

**ISO/TC 37/SC 4**

Date: 2017-03-28

**ISO 24615-2:2017(E)**

ISO/TC 37/SC 4/WG 6

Secretariat: KATS

**Language resource management — Syntactic annotation framework  
(SynAF) — Part 2: XML-serialization (ISOTiger)**

*Gestion des ressources linguistiques — Cadre d'annotation syntaxique — Partie 2: Sérialisation XML  
(ISOTiger)*

Document type: International Standard  
Document subtype:  
Document stage: (60) Publication  
Document language: E

X:\TA2\TG2-2\NA105\Projekte\ISO\_24615-2\ISO\_24615-2\ISO\_24615-2\_(E).docx STD Version 2.8f



**COPYRIGHT PROTECTED DOCUMENT**

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

<b>Contents</b>		<b>Page</b>
<b>Foreword</b> .....		<b>iv</b>
<b>Introduction</b> .....		<b>v</b>
<b>1</b>	<b>Scope</b> .....	<b>1</b>
<b>2</b>	<b>Normative references</b> .....	<b>1</b>
<b>3</b>	<b>Terms and definitions</b> .....	<b>1</b>
<b>4</b>	<b>Graph structure and meta model</b> .....	<b>2</b>
<b>5</b>	<b>Meta-model objects in XML serialization</b> .....	<b>4</b>
<b>6</b>	<b>Primary data and terminal representation</b> .....	<b>5</b>
<b>6.1</b>	<b>Introduction</b> .....	<b>5</b>
<b>6.2</b>	<b>Sequential representation</b> .....	<b>5</b>
<b>6.3</b>	<b>Standoff representation</b> .....	<b>6</b>
<b>6.4</b>	<b>Typing of nodes and edges</b> .....	<b>7</b>
<b>7</b>	<b>Annotations</b> .....	<b>7</b>
<b>7.1</b>	<b>Introduction</b> .....	<b>7</b>
<b>7.2</b>	<b>Domain declaration</b> .....	<b>8</b>
<b>7.3</b>	<b>Declaring annotations in an external file</b> .....	<b>8</b>
<b>7.4</b>	<b>Feature values</b> .....	<b>9</b>
<b>7.5</b>	<b>Data category references</b> .....	<b>9</b>
<b>7.6</b>	<b>Using the @type attribute</b> .....	<b>10</b>
<b>8</b>	<b>Corpus structure</b> .....	<b>11</b>
<b>Bibliography</b> .....		<b>13</b>

## Foreword

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

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

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

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

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

ISO 24615-2 was prepared by Technical Committee ISO/TC 37, Terminology and other language and content resources, Subcommittee SC 4, Language resource management, in collaboration with Humboldt-Universität zu Berlin and INRIA.

## Introduction

The need for standardization of syntactic annotation was recognized and addressed in detail with the publication of ISO 24615-1:2014 (SynAF). As a result of the work on ISO 24615-1:2014, it was anticipated that such a reference model for syntactic annotation should be associated with a concrete XML serialization in order to meet the specific needs of such applications as syntactic parsers or syntactic treebanks, where representations have to be exchanged and re-used. Furthermore, such a serialization should be independent from the theoretical orientation and specific details of any specific annotation scheme.

ISO 24615-2 answers this need on the basis of the seminal work carried out on the TigerXML format (Mengel & Lezius 2000). This starting point was chosen as a reference because it is widely used as a de facto standard for unrelated XML treebanks, with the advantages in terms of interoperability offered by its XML-based representations, as opposed to other frequently used formats, in particular the Penn Treebank bracketing format (Bies et alii, 1995) or the CoNLL format for dependency structures (cf. Nilsson et alii, 2007).

The document is designed to complement ISO 24615-1:2014 and to coordinate closely with ISO 24610, ISO 24612, ISO 24611, and ISO 12620.

This document therefore extends ISO 24615-1:2014 (SynAF) with an XML model called ISO Tiger for the interchange of syntactically annotated data which is both standardized as well as language- and theory-independent. The proposed format directly instantiates all features of the meta-model defined in ISO 24615-1:2014 and defines concrete serialized interfaces to the complementary ISO 24611 (MAF, Morpho-syntactic annotation framework) and ISO 12620, which provides the background for the ISOCat.org data category registry.



# Language resource management — Syntactic annotation framework (SynAF) — Part 2: XML-serialization (ISOTiger)

## 1 Scope

This document describes ISOTiger, an XML-conformant serialization of the ISO 24615-1:2014 meta-model, with the objective of supporting interoperability across language resources or language processing components in the domain of syntactic annotations. As an extension of ISO 24615-1:2014, this document is also coordinated with ISO 24612 (LAF, Linguistic Annotation Framework).

## 2 Normative references

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

ISO 12620, *Terminology and other language and content resources — Specification of data categories and management of a Data Category Registry for language resources*

ISO 24610 series, *Language resource management — Feature structures*

ISO 24611, *Language resource management — Morpho-syntactic annotation framework (MAF)*

ISO 24612, *Language resource management — Linguistic annotation framework (LAF)*

ISO 24615-1, *Language resource management — Syntactic annotation framework (SynAF) — Part 1: Syntactic model*

## 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 12620, in the ISO 24610 series, in ISO 24611, ISO 24612, ISO 24615-1, and the following apply.

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

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

### 3.1

#### **domain**

class of elements to which a certain set of labels can be assigned

Note 1 to entry: Domains can refer generally to the set of all edges, terminal nodes or non-terminal nodes.

### 3.2

#### **label**

unit of annotation consisting of the name of a feature and a value, which together can be applied to appropriate model elements and add arbitrary feature-value annotations to such elements.

### 3.3

#### **primary data**

initial raw linguistic content that is being encoded

### 3.4

#### **sequential representation**

representation of annotation content where the XML element structure mirrors the sequence of linguistic objects in the primary source

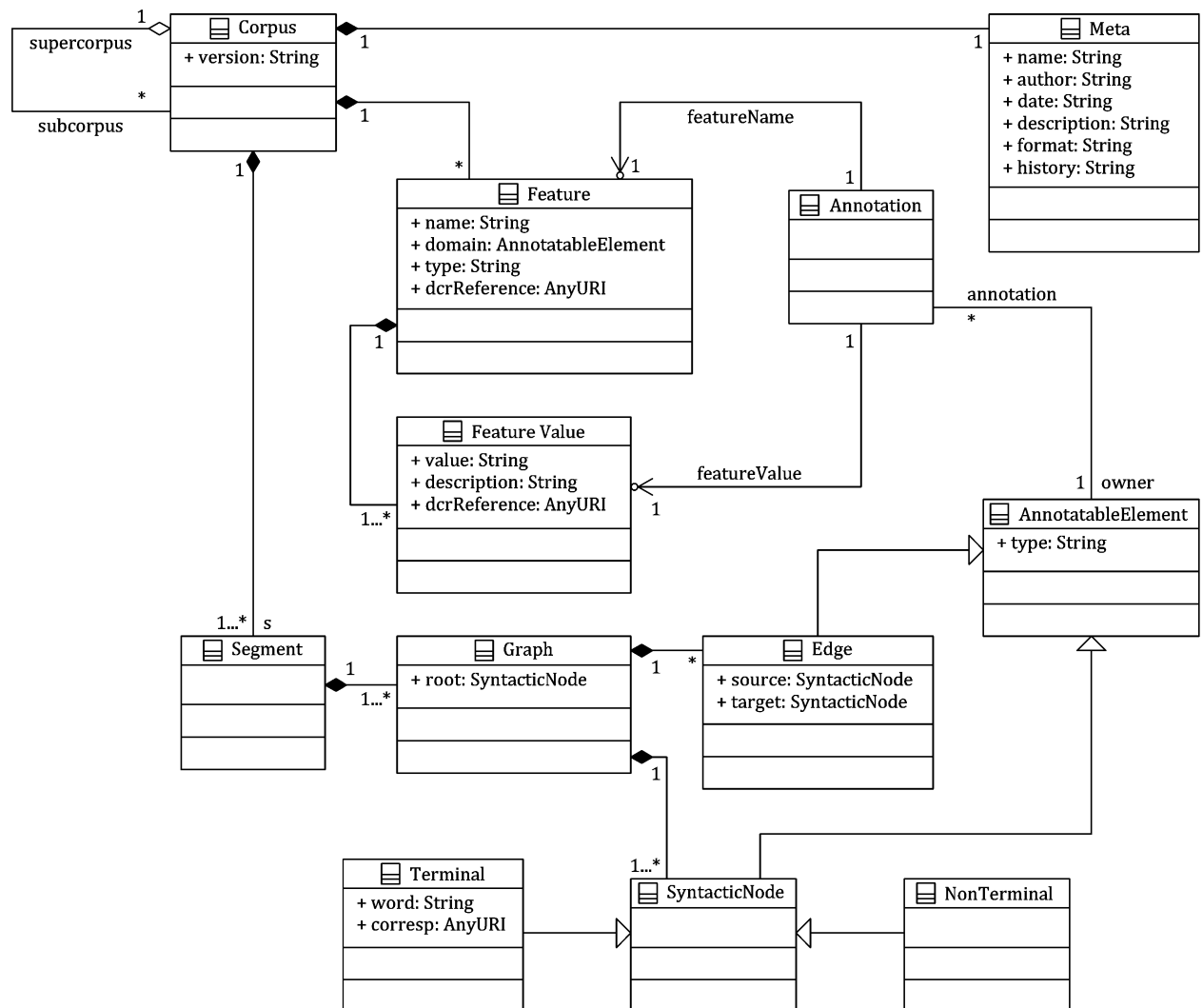
## **4 Graph structure and meta model**

In ISOTiger, annotations are represented in a graph structure. The graph structure can be described as  $G = (V, E, A)$  with:

- a set of nodes  $V$ , and
- a set of edges  $E$  with  $e = (v \in V, v \in V) \in E$ , and
- a set of annotations  $A$  where an annotation  $a$  is defined by a feature-value pair, and
- a function *annot*:  $E \cup V \rightarrow A$

A graph represents a bundle of interrelated nodes and edges. It is not specified which parts of a primary text are covered by a single graph: e.g. a sentence, a sub-sentence, a chapter or a whole text. Linguistic annotations represented by labels can be attached to nodes as well as edges.





**Figure 1 — Meta-model of ISOTiger**

The ISOTiger meta-model consists of three parts:

- a) the structural organization of corpora and associated meta-data;
- b) an annotation tagset definition; and
- c) the linguistic annotation graph.

The structural organization of corpora is represented by a recursively defined corpus element (Corpus) and its corresponding metadata (Meta). A corpus can contain subcorpora.

The annotation tagset definition is represented by a list of categories (Feature) containing the name of a category (Feature.name) and a list of category values (FeatureValue). Each FeatureValue contains a string representation of the value (FeatureValue.value). Together, both elements declare a tagset which is part of a specific corpus object. Such a tagset declaration is derivable, which means that all categories defined in a supercorpus object can also be used by its subcorpus objects. Further attributes are used to declare to which types of nodes and edges a category is applicable. Both elements allow reference to ISOcat entries in compliance with ISO 12620 via URIs (Unified Resource Identifiers) in the attributes Feature.dcrReference and FeatureValue.dcrReference.

The final part of the ISOTiger meta-model, the linguistic annotation graph, defines a set of elements containing the primary data and the annotation structure covering the primary data. It consists of the graph element itself (Graph), two classes of syntactic nodes (Terminal and NonTerminal), an edge element (Edge), and an annotation element (Annotation) realizing the *annot*-function and therefore referring to a feature name and its value. Graph is contained within Segment, which is a grouping mechanism to aggregate a set of syntactic nodes together. Such a group can have linguistic structural semantics, corresponding usually to a sentence, but possibly also to a line in a manuscript or other meaningful segments depending on the application and annotation scheme used by a specific project.

A terminal node (in ISO 24615-1 referred to as T\_Node) constitutes the point of reference to the primary data. This can be a direct reference to a text-span within the ISOTiger document or an indirect reference to an object outside the ISOTiger model, e.g. an element contained by a MAF file in compliance with ISO 24611, the Morpho-syntactic annotation framework. A non-terminal node is an inner node, referring directly or indirectly to a terminal node within the ISOTiger document.

An Edge always has to have a source and a target node. Both of them can be either a terminal or a non-terminal node.

### 5 Meta-model objects in XML serialization

The names of the XML elements and attributes in ISOTiger follow those of the corresponding meta-model elements and attributes. All ISOTiger-elements belong to the following namespace: <http://www.iso.org/ns/SynAF>. In the present standard, unless specified otherwise, all XML-elements will be assumed to belong to this namespace.

Terminal nodes in the XML serialization are represented by the <t> element and are nested together in a <terminals> element.

A Segment node is represented by the <s> element. The <s> element must contain one or more <graph> elements, which may be used to express possible multiple annotation graphs alternating within a single segment or to represent a sequence of subgraphs.

A non-terminal node is represented by the <nt> element and is nested in the <nonterminals> element.

An edge is represented by the <edge> element and its source is given by the surrounding node element. An <edge> element is therefore always the child element of a <t> or an <nt> element. The target is given by the @target attribute and must refer to a node within the same ISOTiger-document.

Example 1 shows the ISOTiger-representation of the graph structure.

#### EXAMPLE 1 ISOTiger graph structure

```
<s>
  <graph xml:id="s1_g1">
    <terminals>
      <t xml:id="s1_t1"/>
      <t xml:id="s1_t2"/>
    </terminals>
    <nonterminals>
      <nt xml:id="s1_nt1">
        <edge xml:id="s1_e1" target="#s1_t1" />
        <edge xml:id="s1_e2" target="#s1_t2" />
      </nt>
    </nonterminals>
  </graph>
</s>
```

## 6 Primary data and terminal representation

### 6.1 Introduction

Terminal nodes and the primary data to which they refer can be defined in two different ways. The primary data can either be included within the ISOTiger-document (sequential representation), or they can be specified in another file and referred to externally (standoff representation). The sequential representation makes it possible to have all data in just one ISOTiger-document, whereas the standoff representation makes it possible to refer to other formats, for instance, a MAF file containing tokens and wordForms.

Both options are possible and the decision is largely made based on the needs of the corpus project.

### 6.2 Sequential representation

In a sequential representation, primary data is represented sequentially directly within the <t> elements of each <s> element in the document. As a result, all the primary data of a document is grouped together in one single XML-document. This improves human readability and basic machine processing, but reduces representational flexibility. Tokens of the primary data are encoded as values of the @word attribute of the <t> element, as illustrated in Example 2.

EXAMPLE 2 Simple sequential representation of word forms

```
<t xml:id="s1_t1" word="two"/>
<t xml:id="s1_t2" word="words"/>
```

Example 2 shows two terminal nodes, the first containing the word "two" and the second containing the word "words". The order of the tokens in the primary data is mirrored by the order of the XML-elements.

However, it should be pointed out that this representation does not need to fully preserve primary data, since only textual material that has a corresponding <t> element is preserved. This means that parts of an utterance that are not considered to be tokens (untokenized material), such as whitespaces between words or enumerations, non-linguistic characters, etc., will be lost. Example 3 and Example 4 illustrate this.

EXAMPLE 3 Primary data

This is a sample.

EXAMPLE 4 Sequential representation

```
<t xml:id="s1_t1" word="This"/>
<t xml:id="s1_t2" word="is"/>
<t xml:id="s1_t3" word="sample"/>
<t xml:id="s1_t4" word="."/>
```

In these examples, it is not possible to resolve whether there was originally a whitespace character between the tokens corresponding to the @word attributes of the terminals or not. In the case of the terminals "s1\_t1" and "s1\_t2" there was one, whereas between "s1\_t3" and "s1\_t4" there was no intervening whitespace. This can be handled by means of the @join attribute as provided by ISO 24611 (MAF). It is also possible to omit textual material on purpose, leading to the text span corresponding to the word "a" becoming lost in the ISOTiger-document.

Because of the restrictions of XML-attributes, reserved XML-signs such as angle brackets (" $<$ ") cannot be represented inside an XML-attribute-value. These are therefore escaped using the sequence "&lt;" (see Example 5), which, in some cases, can cause confusion in character-based offset calculations.

EXAMPLE 5 Sequential representation with character escaping

```
<t xml:id="s1_t4" word="&lt;"/>
```

Cases where the primary data contains additional mark-up can be difficult to handle on a sequential encoding strategy.

### 6.3 Standoff representation

Problems concerning special characters or mark-up embedded in the primary data, described in Section 6.2, can be avoided by using a standoff representation. In a standoff representation, the @corresp attribute of the <t> element is used to point to a sequence of tokens or word forms in the primary data. The @corresp attribute replaces the @word attribute and contains a URI reference to another resource, for example a MAF file. If both attributes are provided, the @word attribute takes precedence and the @corresp attribute can usually be ignored. The following examples show the use of the standoff representation with the @corresp attribute. For these examples the token and wordForm elements from ISO 24611 (MAF) data model are applied, and the references into the primary data utilize anchors to refer to locations in between the base units of the data representation as specified in ISO 24612 (LAF).

EXAMPLE 6 Locations in the primary data document

```
<s xml:id="s1">We can...</s>
```

This sentence is associated with the following segmentation, with an inter-character numbering:

```
|W|e| |c|a|n|...  
0 1 2 3 4 5 6
```

EXAMPLE 7 MAF standoff representation, in TEI flavour

```
<w xml:id="t1" corresp="#string-range(s1,0,2)">  
<w xml:id="t2" corresp="#string-range(s1,3,6)">
```

...

```
<span type="wordForm" xml:id="wf_1" lemma="we" corresp="#t1"/>  
<span type="wordForm" xml:id="wf_2" lemma="can" corresp="#t2"/>
```

EXAMPLE 8 ISOTiger standoff representation

```
<terminals>  
  <t xml:id="s9_1" corresp="example.maf#wf_1"/>  
  <t xml:id="s9_2" corresp="example.maf#wf_2"/>  
</terminals>
```

These examples refer to three different files: the primary text document, an ISO MAF file containing the tokenization and an aggregation of tokens to wordForms, and an ISOTiger-document referring to the MAF-document.

## 6.4 Typing of nodes and edges

Nodes and edges can also be given types in ISOTiger. A type is a special kind of annotation and is represented by the @type attribute contained in a <t>, <nt> or <edge> element. For example, the type of an edge can be "dep" to denote a dependency edge and to distinguish it from constituency edges.

The domain of possible @type values is not defined by ISOTiger, though it is recommended to use ISOcat (an implementation of ISO 12620) for the identification of relevant data categories.

For a more specific definition of a @type value, it is possible to define the domain of the values via an annotation with a <feature> element (see detail in 7.6). If the @type attribute is not used within a <t>, <nt> or <edge> element, they will be implicitly set to a default value. In the case of an edge, the type will be set to "edge", in the case of a terminal node it will be set to "t" and in the case of a non-terminal node it will be set to "nt".

The @type attribute is also used for the declaration of the domain for annotation-feature names and the domain for annotation-feature values as described in Clause 7.

## 7 Annotations

### 7.1 Introduction

Annotations are attribute-value pairs that can be appended to nodes and edges. In ISOTiger, this is mirrored by the names and values of XML attributes. In general, in ISOTiger there are no restrictions for setting names or values for annotations depending on specific linguistic theories. Therefore, it is possible to append any feature-value pair to a <t>, <nt> or <edge> element, except for the reserved attributes of ISOTiger, for example @type, @word, @corresp and @domain. These attributes have fixed semantics and cannot be overridden.

Example 9 shows the annotation mechanism for nodes and edges.

**EXAMPLE 9** Annotation declaration for <t>, <nt> and <edge> elements

```
<terminals>
  <t xml:id="s1_t1" word="I" lemma="I" pos="PP"/>
  <!-- ... -->
</terminals>
<nonterminals>
  <nt xml:id="s1_nt1" cat="NP">
    <edge xml:id="s1_e1" label="HD" target="#s1_t1"/>
  </nt>
  <!-- ... -->
</nonterminals>
```

In Example 9, the word "I" is annotated with the part-of-speech annotation "PP" (standing for "personal pronoun"), and the lemma-annotation "I". Neither "pos" nor "lemma" are reserved attribute names – they could also be named, e.g., "part-of-speech" or "lem", or use any other non-reserved name, so it is recommended to use references to the ISOcat Data Category Registry for disambiguation (see 7.5). The non-terminal "s1\_nt1", annotated with the attribute @cat (for "category") with the value "NP" (for "nominal phrase") is syntactically headed by the terminal "s1\_t1". This relationship is expressed by the edge "s1\_e1" which has as its source "s1\_nt1", and as its target "s1\_t1", and bears the annotation ("label='HD'"), which indicates headedness.

## 7.2 Domain declaration

In ISOTiger, it is recommended to define a domain for annotation names. This must be done in the header section of the corpus, in the <annotation> element inside the element <head>. The @domain attribute indicates whether the annotation being defined should be applied to terminals, non-terminals or edges. For the annotations present in Example 9, the relevant domains are set as shown in Example 10.

### EXAMPLE 10 Domain declaration for annotation attributes

```
<head>
  <annotation>
    <feature xml:id="f1" name="pos" domain="t" />
    <feature xml:id="f2" name="lemma" domain="t" />
    <feature xml:id="f3" name="cat" domain="nt"/>
    <feature xml:id="f4" name="label" domain="edge"/>
  </annotation>
</head>
```

The mapping between the annotation attribute and the domain declaration is achieved by resolving the name of the XML attribute of the annotation and the @name attribute of the element <feature>. The attribute @domain specifies that the declared feature is applicable only to the given kind of ISOTiger-elements. In the given example, this means that the feature declarations for "pos" and "lemma" can only be used by <t> elements, "cat" can only be used by <nt> elements and "label" can only be used by <edge> elements. The attribute @domain is optional and need not be set. If it is not set, it means that the feature declaration is available for all three kinds of elements.

## 7.3 Declaring annotations in an external file

It is also possible to declare annotations in a separate XML file referenced from within the corpus. This can be useful when authoring or curating large numbers of files which all have the same annotation scheme. In this case, the element <external> and the attribute @corresp should be used to specify the location of the annotation declaration file. Example 11 illustrates the use of external annotation declaration files.

### EXAMPLE 11 Referencing external annotation declarations

```
<corpus xmlns="http://www.iso.org/ns/SynAF">
  <head>
    <annotation>
      <external corresp="MyAnnotations.xml"/>
    </annotation>
  </head>
</corpus>
```

An external annotation declaration looks exactly like an annotation declaration (see 7.2). Example 12 shows how such a declaration works.

### EXAMPLE 12 Content of an external declaration annotation

```
<annotation xmlns="http://www.iso.org/ns/SynAF">
  <feature xml:id="f1" name="pos" domain="t" />
  <feature xml:id="f2" name="lemma" domain="t" />
  <feature xml:id="f3" name="cat" domain="nt"/>
  <feature xml:id="f4" name="label" domain="edge"/>
</annotation>
```

## 7.4 Feature values

Beyond domains for annotation names, it is also possible to restrict the use of annotation values. This can be achieved by using the <value> element inside the <feature> element. The <value> element has a @name attribute, which identifies the value name and also works as an anchor for annotation values. The element can also contain text, which acts as a description for the annotation-value. Example 13 shows the use of an annotation value declaration.

EXAMPLE 13 Referencing domains for annotation feature name and value

```
<head>
  <annotation>
    <feature xml:id="f2" name="pos" domain="t">
      <value xml:id="f2_1" name="PP">Personal pronoun</value>
    </feature>
  </annotation>
</head>
<!-- ... -->
<terminals>
  <t id="s1_t1" word="I" pos="PP"/>
</terminals>
```

In the case of Example 13, the set of possible values for an annotation with the name "pos" only contains a "PP"-element, standing for a personal pronoun. In this case, no other values are possible for this annotation of terminal nodes. To increase the set of possible annotation values, it is possible to add further <value> elements to the <feature> element. The number of possible annotation values is not limited. If there are no <value> elements given inside a <feature> element, the set of annotation values is not defined and can be any string legal as an XML-attribute value. In this case, validation of annotation values cannot take place. This is useful e.g. for lemma annotations, for which it is typically not recommended to define the domain of possible annotation values.

## 7.5 Data category references

In many cases, annotation declarations are not only interesting for one specific corpus or project, but are of interest to a whole community. ISOTiger therefore offers the possibility of linking annotations to ISOcat data categories (see also <http://www.isocat.org/>). The attribute @datcat can be used in <feature> and <value> elements to refer to ISOcat categories via URI syntax. Example 14 shows the use of the @datcat attribute.

EXAMPLE 14 Using a data category registry in annotation declarations

```
<corpus xmlns="http://www.iso.org/ns/SynAF" version="2.0.5"
  xml:id="c1" xmlns:dcr="http://www.isocat.org/ns/dcr">
  <head>
    <annotation>
      <feature xml:id="f2" name="pos" domain="t"
        dcr:datcat="http://www.isocat.org/datcat/DC-396">
        <value
          xml:id="f2_1" name="PP"
          dcr:datcat="http://www.isocat.org/datcat/DC-1463"/>
        </feature>
      </annotation>
    </head>
    <!-- ... -->
    <terminals>
      <t id="s1_t1" word="I" pos="PP"/>
    </terminals>
  </corpus>
```

In Example 14, a terminal covering the word "I" is annotated by a part-of-speech annotation ("pos") with the value "personal pronoun". The domain for the annotation name and the annotation value is declared in the <annotation> element, as seen in the examples above. Additionally, in this example, the @datcat attribute is used to refer to an ISOcat data point. The annotation name is not only declared within the ISOTiger-document, but also in the shared registry. This can also substitute the description inside the <value> element, provided that an adequate description is given by the ISOcat entry, though ISOTiger does not prohibit using both mechanisms concurrently.

## 7.6 Using the @type attribute

In cases where a finer granularity is required for the declaration of annotation domains, it is possible to use the @type attribute inside the <t>, <nt> and <edge> elements. As mentioned in 6.4, this attribute determines the type of the given element and also determines a correspondence to the declaration of the annotation domain. The element <feature> also contains an optional @type attribute, which can restrict the subset of nodes and edges defined by the annotation domain. Example 15 demonstrates this mechanism.

**EXAMPLE 15** Using the @type attribute for annotation domain declaration

```
<head>
  <annotation>
    <feature xml:id="f2" type="wordform" name="lemma" domain="t">
      <value xml:id="f2_1" name="PP">Personal pronoun</value>
    </feature>
  </annotation>
</head>
...
<s>
  <graph>
    <terminals>
      <t id="s1_t1" word="I" type="wordform" lemma="I" pos="PP"/>
    </terminals>
    ...
  </graph>
</s>
```

In this example, an annotation name for terminal "s1\_t1" is defined by the feature "f2". The @type attribute guarantees that the feature "f2" only declares an annotation name for terminals of the type "wordform". A terminal without this type cannot be referenced within that domain. If no @type attribute is given in the <feature> element, the domain works for all elements respecting the domain declaration (<t>, <nt> or <edge> elements).

In ISOTiger, it is also possible to declare the values of the @type attribute. The relevant mechanism is the same as for declaring annotation values, namely via the <feature> and <value> element. Example 16 shows the declaration of the domain for types.

**EXAMPLE 16** Declaration of values of the @type attribute

```
<feature xml:id="t1" name="type" domain="t">
  <value xml:id="t1_1" name="wordform">Abstraction over token(s), corresponding to a
  linguistic entity, cf. wordForm in ISO MAF
</value>
...
</feature>
<feature xml:id="t2" name="type" domain="edge">
  <value xml:id="t2_1" name="dep">Grammatical dependency</value>
  ...
</feature>
```



Example 16 declares sample values of the @type attribute within the domains of terminals and edges and provides glosses for these values.

## 8 Corpus structure

In ISOTiger, it is possible to represent a complete corpus structure containing several sub-corpora in a single ISOTiger-document. But in each ISOTiger-document, only one root corpus may exist, represented by the <corpus> element. The corpus element must specify the format version of the data using the @version attribute. A subordinate corpus structure, or more precisely a subcorpus, can be created by using the <subcorpus> element inside the <corpus> element. @version is inherited by subcorpora and should not be specified individually for each subcorpus. The <subcorpus> element is a recursive structure which allows for an unbounded depth of embedded subcorpora. Example 17 shows the corpus structure in ISOTiger.

### EXAMPLE 17 Corpus structure with subcorpora

```
<corpus xml:id="c1" version="2.0.5" xmlns="http://www.iso.org/ns/SynAF">
  <head>
    ...
  </head>
  <body/>
  <subcorpus xml:id="c2">
    <head/>
    <body/>
    <subcorpus xml:id="c3">
      <head/>
      <body/>
    </subcorpus>
  </subcorpus>
  <subcorpus xml:id="c4">
    <head/>
    <body/>
  </subcorpus>
</corpus>
```

Example 17 shows a corpus "c1" containing in total 3 subcorpora. Two of them ("c2" and "c4") are directly contained in "c1", and "c3" is directly contained in "c2" and therefore indirectly contained in "c1".

The element <head> groups the corpus metadata, while the element <body> contains segments (<s>), which may contain one or more <graph> elements, see Example 18.

### EXAMPLE 18 The content of <body>

```
<body>
  <s xml:id="s1">
    <graph xml:id="s1_g1"/>
    <graph xml:id="s1_g2"/>
    ...
  </s>
  <s xml:id="s2">
    <graph xml:id="s2_g1"/>
    <graph xml:id="s2_g2"/>
    ...
  </s>
  ...
</body>
```

The relationship between the contents of the <graph> elements is not prescribed by this standard. They may cover exactly the same primary data, creating alternative annotation graphs, they may cover separate spans of <s>, and they may also partially overlap.

A corpus or subcorpus can be annotated by meta-annotations. Using meta-annotations differs from the way nodes and edges are annotated. ISOTiger provides a fixed set of possible meta-annotations, though it is possible to link an ISOTiger-document in a standoff fashion to an XML-file specifying meta-annotations in either ISOTiger or even a different format. Meta-annotations are limited to the header of a corpus or subcorpus. They are identified by the enclosing element <meta>. The meta-annotations natively supported in the ISOTiger-format come from the TigerXML-format and are usable for the <corpus> and the <subcorpus> elements. The possible native meta-annotations are the following: name, author, date, description, format and history. Of these, only the name is obligatory. A meta-annotation is represented by an XML-element with the same name, containing the value as text inside the element. Example 19 shows the use and the set of possible meta-annotations.

### EXAMPLE 19 Meta-data representation

```
<meta>
  <name>name of the corpus</name>
  <author>author of the corpus</author>
  <date>creation date of the corpus</date>
  <description>description of the corpus</description>
  <format>original format, if the corpus was not born as an ISOTiger corpus</format>
  <history>version of the corpus</history>
</meta>
```

Linking to an external meta-data file is done as follows ("name" must still be specified within the ISOTiger-document).

### EXAMPLE 20 Meta-data referring to an external XML-file

```
<corpus xmlns="http://www.iso.org/ns/SynAF">
<head>
  <meta>
    <name>My Corpus</name>
    <external corresp="MyMetadata.xml"/>
  </meta>
  ...
</head>
```

The external definition of meta-data looks the same as an internal use of meta-data.

## Bibliography

- [1] Bies, A., Ferguson, M., Katz, K. and MacIntyre, R. 1995. *Bracketing Guidelines for Treebank II Style. Penn Treebank Project*. Technical report, University of Pennsylvania.
- [2] Bosch S., K.-S. Choi, É. Villemonte De La Clergerie, A. Chengyu Fang, G. Faass, K. Lee, A. Pareja-Lora, L. Romary, A. Witt, A. Zeldes, and F. Zipser. 2012. Tiger2 As a standardized serialisation for ISO 24615 - SynAF. Hendrickx, Iris and Kübler, Sandra and Simov, Kiril. *TLT11 - 11th international workshop on Treebanks and Linguistic Theories - 2012*, Nov 2012, Lisbon, Portugal. Edições Colibri, pp. 37-60 — <http://hal.inria.fr/hal-00765413>
- [3] Mengel, A. and Lezius, W. 2000. An XML-based Encoding Format for Syntactically Annotated Corpora. In *Proceedings of the Second International Conference on Language Resources and Engineering (LREC 2000)*. Athens, 2000, pp. 121-126.
- [4] Nilsson J., S. Riedel, and D. Yuret. 2007. "The CoNLL 2007 shared task on dependency parsing", in *Proceedings of the CoNLL shared task session of EMNLP-CoNLL*, pp. 915-932.
- [5] Romary L., A. Zeldes and F. Zipser. 2014. [<Tiger2/> - Serialising the ISO SynAF Syntactic Object Model](http://hal.inria.fr/inria-00612833). *Language Resource and Evaluation* — <http://hal.inria.fr/inria-00612833>
- [6] Romary L., A. Zeldes and F. Zipser. [<Tiger2/> Documentation](https://hal.inria.fr/inria-00593903v2). Technical Report, 2010 — <https://hal.inria.fr/inria-00593903v2>