ELAN Annotation Format

EAF

Schema version: 2.7

(Schema release date: Dec. 2010)

The Language Archive
Max Planck Institute for Psycholinguistics
Nijmegen, The Netherlands

Document last modified: 19 Januari 2012

Table of Contents

Introduction

This document describes the structure of EAF, the ELAN Annotation Format (also known as the EUDICO Annotation Format), on the basis of the main XML elements and attributes of the EAF schema. EAF is the format used for serializing objects that are part of the Abstract Corpus Model [1].
Since it is not possible to capture all types of constraints in a schema, some prose descriptions are added explaining how ELAN [2] interprets certain elements and what assumptions are made when reading an EAF file. These descriptions form the additional requirements that have to be met in order to make a document a valid, editable ELAN document.
A few base constraints are:

- Annotations on the same tier cannot (time-wise) overlap
- A mix of alignable and reference annotations on the same tier is not allowed

1. UML class diagram

Figure 1 shows a (simplified) UML class diagram of the EAF document model (and, since an EAF document is a fairly straightforward serialization of objects in an ELAN transcription, of the ELAN transcription objects). Annotations are contained in Tier objects, a tier can be associated with a parent tier and there are two distinct types of annotations, alignable and reference annotations.

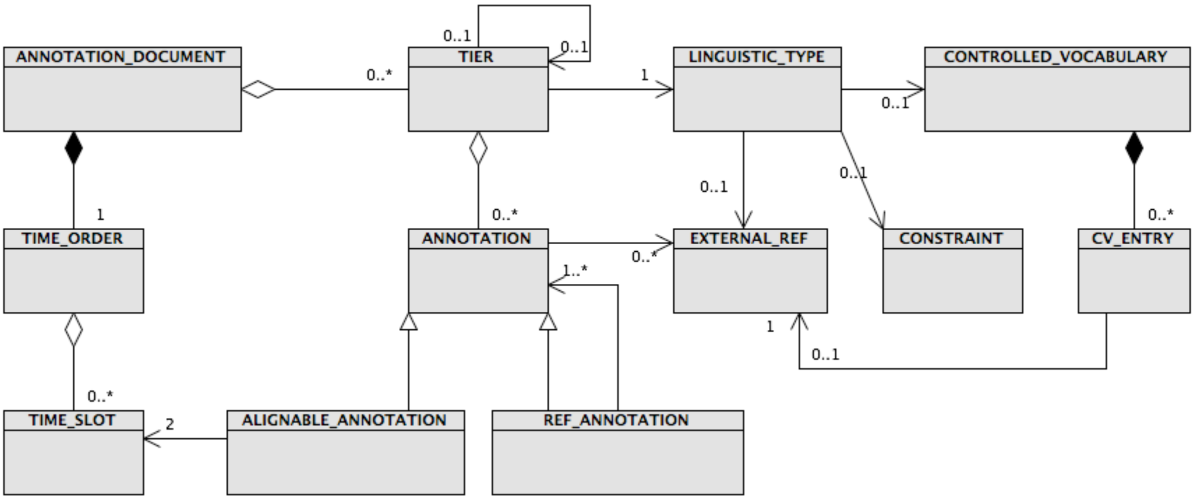

Figure 1 simplified UML class diagram

2. The XML Schema

In the next paragraphs the main elements and attributes of the EAF Schema will be discussed. In some cases the schema snippets are accompanied by EAF fragments that illustrate the usage of the particular element. When the first version of the schema was designed a choice has been made for uppercase element and attribute names.
The schema is defined in an XML Schema Definition file (.xsd), the url of the latest version of the schema is: http://www.mpi.nl/tools/elan/EAFv2.7.xsd

2.1. Element: **ANNOTATION_DOCUMENT**

The root element of an EAF document. It contains most of the other elements and has four attributes:
AUTHOR - the person or program that created the file, required
DATE - the creation date, required
FORMAT - by convention the same as VERSION, optional
VERSION - the version of the schema, required. By default it has the form <major>.<minor>

ANNOTATION_DOCUMENT

```
<xsd:element name="ANNOTATION_DOCUMENT">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="HEADER" type="headType"/>
            <xsd:element name="TIME_ORDER" type="timeType"/>
            <xsd:element name="TIER" type="tierType" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="LINGUISTIC_TYPE" type="lingType" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="LOCALE" type="localeType" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="CONSTRAINT" type="constraintType" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="CONTROLLED_VOCABULARY" type="convocType"
minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="LEXICON_REF" type="lexRefType" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element name="EXTERNAL_REF" type="extRefType" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="DATE" type="xsd:dateTime" use="required"/>
        <xsd:attribute name="AUTHOR" type="xsd:string" use="required"/>
        <xsd:attribute name="VERSION" type="xsd:string" use="required"/>
        <xsd:attribute name="FORMAT" type="xsd:string" use="optional" default="2.7"/>
    </xsd:complexType>
</xsd:element>
```

Example

```
<ANNOTATION_DOCUMENT AUTHOR="ELAN" DATE="2011-05-13T09:47:25+01:00" FORMAT="2.7"
VERSION="2.7" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.mpi.nl/tools/elan/EAFv2.7.xsd">
```

At this level of the schema are also definitions of key-keyref pairs that:

- Ensure that unique names are used there where attributes are not of type xsd:ID but instead of type xsd:string, e.g. TIER_ID is a string (for historic reasons).
- Record to which attribute of another element an attribute of this element refers, e.g. PARENT_REF of TIER refers to a TIER_ID of TIER.

2.2. Element: **HEADER**

There should be exactly one HEADER element. It can contain sequences of three elements and has two attributes:
MEDIA_FILE - the name or path of a media file, optional. This attribute is deprecated and ignored by ELAN. Instead MEDIA_DESCRIPTOR elements can be used
TIME_UNITS - milliseconds or NTSC-frames or PAL-frames, optional, default is milliseconds. ELAN only supports (and assumes) milliseconds

HEADER

```
    <xsd:complexType name="headType">
        <xsd:sequence>
            <xsd:element name="MEDIA_DESCRIPTOR" minOccurs="0"
maxOccurs="unbounded">
.....
            </xsd:element>
            <xsd:element name="LINKED_FILE_DESCRIPTOR" minOccurs="0"
maxOccurs="unbounded">
.....
            </xsd:element>
          <xsd:element name="PROPERTY" type="propType" minOccurs="0"
maxOccurs="unbounded"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="MEDIA_FILE" use="optional" type="xsd:string">
        </xsd:attribute>
        <xsd:attribute name="TIME_UNITS" use="optional" default="milliseconds">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="NTSC-frames"/>
                    <xsd:enumeration value="PAL-frames"/>
                    <xsd:enumeration value="milliseconds"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
```

2.2.1. Element: **MEDIA_DESCRIPTOR**

This element describes one primary media source, typically an audio or video file. It contains information about the location and type of the file and other relevant properties.

MEDIA_DESCRIPTOR

```
<xsd:element name="MEDIA_DESCRIPTOR" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
        <xsd:attribute name="MEDIA_URL" type="xsd:anyURI" use="required"/>
        <xsd:attribute name="RELATIVE_MEDIA_URL" type="xsd:anyURI" use="optional"/>
        <xsd:attribute name="MIME_TYPE" type="xsd:string" use="required"/>
        <xsd:attribute name="TIME_ORIGIN" type="xsd:long" use="optional"/>
        <xsd:attribute name="EXTRACTED_FROM" type="xsd:anyURI" use="optional"/>
    </xsd:complexType>
</xsd:element>
```

2.2.2. Element: **LINKED_FILE_DESCRIPTOR**

This element describes a “secondary”, additional source that is associated with the (media in the) annotation document. As an example timeseries files can be mentioned, as produces by motion capture systems, eye trackers etc.

LINKED_FILE_DESCRIPTOR

```
<xsd:element name="LINKED_FILE_DESCRIPTOR" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
        <xsd:attribute name="LINK_URL" type="xsd:anyURI" use="required"/>
        <xsd:attribute name="RELATIVE_LINK_URL" type="xsd:anyURI" use="optional"/>
        <xsd:attribute name="MIME_TYPE" type="xsd:string" use="required"/>
        <xsd:attribute name="TIME_ORIGIN" type="xsd:long" use="optional"/>
        <xsd:attribute name="ASSOCIATED_WITH" type="xsd:anyURI" use="optional"/>
    </xsd:complexType>
</xsd:element>
```

2.2.3. Element: **PROPERTY**

This is a general purpose element for storing key-value pairs; the NAME attribute is the key, the content of the element the value.

PROPERTY

```
<xsd:element name="PROPERTY" type="propType" minOccurs="0" maxOccurs="unbounded"/>
...
<xsd:complexType name="propType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="NAME" type="xsd:string" use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

2.3. Element: **TIME_ORDER**

The TIME_ORDER element is a container for ordered TIME_SLOT elements. There is always exactly one TIME_ORDER element in a document, which means that the TIME_SLOTs used by annotations of all tiers are stored in a single ordered list.

TIME_ORDER

```
<xsd:element name="TIME_ORDER" type="timeType"/>
.....
<xsd:complexType name="timeType">
    <xsd:sequence>
        <xsd:element name="TIME_SLOT" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:attribute name="TIME_SLOT_ID" type="xsd:ID" use="required"/>
                <xsd:attribute name="TIME_VALUE" type="xsd:unsignedInt"
use="optional"/>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
```

2.3.1. Element: **TIME_SLOT**

A TIME_SLOT indicates a single point in the timeline of the media. The naming of the element is explicit in its intended use (instead of a more generic naming like “anchor”, which is used by several other formats). A TIME_SLOT has two attributes, TIME_SLOT_ID and TIME_VALUE. In principle the interpretation of the value depends on the TIME_UNITS attribute of the HEADER, but ELAN assumes milliseconds for the TIME_VALUE. The TIME_VALUE attribute is optional; this allows a TIME_SLOT to be used as an anchor or a marker in a sequence of partially aligned annotations. It is also possible to have multiple TIME_SLOTs with the same TIME_VALUE. In the case of ELAN, the list of TIME_SLOTs is a straight serialisation of corresponding objects in a loaded transcription; if annotations with the same time alignment are on tiers without inter-dependencies, then multiple TIME_SLOTs with the same TIME_VALUE are written. Likewise, ELAN assumes when loading an EAF file, that a single TIME_SLOT is not referenced by multiple annotations if they don’t depend on each other. This is just a matter of implementation.

Example

```
<TIME_ORDER>
   <TIME_SLOT TIME_SLOT_ID="ts1" TIME_VALUE="610"/>
   <TIME_SLOT TIME_SLOT_ID="ts2" TIME_VALUE="1950"/>
   <TIME_SLOT TIME_SLOT_ID="ts3" TIME_VALUE="1950"/>
   <TIME_SLOT TIME_SLOT_ID="ts4"/>
   <TIME_SLOT TIME_SLOT_ID="ts5"/>
   <TIME_SLOT TIME_SLOT_ID="ts6" TIME_VALUE="8420"/>
</TIME_ORDER>
```

2.4. Element: **TIER**

A TIER is a container for a sequence of ANNOTATIONs. The name of a TIER is used as the TIER_ID. The name of a TIER should be unique within the collection of tier names in the document. Although the TIER_ID is defined as a string, the definition of an “xsd:key” for this attribute ensures that tier names are unique. Other attributes are:
PARTICIPANT - an id of the participant (speaker, signer etc.) to whom the annotations on this tier refer, optional
ANNOTATOR - an id of the annotator or coder for this tier, optional
LINGUISTIC_TYPE_REF - a reference to a type object that defines constraints for this tier

DEFAULT_LOCALE - a reference to a locale element, optional. In ELAN this attribute is used for selecting an input method, it does not necessarily refer to the language spoken by the participant
PARENT_REF - a reference to the TIER_ID of the parent tier, optional

TIER

```
<xsd:element name="TIER" type="tierType" minOccurs="0" maxOccurs="unbounded"/>
.....
<xsd:complexType name="tierType">
        <xsd:sequence>
                <xsd:element name="ANNOTATION" type="annotationType" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="TIER_ID" type="xsd:string" use="required"/>
        <xsd:attribute name="PARTICIPANT" type="xsd:string" use="optional"/>
        <xsd:attribute name="ANNOTATOR" type="xsd:string" use="optional"/>
        <xsd:attribute name="LINGUISTIC_TYPE_REF" type="xsd:string" use="required"/>
        <xsd:attribute name="DEFAULT_LOCALE" type="xsd:IDREF" use="optional"/>
        <xsd:attribute name="PARENT_REF" type="xsd:string" use="optional"/>
</xsd:complexType>
```

2.4.1. Element: **ANNOTATION**

The ANNOTATION element is a container element for either an ALIGNABLE_ANNOTATION or a REF_ANNOTATION. A TIER should not contain a mix of these two types (but this constraint is not enforced by the schema).

ANNOTATION

```
<xsd:element name="ANNOTATION" type="annotationType" minOccurs="0" maxOccurs="unbounded"/>
.....
<xsd:complexType name="annotationType">
        <xsd:choice>
                <xsd:element name="ALIGNABLE_ANNOTATION" type="alignableType"/>
                <xsd:element name="REF_ANNOTATION" type="refAnnoType"/>
        </xsd:choice>
</xsd:complexType>
```

2.4.2. Element: **ALIGNABLE_ANNOTATION**

This element defines an annotation that is associated with a segment of the media by means of references to two TIME_SLOTs. Apart from having an ANNOTATION_ID it has optional references to a graphical svg element and to an EXTERNAL_REF elements. An annotation has one ANNOTATION_VALUE element containing the textual value.

ALIGNABLE_ANNOTATION

```
<xsd:element name="ALIGNABLE_ANNOTATION" type="alignableType"/>
.....
<xsd:complexType name="alignableType">
        <xsd:sequence>
                <xsd:element name="ANNOTATION_VALUE" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="annotationAttribute"/>
        <xsd:attribute name="TIME_SLOT_REF1" type="xsd:IDREF" use="required"/>
        <xsd:attribute name="TIME_SLOT_REF2" type="xsd:IDREF" use="required"/>
        <xsd:attribute name="SVG_REF" type="xsd:string" use="optional"/>
</xsd:complexType>
```

2.4.3. Element: **REF_ANNOTATION**

This element defines an annotation that is not directly associated with a segment of the media (but only indirectly by inheriting alignment of an ancestor annotation). This type of annotation holds a reference to its parent annotation in its required ANNOTATION_REF attribute. Optionally it can have the attribute PREVIOUS_ANNOTATION, a reference to the previous annotation on the same tier having the same parent annotation. This allows for the construction of a chain or sequence of sibling annotations. REF_ANNOTATION also has an ANNOTATION_ID attribute and possibly references to EXTERNAL_REF elements. The annotation value is again an ANNOTATION_VALUE element.

REF_ANNOTATION

```
<xsd:element name="REF_ANNOTATION" type="refAnnoType"/>
.....
<xsd:complexType name="refAnnoType">
        <xsd:sequence>
                <xsd:element name="ANNOTATION_VALUE" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="annotationAttribute"/>
        <xsd:attribute name="ANNOTATION_REF" type="xsd:IDREF" use="required"/>
        <xsd:attribute name="PREVIOUS_ANNOTATION" type="xsd:IDREF" use="optional"/>
</xsd:complexType>
```

2.4.4.Element: **ANNOTATION_VALUE**

This element just holds the textual value (Unicode) of an annotation and has no further attributes.

ANNOTATION_VALUE

```
<xsd:element name="ANNOTATION_VALUE" type="xsd:string"/>
```

2.4.5.Attribute group: **annotationAttribute**

This group contains attributes shared by alignable and reference annotations. It contains an attribute ANNOTATION_ID and the optional EXT_REF attribute which can hold multiple references to EXTERNAL_REF elements.

annotationAttribute

```
<xsd:attributeGroup name="annotationAttribute">
        <xsd:attribute name="ANNOTATION_ID" type="xsd:ID" use="required"/>
        <xsd:attribute name="EXT_REF" type="xsd:IDREFS" use="optional"/>
</xsd:attributeGroup>
```

Following two examples illustrate the usage of the tier and annotation elements.

Example 1: tiers with alignable annotations

```
<TIER TIER_ID="Sp-A" LINGUISTIC_TYPE_REF="speech" ANNOTATOR="AB" PARTICIPANT="A" >
  <ANNOTATION>
    <ALIGNABLE_ANNOTATION ANNOTATION_ID="a2" TIME_SLOT_REF1="ts2"
TIME_SLOT_REF2="ts5">
      <ANNOTATION_VALUE>so it starts out with a rooster crows</ANNOTATION_VALUE>
    </ALIGNABLE_ANNOTATION>
  </ANNOTATION>
  <ANNOTATION>
    <ALIGNABLE_ANNOTATION ANNOTATION_ID="a3" TIME_SLOT_REF1="ts6"
TIME_SLOT_REF2="ts9">
      <ANNOTATION_VALUE>and then you see um a man</ANNOTATION_VALUE>
    </ALIGNABLE_ANNOTATION>
  </ANNOTATION>
</TIER>
...
```

Example 2: tiers with reference annotation

```
<TIER TIER_ID="Motion-A" LINGUISTIC_TYPE_REF="motion" PARENT_REF="Sp-A"
ANNOTATOR="AB" PARTICIPANT="A" >
```

```
   <ANNOTATION>
      <REF_ANNOTATION ANNOTATION_ID="a18" ANNOTATION_REF="a2">
         <ANNOTATION_VALUE>non-motion</ANNOTATION_VALUE>
      </REF_ANNOTATION>
   </ANNOTATION>
   <ANNOTATION>
      <REF_ANNOTATION ANNOTATION_ID="a19" ANNOTATION_REF="a3">
         <ANNOTATION_VALUE>motion</ANNOTATION_VALUE>
      </REF_ANNOTATION>
   </ANNOTATION>
</TIER>
...
```

2.5. Element: **LINGUISTIC_TYPE**

A LINGUISTIC_TYPE object is a collection of attributes and constraints for TIER objects. It is a reusable assembly to which multiple tiers can refer. It is in fact a definition of a type of tier and as such a naming as “TIER_TYPE” would probably be more intuitive. The element LINGUISTIC_TYPE has no content, only attributes:
LINGUISTIC_TYPE_ID - the name and id of the type
TIME_ALIGNABLE - a flag indicating whether this type (and tiers based on it) are time alignable, optional
CONSTRAINTS - a reference to one of the predefined CONSTRAINT elements, optional
GRAPHIC_REFERENCES - a flag that indicates whether annotations on tiers of this type allow references to 2D graphical objects, optional. Not actively used by ELAN.
CONTROLLED_VOCABULARY_REF - a reference to a CONTROLLED_VOCABULARY element, optional
EXT_REF - a reference to an EXTERNAL_REF element that contains a reference to a document external entity
LEXICON_REF - a reference to a LEXICON_REF element which is a collection of attributes identifying and specifying a lexicon service

The value of TIME_ALIGNABLE should be consistent with the referred CONSTRAINTS and the latter should have precedence over the former. The following rules apply:
If there is no CONSTRAINTS attribute alignable should be true. If CONSTRAINTS refers to “Time_Subdivision” or “Included_In” alignable should also be true, for "Symbolic_Subdivision" and “Symbolic_Association” alignable should be false.

LINGUISTIC_TYPE

```
<xsd:element name="LINGUISTIC_TYPE" type="lingType" minOccurs="0" maxOccurs="unbounded"/>
.....
<xsd:complexType name="lingType">
        <xsd:attribute name="LINGUISTIC_TYPE_ID" type="xsd:string" use="required"/>
        <xsd:attribute name="TIME_ALIGNABLE" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="CONSTRAINTS" type="xsd:IDREF" use="optional"/>
        <xsd:attribute name="GRAPHIC_REFERENCES" type="xsd:boolean" use="optional"/>
        <xsd:attribute name="CONTROLLED_VOCABULARY_REF" type="xsd:string" use="optional"/>
        <xsd:attribute name="EXT_REF" type="xsd:IDREF" use="optional"/>
        <xsd:attribute name="LEXICON_REF" type="xsd:IDREF" use="optional"/>
</xsd:complexType>
```

Example

```
<LINGUISTIC_TYPE LINGUISTIC_TYPE_ID="Sp Transcript" TIME_ALIGNABLE="true"
GRAPHIC_REFERENCES="false"/>
<LINGUISTIC_TYPE LINGUISTIC_TYPE_ID="Translation" TIME_ALIGNABLE="false"
GRAPHIC_REFERENCES="false" CONSTRAINTS="Symbolic_Association"/>
CONTROLLED_VOCABULARY_REF="Number"/>
```

2.6. Element: **CONSTRAINT**

A CONSTRAINT element marks the type of constraints that apply to a tier and its annotations. There are four predefined CONSTRAINT variants and ELAN writes all four of them in each EAF document even if they are not used in that document. The element has the following attributes:

STEREOTYPE - the id and also a short description of the general principle of the constraint
DESCRIPTION - a more verbose description of the constraint expressing the rules that apply to annotations and the relations there can be between annotations

CONSTRAINT

```
<xsd:element name="CONSTRAINT" type="constraintType" minOccurs="0" maxOccurs="unbounded"/>
.....
<xsd:complexType name="constraintType">
		<xsd:attribute name="STEREOTYPE" type="xsd:ID" use="required"/>
		<xsd:attribute name="DESCRIPTION" type="xsd:string" use="optional"/>
</xsd:complexType>
```

The constraints are treated as constants and the stereotypes could be part of the schema somehow. ELAN does not support other constraints than those that are predefined, behaviour will be unpredictable in case anything else is used.

The four stereotypes:

```
<CONSTRAINT STEREOTYPE="Time_Subdivision" DESCRIPTION="Time subdivision of parent annotation's time
interval, no time gaps allowed within this interval"/>
<CONSTRAINT STEREOTYPE="Symbolic_Subdivision" DESCRIPTION="Symbolic subdivision of a parent annotation.
Annotations refering to the same parent are ordered"/>
<CONSTRAINT STEREOTYPE="Symbolic_Association" DESCRIPTION="1-1 association with a parent annotation"/>
<CONSTRAINT STEREOTYPE="Included_In" DESCRIPTION="Time alignable annotations within the parent annotation's
time interval, gaps are allowed"/>
```

2.7. Implementation of constraints and tier types in ELAN

Here a summary is given of the different tier types in ELAN and the way the predefined constraints of the schema are applied to the annotations:

Top-level tier: linguistic type without a constraint attribute, annotations are Alignable Annotations, annotations cannot overlap, gaps are allowed and there is no sharing of time slots between annotations on the same tier

Time Subdivision tier: linguistic type with a constraint of stereotype Time_Subdivision, annotations are Alignable Annotations, no gaps allowed, chaining of annotations by sharing of time slots, the first and last child annotations share a time slot with the parent annotation

Included In tier: linguistic type with constraint of stereotype Included_In, annotations are Alignable Annotations, gaps are allowed, no sharing of time slots

Symbolic Subdivision tier: linguistic type with constraint of stereotype Symbolic_Subdivision, annotations are Ref Annotations, reference to a parent annotation required, chaining by reference to previous annotation

Symbolic Association tier: linguistic type with constraint of stereotype Symbolic_Association, annotations are Ref Annotations, reference to a parent annotation is required, maximal one child annotation per parent

Figure 2 illustrates the usage of time slots on top-level tiers, Time Subdivision tiers and Included In tiers. Parent_1 and Parent_2 are top-level tiers, Child_1 is a Time Subdivision dependent tier of Parent_1 and Child_2 is an Included In dependent tier of Parent_2. The image shows how the shared use of time slots on Time Subdivision tiers enforces the constraints for that type of tiers. A corresponding snippet of EAF complements the image.

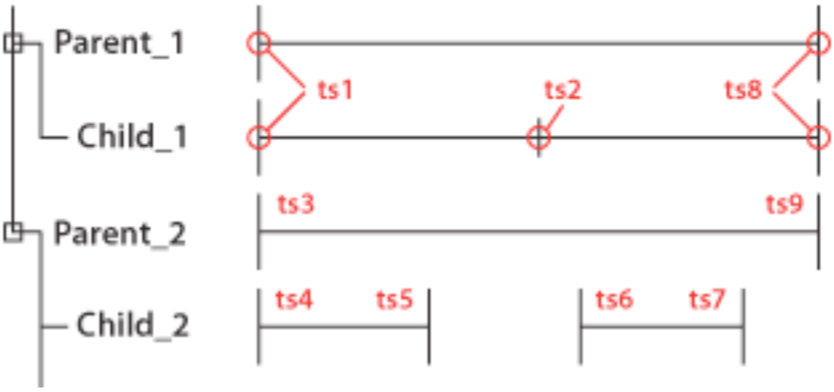

Figure 2 example of time-slot usage in ELAN

Usage of TIME_SLOTs in ELAN

```
  <TIME_ORDER>
    <TIME_SLOT TIME_SLOT_ID="ts1" TIME_VALUE="1000"/>
    <TIME_SLOT TIME_SLOT_ID="ts2"/>
    <TIME_SLOT TIME_SLOT_ID="ts3" TIME_VALUE="1000"/>
    <TIME_SLOT TIME_SLOT_ID="ts4" TIME_VALUE="1000"/>
    <TIME_SLOT TIME_SLOT_ID="ts5" TIME_VALUE="1730"/>
    <TIME_SLOT TIME_SLOT_ID="ts6" TIME_VALUE="2220"/>
    <TIME_SLOT TIME_SLOT_ID="ts7" TIME_VALUE="2840"/>
    <TIME_SLOT TIME_SLOT_ID="ts8" TIME_VALUE="3000"/>
    <TIME_SLOT TIME_SLOT_ID="ts9" TIME_VALUE="3000"/>
  </TIME_ORDER>
```

2.8. Element: **CONTROLLED_VOCABULARY**

A CONTROLLED_VOCABULARY is a container for CV_ENTRY elements. It can be associated with tiers so that the list of entries can be used in the course of the annotation process to reduce typing efforts and improve consistency.
It has the following attributes:
CV_ID - the name and id
DESCRIPTION - a description of the controlled vocabulary
EXT_REF - a reference to an EXTERNAL_REF element representing an externally defined controlled vocabulary

CONTROLLED_VOCABULARY

```
<xsd:element name="CONTROLLED_VOCABULARY" type="convocType" minOccurs="0"
maxOccurs="unbounded"/>
.....
<xsd:complexType name="convocType">
        <xsd:sequence>
                <xsd:element name="CV_ENTRY" type="cventryType" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="CV_ID" type="xsd:string" use="required"/>
        <xsd:attribute name="DESCRIPTION" type="xsd:string" use="optional"/>
        <xsd:attribute name="EXT_REF" type="xsd:IDREF" use="optional">
                <xsd:annotation>
                        <xsd:documentation>
                        A reference to an url of an external Controlled Vocabulary.
                        Is intended to be mutually exclusive with a sequence of CV_ENTRY elements.
                        </xsd:documentation>
                </xsd:annotation>
        </xsd:attribute>
</xsd:complexType>
```

2.9. Element: **CV_ENTRY**

Represents a single entry in a controlled vocabulary. The contents of the CV_ENTRY element is its value. There can be two attributes:
DESCRIPTION - a description of the entry, optional
EXT_REF - a reference to an EXTERNAL_REF element representing e.g. an ISOcat data category, optional

CV_ENTRY

```
<xsd:complexType name="cventryType">
        <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                        <xsd:attribute name="DESCRIPTION" type="xsd:string" use="optional"/>
                        <xsd:attribute name="EXT_REF" type="xsd:IDREF" use="optional"/>
                </xsd:extension>
        </xsd:simpleContent>
</xsd:complexType>
```

Example:

```
<CONTROLLED_VOCABULARY CV_ID="Gesture Hand" DESCRIPTION="Hand gesture codes">
  <CV_ENTRY DESCRIPTION="Right Hand">R</CV_ENTRY>
  <CV_ENTRY DESCRIPTION="Left Hand">L</CV_ENTRY>
  <CV_ENTRY DESCRIPTION="Both Hands">B</CV_ENTRY>
</CONTROLLED_VOCABULARY>
```

2.10. Element: **EXTERNAL_REF**

An EXTERNAL_REF is an element that represents a reference to an external entity. There are a number of predefined entity types as part of the schema.
EXTERNAL_REF has the following attributes:
EXT_REF_ID - the id of the element
TYPE - the type of the external entity, one of the following:

- iso12620, the id of an ISO Data Category
- ecv, an external (closed) controlled vocabulary
- cve_id, reference to the id of an entry in an external Controlled Vocabulary
- lexen_id, reference to the id of a lexical entry
- resource_url, a url or hyperlink to any type of document

VALUE - the value of the element, the interpretation of the value depends on the type

EXTERNAL_REF

```
<xsd:complexType name="extRefType">
        <xsd:attribute name="EXT_REF_ID" type="xsd:ID" use="required"/>
        <xsd:attribute name="TYPE" use="required">
                <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="iso12620">
                                        <xsd:annotation>
                                                <xsd:documentation>
                                        A reference to the id of an ISO Data Category (url including id).
                                                </xsd:documentation>
                                        </xsd:annotation>
                                </xsd:enumeration>
                                <xsd:enumeration value="ecv">
                                        <xsd:annotation>
                                                <xsd:documentation>
                                        A reference to an external (closed) Controlled Vocabulary (url).
                                                </xsd:documentation>
                                        </xsd:annotation>
                                </xsd:enumeration>
                                <xsd:enumeration value="cve_id">
                                        <xsd:annotation>
                                        <xsd:documentation>
                              A reference to the id of an Entry in an external Controlled Vocabulary (id).
                                        </xsd:documentation>
                                    </xsd:annotation>
                                </xsd:enumeration>
                                <xsd:enumeration value="lexen_id">
                                    <xsd:annotation>
                                        <xsd:documentation>
                        A reference to the id of an entry in a lexicon (url, url+id or id)
                                        </xsd:documentation>
                                    </xsd:annotation>
                                </xsd:enumeration>
                                <xsd:enumeration value="resource_url">
                                    <xsd:annotation>
                                        <xsd:documentation>
                                        A reference or hyperlink to any type document (url)
                                        </xsd:documentation>
                                    </xsd:annotation>
                                </xsd:enumeration>
                            <!-- other external reference types can be added later -->
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:attribute>
                    <xsd:attribute name="VALUE" type="xsd:string" use="required"/>
</xsd:complexType>
```

An example with references to an external controlled vocabulary, entries in such vocabularies and to an ISOcat data category.

Example

```
<ANNOTATION>
  <ALIGNABLE_ANNOTATION ANNOTATION_ID="a1" EXT_REF="er1" TIME_SLOT_REF1="ts1"
TIME_SLOT_REF2="ts2">
    <ANNOTATION_VALUE>movement unit</ANNOTATION_VALUE>
  </ALIGNABLE_ANNOTATION>
</ANNOTATION>
<ANNOTATION>
  <ALIGNABLE_ANNOTATION ANNOTATION_ID="a1" EXT_REF="er4" TIME_SLOT_REF1="ts1"
TIME_SLOT_REF2="ts2">
    <ANNOTATION_VALUE>n</ANNOTATION_VALUE>
  </ALIGNABLE_ANNOTATION>
</ANNOTATION>
<CONTROLLED_VOCABULARY CV_ID="Structure" DESCRIPTION="" EXT_REF="er3"/>
<EXTERNAL_REF EXT_REF_ID="er1" TYPE="cve_id" VALUE="CVE_ID40"/>
<EXTERNAL_REF EXT_REF_ID="er2" TYPE="cve_id" VALUE="CVE_ID41"/>
<EXTERNAL_REF EXT_REF_ID="er3" TYPE="ecv"
VALUE="http://www.mpi.nl/tools/elan/atemp/gest.ecv"/>
<EXTERNAL_REF EXT_REF_ID="er4" TYPE="iso12620" VALUE="http://www.isocat.org/datcat/DC-
1333"/>
```

2.11. Element: **LOCALE**

The LOCALE element identifies a locale by a LAGUAGE_CODE attribute optionally combined with a COUNTRY_CODE and a VARIANT attribute. In ELAN LOCALE is not used for specifying the language spoken by a participant but instead for determining which input method (virtual keyboard, lookup list) to use, if any.

LOCALE

```
<xsd:element name="LOCALE" type="localeType" minOccurs="0" maxOccurs="unbounded"/>
```

```
.....
<xsd:complexType name="localeType">
        <xsd:attribute name="LANGUAGE_CODE" type="xsd:ID" use="required"/>
        <xsd:attribute name="COUNTRY_CODE" type="xsd:string" use="optional"/>
        <xsd:attribute name="VARIANT" type="xsd:string" use="optional"/>
</xsd:complexType>
```

Example

```
<LOCALE LANGUAGE_CODE="ru" VARIANT="YAWERTY (Phonetic)"/>
```

2.12. Element: **LEXICON_REF**

The LEXICON_REF is used for specifying a name and location of a lexicon or lexicon service and optionally a specific data category within a lexicon. A linguistic type can be associated with a lexicon, the content of which can then be used in the annotation process.
The following attributes are allowed:
LEX_REF_ID - the internal id
NAME - the display name of this lexicon service
TYPE - the type of the lexicon service
URL - the location of the lexicon or the lexicon service
LEXICON_ID - the identifier of the lexicon as defined by the service
LEXICON_NAME - the name of the lexicon as returned by the service
DATCAT_ID - the identifier of a data category or property of a lexical entry
DATCAT_NAME - the display name of a data category or property of a lexical entry

LEXICON_REF

```
<xsd:element name="LEXICON_REF" type="lexRefType" minOccurs="0" maxOccurs="unbounded"/>
.....
<xsd:complexType name="lexRefType">
        <xsd:attribute name="LEX_REF_ID" type="xsd:ID" use="required"/>
        <xsd:attribute name="NAME" type="xsd:string" use="required"/>
        <xsd:attribute name="TYPE" type="xsd:string" use="required"/>
        <xsd:attribute name="URL" type="xsd:string" use="required"/>
        <xsd:attribute name="LEXICON_ID" type="xsd:string" use="required"/>
        <xsd:attribute name="LEXICON_NAME" type="xsd:string" use="required"/>
        <xsd:attribute name="DATCAT_ID" type="xsd:string" use="optional"/>
        <xsd:attribute name="DATCAT_NAME" type="xsd:string" use="optional"/>
</xsd:complexType>
```

The following example shows how a connection to a LEXUS webservice is stored in a LEXICON_REF element.

Example

```
   <LINGUISTIC_TYPE LEXICON_REF="lr1" LINGUISTIC_TYPE_ID="SL_Lexicon"
TIME_ALIGNABLE="true"/>
.....
<LEXICON_REF DATCAT_ID="MmM5MDkwYTIyZjExMmMyYzAxMmY0NDAyOWJjZDA5ZjU="
DATCAT_NAME="Begripnaam"
LEXICON_ID="MmM5MDkwYTIyZjExMmMyYzAxMmY0NDAyOWMzYTBhMjI="
LEXICON_NAME="SignLinC" LEX_REF_ID="lr1" NAME="SL_Lex1" TYPE="LEXUS (MPI)"
URL="http://corpus1.mpi.nl/mpi/lexusDojo/services/LexusWebService"/>
```

References:

[1] Brugman, H. & Wittenburg, P. (2001). The application of annotation models for the construction of databases and tools. *IRCS Workshop on Linguistic Databases, University of Pennsylvania. Philadelphia.*
[2] Brugman, H. & Russell, A. (2004). Annotating Multi-media / Multi-modal resources with ELAN.
In: *Proceedings of LREC 2004, Fourth International Conference on Language Resources and Evaluation*