
7.1 Token attachment . . . 19
7.1.1 One token; one word form . . . 19
7.1.2 Several contiguous tokens; one word form . . . 19
7.1.3 Several discontinuous tokens; one word form . . . 19
7.1.4 Zero token; one word form . . . 20
7.1.5 One token; several word forms . . . 20
7.2 Referring lexicon entries . . . 21
7.3 Compound word forms . . . 22
7.4 Formal description: `wordForm` . . . 22

8 Morpho-syntactic content **22**
8.1 Using feature structures . . . 23
8.2 Compact morpho-syntactic tags . . . 23
8.2.1 FSR libraries . . . 23
8.3 Designing tagsets . . . 25
8.4 Formal description: `tagset` . . . 26

9 Handling ambiguities **26**
9.1 Word form Content Ambiguities . . . 26
9.2 Lexical Ambiguities . . . 27
9.3 Structural Ambiguities . . . 27
9.3.1 Structural ambiguities over word forms . . . 27
9.3.2 Structural ambiguities over tokens . . . 28
9.4 Simplified structuring variants . . . 29
9.4.1 Non ambiguous linear representation . . . 29
9.4.2 Mixed linear and lattice representation . . . 29
9.5 Expanding the simplified variants . . . 30
9.5.1 Separating tokens and word forms . . . 30
9.5.2 Wrapping into local lattices . . . 30
9.5.3 Merging local lattices . . . 31
9.5.4 Removing `wfAlt` . . . 32
9.6 Formal description: `wfAlt` and `fsm` . . . 33

10 Header and metadata **33**
10.1 Formal description . . . 34

A (informative) RELAX NG compact schema **35**
A.1 Validating MAF documents . . . 37

B (informative) DTD **39**

C (informative) Illustrative examples **41**
C.1 Tagsets . . . 41
C.2 Demonstrator . . . 41

D (illustrative) Morpho-syntactic Data Categories **43**

E (informative) UML notions used within MAF **45**
E.1 Introduction . . . 45
E.2 The notion of class . . . 45
E.3 The notion of attribute . . . 45
E.4 The notion of relationship . . . 45
E.5 The notion of association . . . 45
E.6 The notion of aggregation . . . 45
E.7 The notion of generalization . . . 46
E.8 The notion of instance . . . 46
E.9 The notion of package . . . 46
E.10 Graphical notations . . . 46

Foreword

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

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

International Standard 24611 was prepared by Technical Committee ISO/TC 37, Terminology and Other Language Resources, Subcommittee SC 4, Language Resource Management.

All the Annexes are for information only.

Introduction

This standard provides a reference format for the representation of morpho-syntactic annotations.

 ©ISO 2005 – All rights reserved

Language Resource Management – Morpho-syntactic Annotation Framework (MAF)

1 Scope

In Natural Language Resource Management, the morpho-syntactic annotation phase assigns to each document segment (either text or speech) one or more *tags* providing *morpho-syntactic* information about the *part of speech* (noun, adjective, verb, ...), *morphological* and *grammatical* features (such as number, gender, person, mood, verbal tense, ...) and possibly other specific linguistic properties. The morpho-syntactic annotations attached to a segment do not refer to other segments or annotations, even if the choice of an annotation may depend on the surrounding context.

2 Normative references

- ISO 8879: 1986 (SGML) as extended by TC2 (ISO/IEC JTC 1/SC 34 N029: 1998-12-06) to allow for XML
- ISO 19757-2, Document Schema Definition Language, part 2, to allow for RELAX NG specifications. RELAX NG is a schema language for XML, standing for REgular Language for XML for Next Generation, and simplifies and extends the features of DTDs, Document Type Definitions.
- ISO 12620 on Data Category Registry (DCR)
- ISO 24610-1 on Feature Structure Representation (FSR)
- ISO 24610-2 on Feature System Declaration (FSD)
- ISO 24612 on Linguistic Annotation Framework (LAF)
- Text Encoding Initiative (TEI) – Chapters to be defined

3 Terms and definitions

For the purpose of ISO 24611, the following terms and definitions apply.

3.1
associative relation
relation by which a linguistic unit is associated with other units. It is a virtual association which does not requires their effective presence and differs from a paradigmatic relation in that the latter only refers to linguistic units associated by substitutability.

3.2
closed data category

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

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

3.3
conceptual domain
finite list of **simple data categories** that may be the values of a complex **data category**

3.4
data category
result of the specification of a given data field or the content of a closed data field

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

3.5
directed acyclic graph
DAG
graph with directed edges and no cycle

3.6
discourse

3.7
feature specification
the assignment of a value to a feature. In MAF, a feature shall denote a **morpho-syntactic feature** of a linguistic unit, such as the mood or tense of a verb.

3.8
feature structure
a set of **feature specifications**, used in MAF to express morpho-syntactic content.

3.9
finite state automata
FSA

finite set of transitions from state to state, with an initial state and a final one
See also **DAG**.

3.10
form
any sequence of letters, pictograms and numerals used to write or pronounce a word

3.11
grammatical category
See also **part of speech**

3.12
inflection
modification or marking of a **s**o that it reflects grammatical (i.e. relational) information, such as grammatical gender, tense, person, etc.

3.13
inflection paradigm
a table illustrating the forms of an **inflected word**

3.14
inflected form
form that a word can take when used in a sentence or a phrase

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

3.15
lattice
term often used in the NLP community to denote (with some slight confusion with the notion of *algebraic lattice*), an directed acyclic graph with an initial node and a final node.
See also **DAG**
See also **FSA**

3.16
lemma – lemmatized form
class of **inflected forms** differing only by inflectional morphology. A lemma is usually referred to by one of these forms, arbitrarily chosen (e.g., infinitive for French verbs).

3.17
lexeme
lexical **morpheme** to be distinguished from a grammatical morpheme by the fact that it belongs to an open list and that it bears an autonomous signification.

3.18
lexicon
resource comprising a collection of **inflected forms** or **lemmas** for a given language

3.19
morpheme
smallest linguistic unit bearing a signification in a discourse and that cannot be divided into smaller meaningful units. A morpheme is either grammatical (grammeme) or lexical (**lexeme**).

3.20
morphological feature

morpho-syntactic feature
category induced from the **inflected form** of a word

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

3.21
morphology of a word
morpho-syntax of a word
description comprising the **lemmatized form** or forms of a word, plus additional information on its `/part of speech/` **data categories**, possibly its inflectional paradigm or paradigms, and possibly its explicitly listed inflected forms.

NOTE: The term morpho-syntax is often used in place of morphology as it describes such features as number, gender, case etc. which are essential for syntactic agreement.

3.22
multi-word expression
MWE
an expression composed of an ordered group of words that has properties that are not predictable from the properties of the individual words or of their normal mode of combination.

NOTE: The group of words making up an MWE can be continuous or discontinuous.

EXAMPLE: "father in law" or "to be over the moon" that mean something different from what they appear to mean.

3.23
natural language processing
NLP
the field of study covering knowledge and techniques which allow computerized processing of linguistic data. This field combines a variety of skills including linguistics, mathematical logic, statistics, and algorithms.

3.24
open data category
data category whose content cannot be fully enumerated due to the organic nature of language

EXAMPLE: Typical open data categories might include `/term/`, `/lemma/`.

3.25
syntagmatic relation
relation by which linguistic units in a discourse are associated.

3.26
morpho-syntactic tag
to an associative relation corresponds a feature, for which the related entities share the same

value. The morpho-syntactic tag lists some of these features (part-of-speech, grammatical category, etc.).

3.27
part of speech
category assigned to a word based on its grammatical and semantic properties
See also **grammatical category**

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

3.28
token
non-empty contiguous discourse sequence identified as such by a morpho-phonological analysis or an automatic processing of the discourse.
This can involve the recognition of a regular or algebraic language (matching of the separators), or a lexicological analysis (recognition of roots, morphological derivation and inflection, etc.).

3.29
tokenization
the process identifying **tokens**

3.30
word-form
morpho-syntactic unit
contiguous or non-contiguous entity from a speech or text sequence identified as such in an **associative relation**. This identification is the basis of morpho-syntactic tagging (**part-of-speech**, grammatical category, agreement feature, etc.). Morpho-syntactic units may have no acoustic or graphic realization, or correspond to one or more **tokens**.

3.31
romanization
transliteration from a non-Latin script into a Latin script

3.32
script
set of graphic characters used for the written form of one or more languages (ISO/IEC 10646-1, 4.14)

3.33
simple data category
data category that may be the possible content of a **closed data category**, but that cannot itself be further sub-divided

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

3.34
transcription
form resulting from a coherent method of writing down speech sounds

3.35
transliteration
form resulting from the conversion of one writing system into another

3.36
word
in the context of a given language, is a description composed of at least a **part of speech** and a **lemmatized form**

NOTE: The description can include more morphological information and/or syntactic and semantic information. A word is either a single word or a **multi-word expression**.

3.37
word class
See also **part of speech**

4 Key standards used by MAF

4.1 ISO 12620 Data Category Registry (DCR)

The designers of any specific MAF tagset shall use data categories from the ISO 12620 DCR. The DCR is a set of data category specifications defined by ISO 12620 and maintained as a global resource by ISO TC 37 in compliance with ISO/IEC 11179-3:2003. Tagset creators can define a set of new data categories to cover data category concepts that are needed and that are not currently available in the DCR. The tagset creators shall be responsible for negotiating the addition of the new data categories to the DCR. This supplemental set of data categories shall be represented and managed in conformance with ISO 12620.

4.2 ISO 24610 Feature Structures (FSR and FSD)

Morpho-syntactic content shall be expressed using the ISO 24610 part 1 document on Feature Structure Representation and validated using the future ISO 24610 part 2 companion document on Feature System Declaration.

4.3 OLAC Metadata

Metadata for MAF shall be expressed following the recommendations and categories proposed by the *Open Language Archives Community* (OLAC), as described in the latest version of *OLAC Metadata Standard* (`http://www.language-archives.org/OLAC/metadata.html`). The OLAC Metadata Set includes the Dublin Core Metadata Set with qualifiers.

4.4 Unified Modeling Language (UML)

MAF complies with the specifications and modeling principles of UML as defined by OMG. MAF uses the subset of UML that is described in Annex E.

5 General characteristics of MAF

5.1 Overview

In the Linguistic Community, morpho-syntactic annotations provide an important layer of linguistic information in a document, even if they do not cover the full range of possible linguistic annotations. Other kinds of annotation on references, discourse, prosody, or parsing may complete morpho-syntactic annotations.

Syntax and semantics can not be avoided in the definition of parts of speech and of grammatical categories. For instance, pronouns and substantives intrinsically carry a reference to some entity; the tense or the aspect of verbs indicate the temporal deixis; the person, modality and other grammatical categories indicate the enunciation context, ... Therefore, it is not easy to provide an exact and precise definition of what morpho-syntactic annotations cover because they are strongly related to many other linguistic properties of a given language in a given context.

Nevertheless, the present proposal tries to delimit minimal and maximal sequences in documents (either text or speech) that can be identified as *morpho-syntactic units* and tries to categorize the linguistic properties that may be used to mark these units, within some larger syntagmatic context. Minimal units can not be broken into sub-parts that could be identified by similar morpho-syntactic criteria, but may however still be broken into smaller units with morphological or phonological properties. Morpho-syntactic units can be nested to form maximal units (such as compound words) that act as elementary units for other level of linguistic analysis, particularly parsing. The exact boundary between morpho-syntax and parsing is sometimes difficult to define.

5.2 MAF Meta-Model

Figure 1 presents a simplified view of the proposed meta-model for morpho-syntactic annotations, while Figure 2 presents a more formal view based on UML. An annotated document is formed by a raw original document and a set of annotations. The annotations are carried by *word forms* covering zero, one or more segments or *tokens* of the original document. A word form may reference a lexicon entry and provides information about its underlying lemma and inflected form. The morpho-syntactic content attached to a word form is expressed by *feature structures* following the guidelines of one or more *tagsets*. The terminology or set of *categories* used in tagsets are described w.r.t. ***registered data categories***. Because of structural ambiguities, both tokens and word forms are organized into one or more flows, materialized by ***lattices***, or more formally by ***Directed Acyclic Graphs*** [DAGs]. The current proposal addresses the representation of segments (through tokens), word forms, morpho-syntactic content, tagsets, and ambiguity. A MAF model is instantiated from the MAF meta-model through the selection of a set of data categories.

6 Segmenting with tokens

Morpho-syntactic annotations are carried by segments, called *tokens*, 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 morpho-syntactic units from their realizations. Some parts of a document may

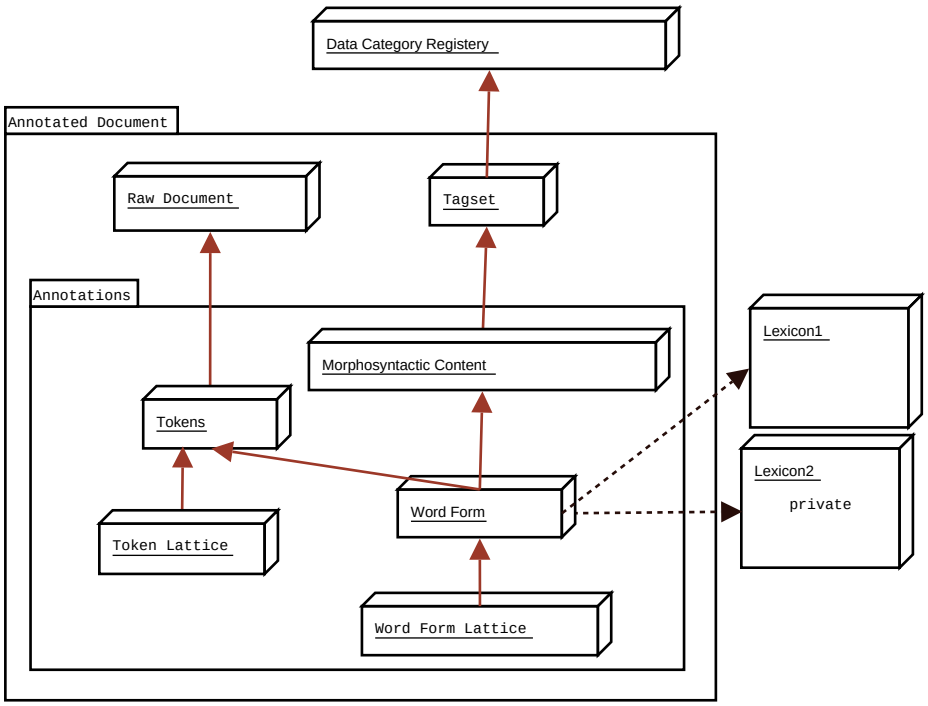

Figure 1: Simplified view of MAF meta-model

carry no annotations (typographic marks, didascalies, markup elements, ...); other parts may not exactly correspond to their segmented form (abbreviations, brachygraphies, typographic errors and variations, typographic and morphological contractions, ...). Also, a morpho-syntactic unit may not correspond to a segment identified by typographic marks (such as white spaces or hyphens), for instance for German compound words, speech transcription, or Sanskrit writing.

The element `token` is used to represent these segments of the original document that, roughly speaking, follow typographical, morphological, or phonological boundaries. The current proposal does not define the linguistic properties of tokens. In different languages, a token may be identified through typographic properties (white-space, hyphens, characters, ...) and/or morphological properties (radical, affix, morpheme, ...). The description of the morphological, phonological or lexicological structures that may define a token is not covered by the current proposal.

Other typographical marks used to format pages or to separate words and paragraphs, as well as encoding information, do not belong to morpho-syntactic annotations and are also not covered by this proposal, but rather by TEI.

6.1 Standoff notation

The element `token` provides an independence from the original document by providing a way to reference intervals in documents. The attributes `from` and `to` are used to define such intervals. The content of these attributes depends on some chosen addressing schema to denote non ambiguous document positions and depends on the nature of the original document.

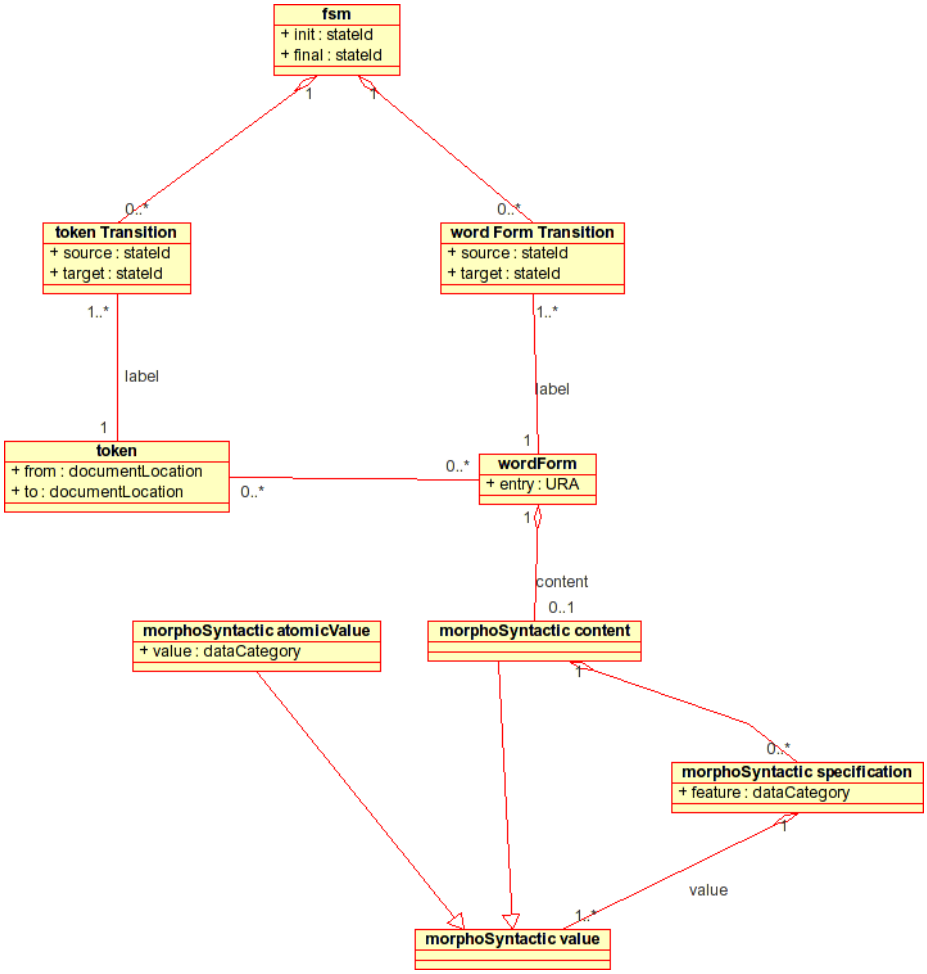

Figure 2: UML view of MAF meta-model

 ©ISO 2005 – All rights reserved

6.2 Embedding notation

It is not always necessary to separate the original document from its annotations. For simple cases, textual content may be directly embedded within `token`.

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

The embedding notation will be used for most of the provided examples for MAF but it should be noted that the use of this notation is not recommended. A first reason is that the morpho-syntactic annotations may conflict with other annotations. A second reason is that the content of the textual material separating the textual content embedded within `token` is not precisely defined (white-space, newlines, no space, hyphen, ...), except by relying on attribute `join`.

6.3 Informative attributes

Tokens address segments of the original document but also provide a level of possible abstraction w.r.t. this document, for instance w.r.t. graphical or phonological variations that are not linguistically pertinent. The non mandatory attributes `form`, `transcription`, `transliteration` may be used to perform this abstraction, providing, for instance, the phonetic transcription of a speech segment, the roman transliteration of some Cyrillic word, the expansion of an abbreviation, the correction of a typographical error, or the choice of a normalized form in presence of variations:

```
<token form="et cetera" id="t1">etc.</token>
<token form="tzar" id="t2">csar</token>
<token form="tzar" id="t3">tsar</token>
<token form="23/02/03" id="t4">February, 23rd 2003</token>
<token form="et cetera" phonetic="/etsettrə/" from="1251" to="1253" id
    ="t5"/>
<token phonetic="/platto/" id="t6">plateau</token>
```

The abstraction provided by the attribute `form` is also adequate to handle the phenomena of contraction and agglutination where two tokens may cover the same segment of the original document for distinct values (see Section 6.4.2).

6.4 Completing the embedding token notation

As above mentioned, the embedding token notation is less precise than the standoff one, in particular to explicit the contiguity and the overlapping of tokens (which are obvious to check using the document positions in the case of the standoff notation).

6.4.1 Joining tokens

The embedding notation for tokens is completed by the attribute `join` used to specify how a token is joined with its sibling tokens. By default, two sibling tokens are considered to be separated by whatever separator is standard for the document language (for instance, space separated for many languages). By using the attribute `join`, it is possible to indicate that a token is contiguous with its left or right sibling or with both.

```
<!-- it is said ... -->
<token id="t1">L'</token>
<token id="t2" join="left">on</token>
<token id="t3">dit</token>
```

It should be noted that a token may enclose material usually considered as separator, such as spaces, newline, dash, apostrophe, ..., even if these tokens do not anchor linguistic units at the level of word forms.

```
<!-- it is said ... -->
<token id="t1">L</token>
<token id="t2" join="both">'</token>
<token id="t3">on</token>
<token id="t4">dit</token>
```

Another example, in Modern Greek, is provided by the idiomatic expression "*καλοκαγαθὸς*" (*good and brave*) that may be segmented in three agglutinated segments "*καλὸς*", "*και*", and "*αγαθὸς*" and represented by:

```
<token form="καλὸς" id="t0">καλο</token>
<token form="και" id="t1" join="left">κ</token>
<token form="αγαθὸς" id="t2" join="left">αγαθὸς</token>
```

6.4.2 Overlapping tokens

Two tokens may overlap, for instance to denote an agglutinated or contracted form (for instance, in French, "*des*" may be seen as a contraction for "*de les*" [of the]), or to denote multi-locutor documents with overlapping discourses. In these cases, a `token` may not mark just the realization of a typographical or vocal sequence, but expresses a deeper linguistic reality pertinent for segmenting a document. It is however still possible not to mention overlapping at the level of tokens and to postpone the issue at the level of linguistic units , i.e. word forms.

The value `overlap` for the token attribute `join` may be used to denote overlapping at the level of embedding tokens. For instance, the following example illustrates the contraction of an abbreviation with a punctuation mark for "*etc.*", for the standoff and embedding notations for element `token`:

a Standoff notation

```
<token form="et cetera" id="t1" from="p1" to="p3"/>
<token form="#dot#" id="t2" from="p1" to="p3"/>
```

b Embedding notation

```
<token form="et cetera" id="t1">etc.</token>
<token form="#dot#" id="t2" join="overlap"/>
```

6.5 Formal description: `token`

```
\token =
  element token {
    attribute id { xsd:ID }?,
    token.information,
    (
      (
       attribute from { DocumentLocation },
       attribute to { DocumentLocation }
      )
     | ## DTD => ,
      (
      [ a:defaultValue = "no" ]
      attribute join { "no" | "left" | "right" | "both" | "overlap" }
        ?,
      text
      )
    )
  }
token.information &= attribute form { string }?
token.information &= attribute phonetic { string }?
token.information &= attribute transcription { string }?
token.information &= attribute transliteration { string }?
```

7 Word Forms as linguistic units

The segments identified by `token` elements are used to anchor *word forms*, that may generally be associated, through attribute `entry`, to a lexical entry in a lexicon. Words forms are also characterized by a part of speech as well as morphological and grammatical properties expressed by feature structures (see Section 8.1). Immediate information about the lemma and inflected forms may also be attached with the attributes `lemma` and `form`. In particular, the attribute `form` is useful when the inflected form attached to the word form does not coincide with the content attached to the covered tokens, because, for instance, of spelling corrections.

A token may be associated to more than one word form and, conversely, a word form may cover more than one token.

For instance, in French, the morphological agglutination of *auquel* ("of which") may have several representations, depending on the *granularity* of the tokenization:

coarse granularity The character sequence *auquel* is not decomposed and is covered by a single `token`, with two word forms covering this segment.

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

fine granularity The tokenizer identifies two agglutinated parts materialized by two tokens, each of them anchoring a word form:

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

The choice of a level of granularity can be motivated by the usage or by the available tools for a given language.

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

The current proposal does not discuss the linguistic choices that define these linguistic units but 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 words).

7.1 Token attachment

7.1.1 One token; one word form

The simplest case of relationship between tokens and word forms is when a word form covers a single token.

```
<token id="t0">apple</token>
<wordForm lemma="apple" tokens="t0"/>
```

7.1.2 Several contiguous tokens; one word form

However, the current proposal allows the handling of more complex cases, as the identification of compound words covering several adjacent tokens:

```
<token id="t0">prime</token>
<token id="t1">minister</token>
<wordForm lemma="prime_minister" tokens="t0 t1"/>
```

7.1.3 Several discontinuous tokens; one word form

A sequence of non contiguous tokens may also be attached to a word form, for instance to handle cases where some material is inserted inside the components of a word form:

 ©ISO 2005 – All rights reserved

```
<token id="t1">afin</token>
<token id="t2">justement</token>
<token id="t3">de</token>
<wordForm lemma="afin_de" tokens="t1 t3"/>
<wordForm lemma="justement" tokens="t2"/>
```

This kind of phenomena may also occur for verbs with detached particles, for instance in English or German. The English infinitive verbal form “to <verb>” may also fit in this scheme.

```
<token id="t1">to</token>
<token id="t2">eventually</token>
<token id="t3">decide</token>
<wordForm lemma="to_decide" tokens="t1 t3"/>
<wordForm lemma="eventually" tokens="t2"/>
```

In order to identify discontinuous word-form while preserving some information about the position of each component in the flow of word forms, one may use word forms covering the same sequence tokens and referring to the same entry (but possibly sub-entries).

```
<token id="t1">to</token>
<token id="t2">eventually</token>
<token id="t3">decide</token>
<wordForm entry="urn:lexicon:en:decide:to" tokens="t1 t3"/>
<wordForm entry="urn:lexicon:en:eventually" tokens="t2"/>
<wordForm entry="urn:lexicon:en:decide:main" tokens="t1 t3"/>
```

7.1.4 Zero token; one word form

Another case that may arise is when one wishes to insert a word form which is not realized in the original document, and is, therefore, associated with an empty sequence of tokens, e.g., some pronouns in Spanish or the hypothesis of traces.

```
<token id="t1">Jean</token>
<token id="t2">propose</token>
<token id="t3">de</token>
<token id="t4">partir</token>
<wordForm lemma="Jean" tokens="t1"/>
<wordForm lemma="proposer" tokens="t2"/>
<wordForm lemma="de" tokens="t3"/>
<wordForm lemma="PRO" tokens=""/>
<wordForm lemma="partir" tokens="t4"/>
```

Even if a word form covers no tokens, it still has a relative position w.r.t. the other word forms. It is this relative position which is pertinent for further processing, rather than some absolute document position.

7.1.5 One token; several word forms

Finally, several word forms may be attached to a same token, as illustrated by the following examples.

```
<!-- Give it to me -->
<token form="damelo" id="t1">Damelo</token>
<wordForm lemma="da" tokens="t1"/> <!-- (Donne) -->
<wordForm lemma="me" tokens="t1"/> <!-- (le) -->
<wordForm lemma="lo" tokens="t1"/> <!-- (moi) -->
```

```
<!-- of which -->
<token id="t0">auquel</token>
<wordForm lemma="à" tag="pos.prep" tokens="t0"/>
<wordForm lemma="lequel" tag="pos.pronrel" tokens="t0"/>
```

7.2 Referring lexicon entries

A word form is a linguistic unit carrying morpho-syntactic properties. Generally, a linguistic unit may be characterized by a label corresponding to an entry if some lexicon. This identification is materialized by the attribute `entry`, whose content should express a reference (an URN) to the lexicon entry.

```
<token id="t1">Prime</token>
<token id="t2">minister</token>
<wordForm entry="urn:lexicon:en:prime_minister" tokens="t1 t2"/>
```

The notion of "*lexicon entry*" is outside the scope of MAF. A reference to a lexicon entry is therefore not precisely defined but, in first approximation, should correspond to an URN (*Uniform Resource Name*). It should be noted that one may wish to reference lexicons "sub-entries" for polysemous entries or for compound forms.

```
<token id="t1">to</token>
<token id="t2">eventually</token>
<token id="t3">decide</token>
<wordForm entry="urn:lexicon:en:decide:to" tokens="t1 t3"/>
<wordForm entry="urn:lexicon:en:eventually" tokens="t2"/>
<wordForm entry="urn:lexicon:en:decide:main" tokens="t1 t3"/>
```

A token or a sequence of tokens may sometimes be identified as forming a word form because of various properties but can not associated to some lexicon entry, either because no lexicon is available or because the word form corresponds to a named entity (a proper name, a date, an address, ...) or to a neologism. In that case, the content of attribute `entry` may be left empty. The other informative attributes `lemma` and `form` may still be used to provide more information about the word form.

```
<token id="t0">October</token>
<token id="t1">,</token>
<token id="t2">23rd</token>
<token id="t3">2005</token>
<wordForm lemma="DATE" form="2005/10/23" tokens="t0 t1 t2 t3"/>
```

For such unknown words, it is however suggested that they can be collected into a document specific lexicon, in order for the unknown words to refer entries in this lexicon.

7.3 Compound word forms

The structure of compound forms (including multi-word expressions) may be expressed using nested word forms, therefore providing information about the subparts even when none is available for the whole, for instance for neologisms:

```
<!-- birthday gift wrapping paper -->
<token form="Geburtstag" id="t1" join="right">Geburtstags</token>
<token form="Geschenk" id="t2" join="right">geschenk</token>
<token form="Papier" id="t3">papier</token>
<wordForm tokens="t1 t2 t3">
   <wordForm entry="urn:lexicon:de:geburstag" lemma="geburstag"
      tokens="t1"/>
   <wordForm entry="urn:lexicon:de:geschenk" lemma="geschenk" tokens
      ="t2"/>
   <wordForm entry="urn:lexicon:de:papier" lemma="papier" tokens="t3
      "/>
</wordForm>
```

Note: Precising the derivational morphology of a compound word is outside the scope of MAF. Still, the addition of a `deriv` attribute on embedded word forms is being investigated, for instance to mention the head of a compound form.

7.4 Formal description: `wordForm`

```
wordForm =
   element wordForm {
      wordForm.identification ,
      wordForm.tokens ,
      wordForm* ,
      wordForm.content ?
   }
wordForm.tokens =
      ( attribute tokens  { xsd:IDREFS }
      | ## DTD => ,
       \token*
      )
wordForm.identification &= attribute entry { xsd:anyURI } ?
wordForm.identification &= attribute lemma { string } ?
wordForm.identification &= attribute form { string } ?
```

8 Morpho-syntactic content

This section explains how to attach morpho-syntactic content to word forms and how to define reusable *tagsets* to provide compact notations through *tags* and to control the validity of these contents.

The previous section explains how to enrich a document with morpho-syntactic annotations. However, it does not define the content of these annotations. What set of features and feature values should we use to express this content (within element `wordForm`) and with which meaning ?

Such a set is usually referred as a *tagset* specifying the content of possible annotations. However, the diversity of approaches and languages makes almost impossible the proposition of an unique tagset. More modestly or pragmatically, the current proposal seeks to provide mechanisms to define tagsets by relying on a Data Category Registry (DCR) and Feature Structures Representations (FSR).

An annotated document will therefore be completed by either adding or referring to a *tagset*.

8.1 Using feature structures

A word form may be completed by a morpho-syntactic content defining its linguistic nature and its grammatical function in its current context. This content is expressed using *Feature Structures*, following the recommendation of ISO 24610 Part 1 document on "*Feature Structure Representation*" [FSR]. In first approximation, a feature structure may attach one or several (possibly complex) values to linguistic properties (i.e., noun to part of speech, present to tense, indicative to mood, ...).

```
<!-- nice -->
<token  id="t0">belle</token>
<wordForm entry="urn:lexicon:fr:beau" lemma="beau" tokens="t0">
  <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>
```

The feature structure content attached to a word form may also provides additional information of interest about a word form.

8.2 Compact morpho-syntactic tags

FSR proposal provides ways for the compact representation of feature structures, by relying on *libraries* naming feature values and **feature specifications** (a feature specification being a pair formed by a feature and a value). These names may be used in `wordForm` attribute `tag` to get compact tags, following a standard practice in the NLP community.

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

The content of attribute `tag` should be similar to the content of attribute `feats` defined in FSR, namely a space-separated sequence of **feature specification** identifiers.

The libraries naming recurrent values and **feature specifications** are part of the tagset(s) coming with the annotated document.

8.2.1 FSR libraries

The generic way provided by FSR to use libraries is illustrated by the following example, with the attribute `feats` of element `fs`:

```
<!-- A feature value library -->
<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>
<!-- A feature specification library -->
<fLib>
    <f xml:id="pos.n" name="pos" fVal="noun"/>
    <f xml:id="num.s" 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>
```

With such a library, following FSR rules, one may write:

```
<wordForm lemma="prime_minister" tokens="t1">
      <fs feats ="pos.n num.s gen.f"/>
</wordForm>
```

or, equivalently, by using attribute `tag`, one may write:

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

Disjunctive values are allowed by FSR and may also be simplified, following the same mechanism:

```
<!-- A feature value library -->
<tagset>
  <fvLib>
      <vAlt xml:id="first.third">
            <symbol value="first"/>
            <symbol value="third"/>
      </vAlt>
      <symbol xml:id="verb" value="verb"/>
      <symbol xml:id="sing" value="singular"/>
  </fvLib>
  <!-- A feature specification library -->
  <fLib>
    <f xml:id="pers.13" name="pers" fVal="first.third">
    </f>
    <f xml:id="pos.v" name="pos" fVal="verb"/>
    <f xml:id="num.s" name="number" fVal="sing"/>
  </fLib>
</tagset>
<!-- Annotated document -->
<token  id="t0">porte</token>
<wordForm tokens="t0"
          entry="urn:lexicon:fr:porter"
```

```
            tag="pos.v pers.13 num.s"/>
```

8.3 Designing tagsets

The features, values, and possibly feature types used to specify morpho-syntactic content are not just labels but carry linguistic meanings, or, in other words, semantic content. To avoid misinterpretations, the semantic content attached to a feature, a value or a 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 feature `/tense/` for nouns in most languages.

MAF does not try to define the semantic content of an unique complete set of such features, values, and types. It would be an almost impossible task given the diversity of languages, and it would be equally impossible to assign to each component a meaning agreed on by the whole community.

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

The first objective of a tagset is to list the terminology used to annotate a document as a set of **data categories** whose meanings is precisely defined in a *Data Category Registry*, following the recommendation of ISO 12620 proposal on "*Data Category Registry*". The process may be seen as selecting a subset of morpho-syntactic data categories (*Data Category Selection – DCS*).

```
<tagset>
  <dcs local="genre" registered="dcs:morphosyntax:gender:fr" rel="eq"/>
  <dcs local="fem" registered="dcs:morphosyntax:gender:fr:feminine" rel=
      "eq"/>
</tagset>
```

The correspondence with a registered data category may not be perfect. The `rel` may be used to specify which relationship exists between the local and registered data categories. For instance, one may introduce a local data category `/advneg/` as being subsumed by a more general registered data category `/adverb/`.

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

It is also possible (but not advised) to introduce a local data category bearing no relationship with any registered data category.

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

When the correspondence is not perfect or missing, a few words of description should be added to define the meaning of a local data category.

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

The second objective of a tagset is to specify the set of valid feature structures based on the selected data categories. It will be achieved by relying on the proposed ISO 24610 Part 2 on "*Feature System Declaration*" [FSD].

The third objective of a tagset is to name the most common morpho-syntactic structures through the use of FSR libraries, as seen in Section 8.2.1.

8.4 Formal description: `tagset`

```
wordForm.content =
   ( attribute tag { xsd:IDREFS }
   | ## DTD => ,
       fs
   )
fs |= notAllowed            # defined in iso-fs-standalone.rnc
tagset =
   element tagset {
      ( attribute ref { xsd:anyURI }
      |  ## DTD => ,
         (dcs* & fsd* & tagset.lib*)
      )
   }
dcs =
   element dcs {
      attribute local { xsd:NCName },
      ( attribute registered { xsd:anyURI },
         attribute rel { "eq" | "subs" | "gen" } )?,
      element description { text }*
   }
fsd |= notAllowed              # defined in future iso-fsd.rnc
tagset.lib |= fvLib
tagset.lib |= fLib
fLib |= notAllowed             # defined in iso-fs-standalone.rnc
fvLib |= notAllowed            # defined in iso-fs-standalone.rnc
```

The `dcs` corresponds to a Data Category Selection part whose exact content is still to be defined.

The `fsd` corresponds to a Feature Structure Declaration part whose normalization is yet to be done.

9 Handling ambiguities

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

9.1 Word form Content Ambiguities

The proposal on Feature Structure Representation provides several ways to represent ambiguities, for instance at the level of feature values. These mechanisms may be used to handle the ambiguities occurring within the morpho-syntactic content of a word-form.

For instance, the French inflected verb form “mange” (to eat) is ambiguous between the 1st and 3rd persons, and this ambiguity can be captured by the **vAlt** element present in FSR:

```
<token  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>
```

A compact tag notation can still be used by registering most frequent cases of ambiguities in FSR libraries (Section 8.2.1).

```
<token id="t0">mange</token>
<wordForm tokens="t0"
          entry="urn:lexicon:fr:manger"
          tag="pos.v aux.avoir mood.i tense.p pers.13 num.s"/>
```

9.2 Lexical Ambiguities

Ambiguities between different lexical entries for a same sequence of tokens can be handled by the element **wfAlt**:

```
<token id="t0">porte</token>
<wfAlt>
  <wordForm tokens="t0" entry="lexicon:porte"  tag="pos.n ..."/>
  <wordForm tokens="t0" entry="lexicon:porter" tag="pos.v ..."/>
</wfAlt>
```

9.3 Structural Ambiguities

9.3.1 Structural ambiguities over word forms

A general and very generic answer is to describe the possible readings as paths through an Directed Acyclic Graph (DAG) whose edges are labeled by a word form. Such DAGs forms a sub-part of Finite State Automata and also cover the notion of *word lattice* used in parsing and speech recognition communities. They are powerful enough to represent ambiguities between several decompositions into compound forms. They can also be used to denote simpler cases of lexical ambiguities.

For instance, the French textual sequence "*fer à cheval*" (horse shoe) can still be decomposed into several readings ("[horse shoe]", "[iron] [on horse]", "[iron] [of] [horse]"), giving the following DAG:

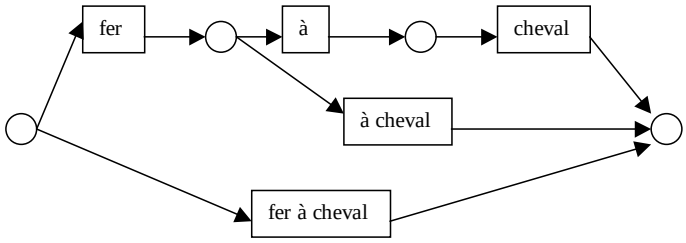

Figure 3: DAG for "fer à cheval"

```
<token id="t1">fer</token>
<token id="t2">à</token>
<token id="t3">cheval</token>
<fsm init="S0" final="S3">
   <transition source="S0" target ="S3">
      <wordForm  tokens="t1 t2 t3"
                 entry="urn:lex:fr:fer_%E0_cheval"
                 lemma="fer_à_cheval"/>
   </transition>
   <transition source="S0" target ="S1">
      <wordForm entry="urn:lex:fr:fer" tokens="t1"/>
   </transition >
   <transition source="S1" target ="S2">
      <wordForm tokens="t2"
                entry="urn:lex:fr:%E0" lemma="à"/>
   </transition>
   <transition source="S2" target ="S3">
      <wordForm  tokens="t3" entry="urn:lex:fr:cheval"/>
   </transition>
   <transition source="S1" target ="S3">
      <wordForm  tokens="t2 t3"
                 entry="urn:lex:fr:%E0_cheval" lemma="à_cheval"/>
   </transition>
</fsm>
```

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

Additional information could be added to edges such as probabilities.

9.3.2 Structural ambiguities over tokens

Structural ambiguities may also arise over sequences of tokens, resulting from ambiguities in the tokenization of the annotated document, e.g. speech documents.

Structural ambiguities over tokens are represented by transitions labeled by tokens. The attributes `tinit` and `tfinal` on elements `fsm` 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 advised that the two lattices share their states, whenever possible.

A validity condition has to be expressed between the two levels of structural ambiguity:

> **the tokens covered by word forms along a word form path belong to some token path.**

9.4 Simplified structuring variants

9.4.1 Non ambiguous linear representation

When there is no ambiguity, MAF allows to replace the global lattice notation by a much simpler linear notation where the `token`, `wordForm` and `wfAlt` elements are implicitly chained following their appearance order, as illustrated by the following example:

```
<token id="t1">fer</token>
<token id="t2">à</token>
<token id="t3">cheval</token>
<wordForm entry="urn:lex:fr:fer" tokens="t1"/>
<wordForm entry="urn:lex:fr:%E0" tokens="t2"/>
<wordForm entry="urn:lex:fr:cheval" tokens="t3"/>
```

9.4.2 Mixed linear and lattice representation

Ambiguities are generally localized and it is tempting to also localize the use of the lattice notation only where it is needed. MAF allows to insert local lattice `fsm` in a linear flow of `token`, `wordForm` and `wfAlt` elements.

```
<token id="t0">afin</token>
<token id="t1">de</token>
<fsm init="s0" final="s2">
  <transition source="s0" target="s2">
     <wordForm tokens="t0 t1"
               entry="urn:lex:fr:afin_de" tag="pos.prep"/>
  </transition>
  <transition source="s0" target="s1">
     <wordForm tokens="t0"
               entry="urn:lex:fr:afin" tag="pos.prep"/>
  </transition>
  <transition source="s1" target="s2">
     <wordForm tokens="t1"
               entry="urn:lex:fr:de" tag="pos.prep"/>
  </transition>
</fsm>
<token id="t2">grandir</token>
<wordForm entry="urn:lex:fr:grandir" tag="pos.verb ..." tokens="t2"/>
<token id="t3">,</token>
<wordForm entry="lexicon: ," tag="pos.ponct" tokens="t3"/>
<token id="t4">il</token>
<wordForm entry="urn:lex:fr:il" tag="pos.pronoun ..." tokens="t4"/>
<token id="t5">mange</token>
```

```
<wordForm tokens="t5"
          entry="urn:lex:fr:manger" tag="pos.verb ..."/>
<token id="t6">des</token>
<wordForm tokens="t6"
          entry="urn:lex:fr:une" form="des" tag="pos.det num@pl ..."/>
<token id="t7">pommes</token>
<token id="t8">de</token>
<token id="t9">terre</token>
<fsm init="s8" final="s11">
  <transition source="s8" target="s11">
     <wordForm tokens="t7 t8 t9"
               entry="urn:lex:fr:pomme_de_terre" tag="pos.noun ..."/>
  </transition>
  <transition source="s8" target="s9">
     <wordForm tokens="t7"
               entry="urn:lex:fr:pomme" tag="pos.noun ..."/>
  </transition>
  <transition source="s9" target="s10">
     <wordForm tokens="t8"
               entry="urn:lex:fr:de" tag="pos.prep"/>
  </transition>
  <transition source="s10" target="s11">
     <wordForm tokens="t9"
               entry="urn:lex:fr:terre" tag="pos.noun ..."/>
  </transition>
</fsm>
```

9.5 Expanding the simplified variants

The simplified variants are allowed because they may always be expanded into a global lattice, by applying the steps sketched in the following sub-sections.

9.5.1 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`) , not changing the relative order between tokens.

```
<wordForm entry="urn:lex:fr:manger" tag="pos.verb ...">
    <token id="t6">des</token>
</wordForm>
```

becomes

```
<token id="t6">des</token>
<wordForm entry="urn:lex:fr:manger" tag="pos.verb ..." tokens="t6"/>
```

Note: There is no clear semantic to handle tokens embedded in word forms, themselves embedded in transitions. This case should be avoided.

9.5.2 Wrapping into local lattices

Tokens and word forms outside transitions are embedded into local lattices, `wfAlt` elements being considered as word forms.

```
<token id="t4">il</token>
<wordForm entry="urn:lex:fr:il" tag="pos.pronoun ..." tokens="t4"/>
<token id="t5">mange</token>
<wordForm entry="urn:lex:fr:manger" tag="pos.verb ..." tokens="t5"/>
<token id="t6">des</token>
```

becomes

```
<fsm tinit="s0" tfinal="s1" init="s0" final="s0">
  <transition source="s0" target="s1">
      <token id="t4">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="t4"/
          >
  </transition>
</fsm>
<fsm tinit="s0" tfinal="s1"  init="s0" final="s0">
  <transition source="s0" target="s1">
      <token id="t5">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="t5"
          />
  </transition>
</fsm>
```

Lattice states are local to each lattice.

9.5.3 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 (resp. token) final state of the first lattice equals the word form (resp. token) initial state of the second lattice. Whenever possible, it is recommended, when merging, to rename the lattice states in such a way that the final (resp. final) states for tokens and word form coincide.

The previous example becomes:

```
<fsm tinit="s0" tfinal="s1" init="s0" final="s1">
  <transition source="s0" target="s1">
      <token id="t4">il</token>
  </transition>
  <transition source="s0" target="s1">
      <wordForm entry="urn:lex:fr:il" tag="pos.pronoun ..." tokens="t4"/
          >
  </transition>
</fsm>
<fsm tinit="s0" tfinal="s1"  init="s0" final="s1">
  <transition source="s0" target="s1">
```

```
        <token id="t5">mange</token>
  </transition>
  <transition source="s0" target="s1">
        <wordForm entry="urn:lex:fr:manger" tag="pos.verb ..." tokens="t5"
            />
  </transition>
</fsm>
```

and then

```
<fsm tinit="s0" tfinal="s2" init="s0" final="s2">
  <transition source="s0" target="s1">
        <token id="t4">il</token>
  </transition>
  <transition source="s0" target="s1">
        <wordForm entry="urn:lex:fr:il" tag="pos.pronoun ..." tokens="t4"/
            >
  </transition>
  <transition source="s1" target="s2">
        <token id="t5">mange</token>
  </transition>
  <transition source="s1" target="s2">
        <wordForm entry="urn:lex:fr:manger" tag="pos.verb ..." tokens="t5"
            />
  </transition>
</fsm>
```

9.5.4 Removing `wfAlt`

A transition over a lexical ambiguity, materialized by a `wfAlt` element, may be expanded into two equivalent simpler transitions.

```
<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>
```

becomes

```
<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>
```

The ordering of transitions inside lattices is not pertinent. On the other hand, the ordering of word forms and tokens outside lattices is pertinent. The relative ordering of local lattices is also pertinent.

9.6 Formal description: `wfAlt` and `fsm`

```
maf.flow = (\token | wordForm | wordForm.alt | fsm )+
fsm =
   element fsm {
      ( attribute init { fsm.state },
        attribute final { fsm.state } ) ?,
      ( attribute tinit { fsm.state },
        attribute tfinal { fsm.state } )?,
      transition+
   }
fsm.state = xsd:Name
transition =
   element transition {
      attribute source { fsm.state },
      attribute target { fsm.state },
      (\token | wordForm | wordForm.alt)
   }
wordForm.alt =
   element wfAlt { wordForm+ }
```

10 Header and metadata

The global `maf` element is introduced as a root element to encapsulate morpho-syntactic annotations and carries global metadata relative to the annotated documents.

Two MAF specific metadata categories are introduced for the token standoff notation, namely the `document` and `addressing` attributes. The `addressing` attribute indicates the addressing schema used to refer positions in the annotated document. A full list of such schema will be provided in ISO 24612 proposal "*Linguistic Annotation Framework*" (LAF). The following fragment illustrates the use of these attributes for a video document:

```
<maf document="interview.mpeg" addressing="mpeg7">
  <token id="t0"
         from="T00:01:16:4484F30000"
         to="T00:01:16:14494F30000"
         transcription="mister"/>
  <wordForm tokens="t0" lemma="mister"> ... </wordForm>
  ...
</maf>
```

The other non-mandatory metadata are handled following the recommendations of the *OLAC Metadata Standard* and should therefore be included in an `olac:olac` element.

```
<maf document="http://abu.cnam.fr/cgi-bin/donner_abu?tdm80j2"
     addressing="char_offset">
  <olac:olac xmlns:olac="http://www.language-archives.org/OLAC/1.0/"
   xmlns="http://purl.org/dc/elements/1.1/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.language-archives.org/OLAC/1.0/
      http://www.language-archives.org/OLAC/1.0/olac.xsd">
   <creator>MySuperMorphoTool</creator>
```

```
    <created>2005/09/30</created>
    <hasVersion>1.1</hasVersion>
    <identifier>TDM80MAF.1.1</identifier>
    <replaces>TDM80MAF.1.0</replaces>
    <requires>http://abu.cnam.fr/cgi-bin/donner_abu?tdm80j2</requires>
    <language xsi:type="olac:language" olac:code="fr">French</language>
    <publisher>MyInstitution</publisher>
    <title xml:lang="fr">Le Tour du Monde en 80 Jours version
MAF</title>
    <abstract xml:lang="en"> A set of MAF annotations for Jules Vernes
        famous novel
    <abstract>
    <rightHolder>MyInstitution</rightHolder>
    <license>LGPL-LR</license>
</olac:olac>
    ...
</maf>
```

10.1 Formal description

```
start =
    element maf {
        ( maf.document ,
          maf.addressing )? ,
        tagset ?,
        maf.metadata ?,
        maf.flow
    }
maf.document = attribute document { xsd:anyURI }
maf.addressing = attribute addressing { xsd:NMTOKEN }
maf.metadata  |= notAllowed # to be imported from OLAC
```

The complete list of *addressing schema* allowed by MAF will be inherited from ISO 24612 document on Linguistic Annotation Framework (LAF). A possible list of such schema could include:

- TEI ptrs,
- XML Xpointers,
- character offsets (depending on the original document encoding)
- MPEG7 multimedia addressing (`MediaTimePointType`)

A (informative) RELAX NG compact schema

Note: The following RELAX NG compact schema may be found online at `http://atoll.inria.fr/~clerger/MAF/maf.rnc`

```
# $Id: maf.rnc,v 1.1 2005/09/06 08:38:02 clerger Exp $

default namespace = ""
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"

## Preliminary Relax NG schema for MAF -- Morpho-syntactic Annotation
    Framework
## Eric de la Clergerie <Eric.De_La_Clergerie@inria.fr>


## The following  is for Feature Structures
include "iso-fs-standalone.rnc"

start =
   element maf {
      ( maf.document,
        maf.addressing )? ,
      tagset ?,
      maf.metadata ?,
      maf.flow
   }
maf.document = attribute document { xsd:anyURI }
## To be defined in LAF
maf.addressing = attribute addressing { xsd:NMTOKEN }
## Global Metadata: to be completed
maf.metadata  |= notAllowed # to be imported from OLAC
\token =
   element token {
      attribute id { xsd:ID }?,
      token.information,
      (
         (
          attribute from { DocumentLocation },
          attribute to { DocumentLocation }
         )
       | ## DTD => ,
         (
         [ a:defaultValue = "no" ]
         attribute join { "no" | "left" | "right" | "both" | "overlap" }
            ?,
         text
         )
      )
   }
token.information &= attribute form { string }?
token.information &= attribute phonetic { string }?
token.information &= attribute transcription { string }?
token.information &= attribute transliteration { string }?
```

 ©ISO 2005 – All rights reserved

```
wordForm =
   element wordForm {
      wordForm.identification,
      wordForm.tokens,
      wordForm*,
      wordForm.content ?
   }
wordForm.tokens =
      ( attribute tokens  { xsd:IDREFS }
      | ## DTD => ,
       \token*
      )
wordForm.identification &= attribute entry { xsd:anyURI } ?
wordForm.identification &= attribute lemma { string } ?
wordForm.identification &= attribute form { string } ?
maf.flow = (\token | wordForm | wordForm.alt | fsm )+
fsm =
   element fsm {
      ( attribute init { fsm.state },
        attribute final { fsm.state } ) ?,
      ( attribute tinit { fsm.state },
        attribute tfinal { fsm.state } )?,
      transition+
   }
fsm.state = xsd:Name
transition =
   element transition {
      attribute source { fsm.state },
      attribute target { fsm.state },
      (\token | wordForm | wordForm.alt)
   }
wordForm.alt =
   element wfAlt { wordForm+ }
wordForm.content =
   ( attribute tag { xsd:IDREFS }
   | ## DTD => ,
      fs
   )
fs |= notAllowed          # defined in iso-fs-standalone.rnc
tagset =
   element tagset {
      ( attribute ref { xsd:anyURI }
      |  ## DTD => ,
        (dcs* & fsd* & tagset.lib*)
      )
   }
dcs =
   element dcs {
      attribute local { xsd:NCName },
      ( attribute registered { xsd:anyURI },
         attribute rel { "eq" | "subs" | "gen" } )?,
      element description { text }*
   }
```

```
fsd |= notAllowed                # defined in future iso-fsd.rnc
tagset.lib |= fvLib
tagset.lib |= fLib
fLib |= notAllowed               # defined in iso-fs-standalone.rnc
fvLib |= notAllowed              # defined in iso-fs-standalone.rnc
DocumentLocation = xsd:NMTOKEN  # defined in LAF
```

A.1 Validating MAF documents

For validating MAF document, the first step is to convert the RELAX NG compact schema into an XML RELAX NG schema (for instance using `trang`). Such a XML RELAX NG schema may be found at `http://atoll.inria.fr/~clerger/MAF/maf.rng`

Then, the validation may be performed, for instance, using `xmllint` (from `libxml2`).

```
xmllint --relaxng maf.rng mafdoc.xml
```

It should be noted that some semantics constraints of MAF are not checked by the RELAX NG schema, in particular the constraint between the word form and token paths expressed in Section 9.3.2.

 ©ISO 2005 – All rights reserved

B (informative) DTD

The following DTD is only be an approximation of the RELAX NG schema.

Note: The current DTD is outdated w.r.t. the RELAX NG schema.

```
<?xml encoding="UTF-8"?>

<!-- $Id: maf.rnc,v 1.1 2005/09/06 08:38:02 clerger Exp $ -->

<!--
Preliminary Relax NG schema for MAF -- Morpho-syntactic Annotation
    Framework
Eric de la Clergerie <Eric.De_La_Clergerie@inria.fr>
-->

<!ELEMENT maf (tagset?,(fsm|token|wordForm|wfAlt)+)>
<!ATTLIST maf
  xmlns CDATA #FIXED ''>

<!ATTLIST maf
  document CDATA #IMPLIED
  addressing NMTOKEN #IMPLIED>

<!ENTITY % DocumentLocation "NMTOKEN">

<!ELEMENT token (#PCDATA)>
<!ATTLIST token
  xmlns CDATA #FIXED ''
  id ID #IMPLIED
  form NMTOKEN #IMPLIED
  transcription NMTOKEN #IMPLIED
  transliteration NMTOKEN #IMPLIED
  from %DocumentLocation; #REQUIRED
  to %DocumentLocation; #REQUIRED
  join (no|left|right|both|overlap) 'no'>

<!ELEMENT wordForm (fs,token*,wordForm*)>
<!ATTLIST wordForm
  xmlns CDATA #FIXED ''
  entry CDATA #IMPLIED
  lemma CDATA #IMPLIED
  form CDATA #IMPLIED
  tag CDATA #REQUIRED
  tokens IDREFS #REQUIRED>

<!ELEMENT fsm (transition)+>
<!ATTLIST fsm
  xmlns CDATA #FIXED ''
  init IDREF #IMPLIED
  final IDREF #IMPLIED
  tinit IDREF #IMPLIED
  tfinal IDREF #IMPLIED>
```

 ©ISO 2005 – All rights reserved

```
<!ELEMENT transition (wordForm|wfAlt|token)>
<!ATTLIST transition
  xmlns CDATA #FIXED ''
  source IDREF #REQUIRED
  target IDREF #REQUIRED>

<!ELEMENT wfAlt (wordForm)+>
<!ATTLIST wfAlt
  xmlns CDATA #FIXED ''>

<!ELEMENT tagset (dcs|fsd|fvLib|fLib)*>
<!ATTLIST tagset
  xmlns CDATA #FIXED ''
  ref CDATA #REQUIRED>

<!ELEMENT dcs (description)*>
<!ATTLIST dcs
  xmlns CDATA #FIXED ''
  local NMTOKEN #REQUIRED
  registered CDATA #IMPLIED
  rel (eq|subs|gen) #IMPLIED>

<!ELEMENT description (#PCDATA)>
<!ATTLIST description
  xmlns CDATA #FIXED ''>

<!-- should be completed by DTD's for Feature Structures -->

<!ELEMENT fsd EMPTY>
<!ATTLIST fsd
  xmlns CDATA #FIXED ''>
```

C (informative) Illustrative examples

C.1 Tagsets

C.2 Demonstrator

A preliminary demonstrator covering most of the functionalities provided by MAF may be tried on line at `http://atoll.inria.fr/mafdemo`.

 ©ISO 2005 – All rights reserved

D (illustrative) Morpho-syntactic Data Categories

This annexe lists and documents the morpho-syntactic data categories used in the MAF examples.

A repository of data categories, including morpho-syntactic data categories, may be found at `http://syntax.inist.fr/`.

`/grammatical gender/` with conceptual values `/feminine/`, `/masculine/`

`/grammatical number/` with conceptual values `/singular/`, `/plural/`

`/grammatical pos/` with conceptual values `/noun/`, `/verb/`, `/preposition/`, `/determiner/`, conceptadverb

`/grammatical mood/` with conceptual values `/indicative/`, `/subjunctive/`

`/grammatical tense/` with conceptual values `/present/`

`/grammatical person/` with conceptual values `/first/`, `/second/`, `/third/`

 ©ISO 2005 – All rights reserved

E (informative) UML notions used within MAF

E.1 Introduction

MAF complies with the specifications and modeling principles of UML as defined by OMG [32]. UML is well defined and broadly used in the industry. MAF uses a subset of UML that is relevant for linguistic description.

The following notions are used:

- The notion of class
- The notion of relationship
- The notion of instance
- The notion of package

E.2 The notion of class

A class is a named descriptor for a set of objects that share the same attribute s and relationships. Classes are described within a class model.

E.3 The notion of attribute

An attribute is the description of a named element of a specified type in a class; each object of a class separately holds a value of the type.

E.4 The notion of relationship

A relation is a connection between classes. This includes association and generalization. Relations are described within a class model.

E.5 The notion of association

An association is a relationship between two specified classes that describes connections among their objects. The extension of the association is a collection of such links. Associations are the *glue* that holds together the model: without associations, there is only a set of isolated classes. An association holds two ends. Each end has "a multiplicity" and an ordering qualifier.

The multiplicity is the specification of the range of allowable cardinality values that a collection may assume. The multiplicity range is an integer interval with its minimum and maximum values.

An ordering qualifier specifies whether the connection forms a set (an unordered collection) or a list (an ordered collection).

E.6 The notion of aggregation

An aggregation is a form of association that specifies a whole-part relationship between an aggregate (a whole) and a constituent part. It is not permissible for both ends to be aggregates.

E.7 The notion of generalization

A generalization relationship is a directed relationship between two classes. On e class is called the parent or the super-class, and the other is called the child or the sub-class. The parent is the description of a set of objects with common properties over all children. The child is a description of a subset of those objects that have the properties of the pa rent but that also have additional properties peculiar to the child. A parent may have more than on e child and a child may have more than one parent. Generalization is a transitive and anti-symmetrical relationship. No directed generalization cycles are allowed. A child inherits the attributes and associations of its parent.

E.8 The notion of instance

An instance is an object that conforms to a class. Instances are not described within a class model but within an instance model (sometimes called an object model).

E.9 The notion of package

A package is a grouping of classes and relations. Usually there is a single root package that owns the entire model for a system. A package may contain nested packages. Packages may have dependencies to other packages.

E.10 Graphical notations

Each notion has a graphical notation that is precisely defined as follows:

References

[1] Gilles Adda, Joseph Mariani, Patrick Paroubek, Martin Rajman, and Josette Lecomte. L'action grace d'évaluation de l'assignation de parties du discours pour le français. *Langues*, 2-2:119-130, 1999.

[2] Consortium Genelex. *Projet Eureka Genelex - Rapport sur la couche Syntaxique - Rapport sur la couche morphologique*, 1993.

[3] Nelson Francis and Henry Kučera. *Manual of Information to accompany a Standard Corpus of Present-day Edited American English, for use with Digital Computers.* Brown University, Providence, Rhode Island, Revised 1989.

[4] Eva Hajicova, Jarmila Panevova, and Petr Sgall. Language ressources need annotations to make them reusable: the prague dependency treebank. In *Proceedings First Conference on Linguistic Resources*, pages 713-718, Granada, 1998.

[5] Nancy Ide, Jean Véronis, and Greg Priest-Dorman. Corpus encoding standard. Technical report, EAGLES/MULTEX, 1996.

[6] Timo Järvinen. Annotating 200 millions words: the bank of english project. In *Proceedings 15th COLING*, pages 565-568, Kyoto, 1994.

[7] Timo Järvinen. *Bank of English and beyond*, chapter Treebanks (éd. Anne Abeillé). Kluwer Academic Publishers, 2000.

[8] Patrick Paroubek and Martin Rajman. *Etiquetage morpho-syntaxique*, volume Ingénierie des langues, chapter in Ingénierie des Langues (éd Jean-Marie Pierrel). HERMES-Science, Paris, 2000.

[9] Antonio Sanfilippo. *EAGLES Subcategorization Standards*, 1996. http://www.icl.pi.cnr.it/EAGLES96/syntax/syntax.html.

[10] Kevin Sinclair. Preliminary recommendations on corpus typology. Technical report, EAGLES, 1996.

[11] Jean Véronis and Liliane Khouri. étiquetage grammatical multilingue : le projet multex. *TAL*, 36, 1995.

[12] Ursula von Rekowski. Elm-fr : Specifications for french morphosyntax, lexicon specification and classification guidelines. EAGLES document, 1996.