Noname manuscript No.
(will be inserted by the editor)

The Linguistic Annotation Framework: A Standard for Annotation Interchange and Merging

Nancy Ide · Keith Suderman

Received: date / Accepted: date

Abstract This paper overviews the International Standards Organization - Linguistic Annotation Framework (ISO - LAF) developed in ISO TC37 SC4. We describe the XML serialization of ISO - LAF, the Graph Annotation Format (GrAF) and discuss the rationale behind the various decisions that were made in determining the standard. We describe the structure of the GrAF headers in detail and provide multiple examples of GrAF representation for text and multi-media. Finally, we discuss the next steps for standardization of interchange formats for linguistic annotations.

Keywords Linguistic annotation · Standards · Language resources · Interoperability

1 Introduction

The Linguistic Annotation Framework (LAF) was developed by the International Standards Organization (ISO)'s TC37 SC4, the ISO sub-committee on Language Resource Management. LAF was the first work item established by the sub-committee in order to provide a broad framework for more specific standards for representing linguistic annotations that have been and continue to be developed in other SC4 working groups. The earliest work on LAF involved identifying the fundamental properties and principles for representing linguistic annotations, and led to the design of an abstract data model that has since served as the basis for SC4 standards for morpho-syntactic and syntactic annotations together with a range of semantic annotation types.

Despite its early start, and while several of the SC4 standards that depend on LAF have been approved and published over the past eight years, LAF has only recently been finalized. However, the overall LAF architecture has not changed since 2001; what has changed is the implementation of a concrete representation format that satisfies

Nancy Ide
Department of Computer Science, Vassar College
Tel.: +1 845 437 5988
Fax: +1 845 437 7498
E-mail: ide@cs.vassar.edu

Keith Suderman
Department of Computer Science, Vassar College

the LAF criteria for expressive adequacy, media independence, flexibility, processability, and–perhaps most critically–mappability to the objects and relations in a variety of formats suitable for different tools and applications. In 2007, the Graph Annotation Format (GrAF) (Ide and Suderman, 2007) was introduced as the final XML serialization of the LAF interchange format; it has since been modified slightly in response to input from experience with full-scale implementation in two multi-layered corpora (OANC[1] and MASC (Ide et al, 2010a)) and implementations for multi-media data, as well as issues that have arisen in the course of developing the ISO standards for specific annotation types. The ISO standard describing LAF and GrAF is published as ISO 24612:2012 (ISO, 2012).

This paper provides an overview of LAF and describes the GrAF XML pivot format, as well as the process and rationale for decisions that fed its final form. For completeness, we provide an outline of the LAF architecture, although this has been described elsewhere (Ide and Romary, 2001, 2003, 2004b, 2007). We describe the structure of the GrAF headers in detail, as this has not been presented elsewhere, and provide multiple examples of GrAF representation for text and multi-media. Finally, we discuss the next steps for standardization of interchange formats for linguistic annotations.

2 Background

2.1 LAF

The motivation for developing LAF is to provide an architecture for annotated language resources that can serve the needs of all the annotation activities in the field of computational linguistics and offer full interoperability among annotation formats. At the time of LAF's initial development, most annotation formats were developed without any underlying data model in mind, and choices were often primarily driven by the needs of particular processing software. Exceptions were the Corpus Encoding Standard (CES, the SGML predecessor of the XML version, the XCES (Ide et al, 2000))[2], which was an early attempt to provide a more principled scheme for linguistic annotation, and which introduced the the concept of "remote markup" (eventually called "standoff markup" (Thompson and McKelvie, 1997)). Later, Annotation Graphs (AG) (Bird and Liberman, 2001), developed primarily for read-only speech data distributed over a timeline, were introduced and subsequently widely adopted in the field. Neither scheme was entirely satisfactory: the XCES was not comprehensive enough for many types of linguistic annotation, and AG posed problems for representing hierarchical relations such as syntactic phrase structure.[3] LAF's development takes these and other established best practices as a starting point for identifying a more comprehensive and general model for representing linguistic annotations.

LAF identifies a set of fundamental principles to inform the development of the architecture. One of the most important is the clear separation of annotation *structure*, i.e., the physical format of annotations, and annotation *content*, which includes the categories or labels used in an annotation scheme to describe linguistic phenomena. A related principle, although seemingly obvious, is the requirement that all annotation

[1] http://www.anc.org/OANC

[2] http://www.cs.vassar.edu/CES/CES1.html

[3] AG was subsequently augmented with *ad hoc* mechanisms to accommodate hierarchical relations, but these were never part of the underlying AG data model.

information be explicitly represented. Some schemes rely on implicit knowledge about particular categories and relations to be interpreted correctly; for example, brackets may signal that the components are a set of alternatives, or that they comprise an ordered list; it is necessary to have knowledge of the categories themselves to determine which applies. This is in itself a major obstacle to interoperability, because processing the annotations often requires the use of specialized software in which this knowledge is embedded.

Based on these principles, the LAF architecture comprises two distinct parts: (1) a data structure for representing relations among annotations, together with a mechanism for associating linguistic categories with appropriate parts of that data structure; and (2) a means to define linguistic categories that is not tied to a specific theory or naming convention. Part (2) ensures semantic coherence; from the outset it was envisaged that this would be provided by a registry of linguistic categories and features that would be universally accessible for reference (Ide and Romary, 2004a). This plan eventually led to the creation of ISOCat (Kemps-Snijders et al, 2009), which effectively became a stand-alone effort. Work on LAF is focused on part 1: the development of an abstract data model for the structure of annotations that could be serialized in a "pivot" XML representation format, into and out of which user-defined formats could be mapped for the purposes of interchange and merging. As a result, LAF has nothing to say about annotation content, *per se*; however, full interoperability for linguistic annotations requires standardization of some organizational practices for interchanging linguistic information that fall in the intersection of representation format and semantic content. See Section 9 for a discussion of next steps for extending LAF to accommodate this need.

The LAF data model must capture the general principles and practices of both existing and foreseen linguistic annotations, including annotations of all media types such as text, audio, video, image, etc. in order to ultimately provide common mechanisms for handling all of them. In addition, the model has to allow for variation in annotation schemes while at the same time enabling comparison and evaluation, merging of different annotations, and development of common tools for creating and using annotated data. To accomplish this, LAF adopts two well-established, generalized data structures: the graph, for representing objects and relations, and feature structures for representing linguistic information. The complete LAF data model ultimately includes (1) a structure for describing media, consisting of *anchors* that reference locations in primary data, and *regions* defined in terms of these anchors; (2) a *graph structure*, consisting of nodes, edges, and links to regions; and (3) an *annotation structure* for representing linguistic information with feature structures. The data model for annotations thus comprises an acyclic di-graph decorated with feature structures (coupled with a moderate admixture of algebra, e.g. disjunction, sets), grounded in n-dimensional regions of primary data. The graph itself is a generalization of models for a wide range of phenomena, including syntax trees, semantic networks, W3C's RDF/OWL, the Unified Modeling Language (UML), entity-relation (ER) models for databases, etc.–not to mention the overall structure of the web, as a dense inter-connected network of effective objects–and grows naturally out of pre-existing annotation models, including Annotation Graphs (Bird and Liberman, 2001) and XML-based formats such as the XCES (Ide et al, 2000). However, LAF differs from other graph-based annotation models in a few significant ways:

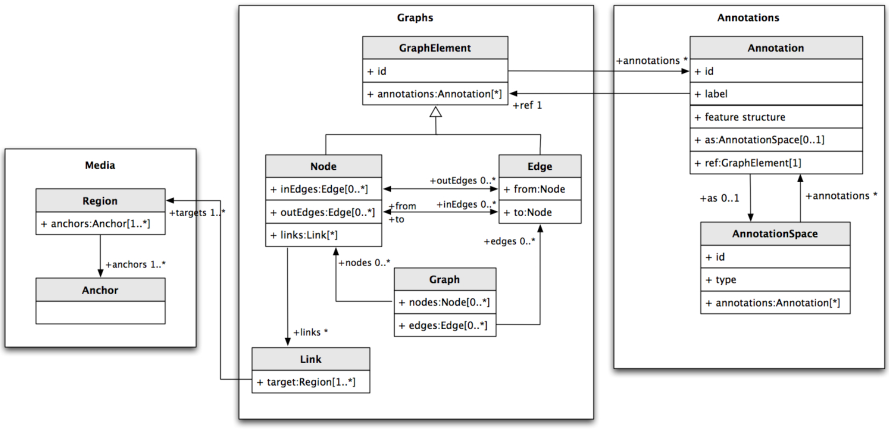

Fig. 1: UML representation of the LAF data model

1. Nodes in the graph do not represent annotations, but rather they are simply place holders that may be associated with zero or more annotations.
2. In addition to connecting nodes (and therefore annotations) via edges to other nodes, a node may be associated with a region or regions in primary data.[4]
3. Edges in the graph are first class citizens of the data model. In many data models the edges between annotations are implied by the nesting of tags (XML, bracket-based schemes such as the Penn Treebank (Marcus et al, 1993) format) or by listing children by reference (W3C DOM, UIMA). In the LAF data model, the edges between annotations are explicitly represented as objects and may also be annotated. Most commonly, annotations on edges specify the structural role of the edge. By default, multiple edges from a single node are assumed to comprise a set of ordered constituents; annotations on the edges can override the default by specifying that the targeted nodes represent a set of alternatives, for example. Similarly, an annotation on an edge can signal that it provides a link to a coreferent, or a temporal link as defined in ISO TimeML (Pustejovsky et al, 2010).

A rendering of the LAF data model in the Unified Modeling Language (UML)[5] is given in Figure 1.

To achieve interoperability among formats while retaining maximal flexibility, LAF prescribes that conformant annotation schemes, either pre-existing or newly developed, are (or may be rendered via mapping) isomorphic to the LAF data model. The mapping between user formats and the LAF abstract data model is via an XML serialization of the data model, called the Graph Annotation Format (GrAF) (Ide and Suderman, 2007). GrAF thus serves as a reference or “pivot” into and out of which annotations may be mapped for interchange, or into which different annotations may be mapped for comparison or merging. We have previously demonstrated the applicability of the model to a wide range of pre-existing annotation types and schemes (Ide and Suder-

[4] Annotation Graphs allow for nodes to be associated with locations in primary data, but not with other nodes in the graph.
[5] http://www.uml.org

man, 2007; Ide et al, 2011) as a proof-of-concept that the model can accommodate a broad range of linguistic annotation types. The Manually Annotated Sub-Corpus (MASC) (Ide et al, 2010a) provides a concrete example where the GrAF rendering enables full interoperability among diverse annotations; at the time of this writing, MASC includes sixteen different annotation types, originally rendered in various formats that have been transduced to a GrAF representation so that they can be searched, retrieved, or otherwise manipulated as a single object. So, for example, a user can extract all annotations of a Penn Treebank NP that includes both a named entity of type *country* and a FrameNet frame element of type *food*.[6]

The overall architecture of a linguistically-annotated resource rendered in GrAF consists of the following:

- One or more *primary data documents*, in any medium[7];
- One or more documents defining a set of regions over each primary data document, each of which may serve as a *base segmentation* for annotations;
- Any number of *annotation documents* containing feature structures associated with nodes and/or edges in a directed graph; all nodes reference either a base segmentation document (in which case the node is a 0-degree node with no outgoing edges) or are connected to other nodes in the same or other annotation documents via outgoing edges;
- *Header documents* associated with each primary data document and annotation document, and a resource header that provides information about the resource as whole.

We describe these components in the following sections. We describe the headers first as they provide information that is relevant for the descriptions of the other components. Note that the full description of GrAF, including GrAF schemas and a description of all components, elements, and attributes, is published as ISO 24612: Language Resource Management – Linguistic Annotation Framework; additional documentation is available at http://www.anc.org/graf.

3 GrAF Headers

Each primary data, segmentation, and annotation document, as well as the resource as a whole, requires a header. The GrAF resource header plays a key role in providing metadata for the resource by establishing resource-wide definitions and relations among files, datatypes, and annotations that can enable automatic validation of the resource file structure and contents. All of the headers have been designed with the aim of facilitating the processing of annotations.

[6] See (Neumann et al, 2013) for a description of the query and visualization tool ANNIS, which enables such queries over MASC data.

[7] The term "document" is applied broadly here to include physical artifacts other than text, and to allow for the possibility that a logical unit of primary data is distributed over multiple computer files.

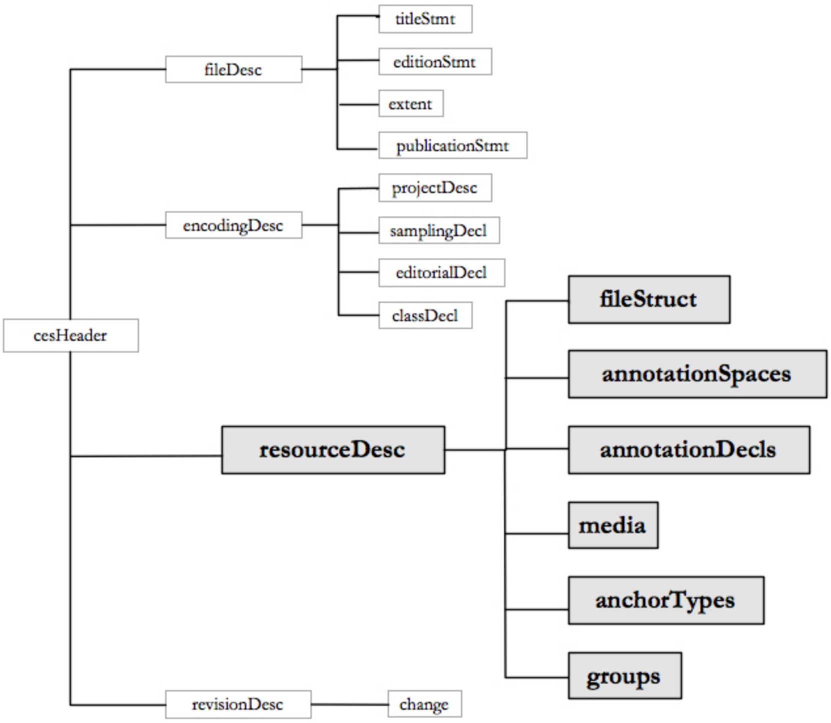

Fig. 2: Main elements of the resourceDesc element in the GrAF resource header.

3.1 Resource header

The GrAF resource header is based on the CES header[8] (which is in turn based on the TEI header[9]), omitting information that is relevant to single documents. The most important addition is a `resourceDesc` (resource description) element, which provides detailed definitions of file naming conventions, annotation types, annotation spaces, encoding specifications, data types, etc. All elements in the `resourceDesc` have an @xml:id attribute, which is used to relate object definitions where applicable. The dependencies among several of these elements are shown graphically in Figure 3, which also shows the use of the @suffix attribute for file types and the @extension attribute for media in a sample file name.

The overall structure of the resource description is shown in Figure 2; the relevant elements are described below.

- **fileStruct**: Provides the file structure of the resource, including the directory structure and the contents of each directory (additional directories and individual files). A set of fileType declarations describe the data files in the resource. Each is associated via attributes with a medium (content type), a set of annotation types, an optional name suffix, an indication of whether or not the file type is required to be

[8] http://www.cs.vassar.edu/CES/CES1-3.html
[9] http://www.tei-c.org/release/doc/tei-p5-doc/en/html/HD.html

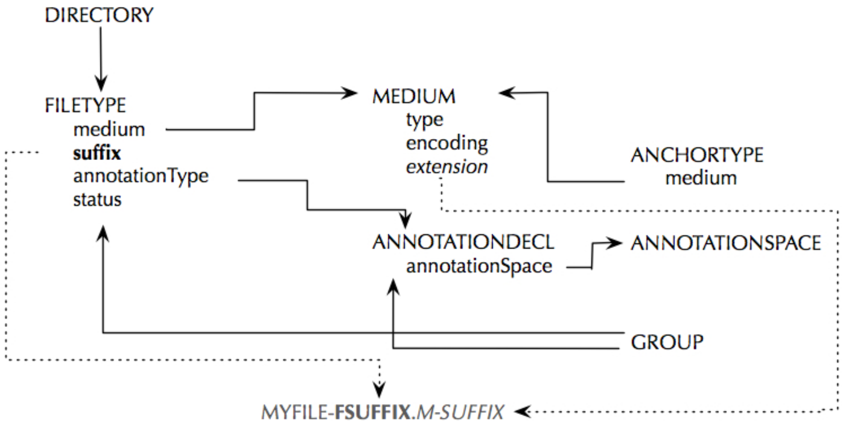

Fig. 3: Dependencies among objects in the resource header

present for each primary data document in the resource, and a list of one or more file types required by this filetype for processing.

- **annotationSpaces**: Provides a set of one or more annotation spaces, which are used in a way similar to XML namespaces. AnnotationSpaces are needed especially when multiple annotations of the same data are merged, to provide context and resolve name conflicts.
- **annotationDecls**: A set of one or more annotation declarations, which provide information about each annotation type included in the resource, including the annotation space it belongs to, a prose description, pointer to the responsible party (creator), the method of creation (automatic, manual, etc.), a pointer to external documentation, of the annotation scheme, and an optional pointer to a schema or schemas providing a formal specification of the annotation scheme.
- **media**: Provides a set of one or more media types that files may contain, the type, encoding (e.g., utf-8), and the file extension used on files containing data of this type.
- **anchorTypes**: a set of one or more types of anchors used to ground annotations in primary data (e.g., character-anchor, time-stamp, line-segment, etc.), the medium with which these anchor types are used, and a pointer to a formal specification of the anchor type.[10] Different anchor types have different definitions and semantics, but all anchors are represented in the same way so that a processor can transform the representation without consulting the definition or having to know the semantics of the representation, which is provided externally by the formal specification.
- **groups**: Definition of one or more groups of annotations that are to be regarded as a logical unit for any purpose. The most common use of groups is to associate annotations that represent a "layer" or "tier", such as a morpho-syntactic or syntactic layer. However, grouping can be applied to virtually any logical set of

[10] Note that all anchor types are associated with one or more media, but a medium is not necessarily associated with an anchor type–in particular, media types associated with documents other than primary data documents (notably, annotation documents) are not associated with an anchor type.

annotations. GrAF provides mechanisms for grouping annotations according to six different criteria:

- *annotation*: annotations with specific values for their labels, as given on the @label attribute of an `a` element in an annotation document, and/or an annotation space. Wildcards may be used to select sets of annotations with common labels and/or in the same annotation space, e.g., in the group member definition `<g.member value="*:tok" type="annotation"/>` the label attribute value **:tok* selects all annotations with the label *tok*, in any annotation space (designated with *:); similarly, the value *xces:** would select all annotations of any type in the xces annotation space.
- *type*: annotations of a specific type or types, by referencing the @xml:id value on an `annotationDecl` element defined in the resource header, e.g., `<g.member value="a.ne" type="type"/>` selects all annotations of type *ne* by referring to the @xml:id value on the prior definition of the *ne* annotation type in the resource header.
- *file*: annotations appearing in a specific file type or types, by referring to the id of a file type defined in the resource header, e.g., `<g.member value="f.entities" type="file"/>` selects all annotations in files conforming to the definition provided by a `fileType` element with the @xml:id value *f.entities*.
- *enumeration*: an enumerated list of ids appearing on `a` elements in a specified annotation document.
- *expression*: an XPath-like[11] expression that can navigate through annotations–for example, the value *@speaker='alice'* would choose all annotations with a feature named *speaker* that has the value *Alice*.
- *group*: refers to a group previously defined. The most common use of this mechanism is to group a set of previously defined enumeration groups; e.g., given these definitions,

```
<group xml:id="g.mygroup">
    <g.member xml:base="myfile1.xml" value="id1 id2 id5"
          type="enumeration"/>
    <g.member xml:base="myfile2.xml" value="id21 id22 id25"
          type="enumeration"/>
</group>
```

  the specification `<g.member value="g.mygroup" type="group"/>` selects the six annotations from annotation documents `myfile1.xml` and `myfile2.xml` whose ids are given above.

Figure 5 provides an example of a groups definition illustrating the grouping mechanisms described above as well as the use of ids for cross-reference among objects defined in the header. It assumes that declarations of the form shown in Figure 4 appear elsewhere in the resource header.

3.2 Primary data document header

The primary document header is stored in a separate XML document with root element `documentHeader`. The document header contains TEI-like elements for describing the

[11] XPath is the XML Path Language defined by W3C; see http://www.w3.org/TR/xpath/

```
<fileType xml:id="f.entities" suffix="ne" a.ids="a.ne"
              medium="xml" requires="f.ptbtok"/>
...
<annotationSpace xml:id="xces" pid="http://www.xces.org/schema/2003"/>
...
<annotationDecl xml:id="a.ne" as="xces">
        <a.desc>named entities</a.desc>
        <a.resp lnk:href="http://www.anc.org">ANC project</a.resp>
        <a.method type="automatic-validated"/>
        <a.doc lnk:href="https://www.anc.org/wiki/wiki/NamedEntities"/>
</annotationDecl>
...
 <medium xml:id="text" type="text/plain" encoding="utf-8" extension="txt"/>
 <medium xml:id="xml" type="text/xml" encoding="utf-8" extension="xml"/>
...
 <anchorType medium="text" default="true"
        lnk:href="http://www.xces.org/ns/GrAF/1.0/#character-anchor"/>
```

Fig. 4: Definitions in the GrAF resource header

```
<groups>
   <group xml:id="g.token">
      <!-- all annotations in any annotation space with label "tok" -->
      <g.member value="*:tok" type="annotation"/>
   </group>
   <group xml:id="g.example">
      <!-- all annotations of type logical -->
      <g.member value="a.logical" type="type"/>
      <!-- all files containing entity annotations -->
      <g.member value="f.entities" type="file"/>
      <!-- all annotations with a feature "speaker" with value "Alice" -->
      <g.member value="@speaker=‘alice’" type="expression"/>
      <!-- annotations with ids "id_1" to "id_n" in file "myfile.xml"-->
      <g.member xml:base="myfile.xml" value="id1 id2 ... idN"
                type="enumeration"/>
      <!-- the annotations included in group g.token, as defined earlier -->
      <g.member value="g.token" type="group"/>
   </group>
</groups>
```

Fig. 5: Group definitions in the GrAF resource header

primary data document, including its title, author, size, source of the original, language and encoding used in the document, etc., as well as a `textClass` element that provides genre/domain information by referring to classes defined in the resource header. Additional elements provide the locations of the primary data document and all associated annotation documents, using either a path relative to the root (declared on a `directory` element in the resource header) or a persistent identifier (PID).

3.3 Annotation documents header

Annotation documents contain both a header and the graph of feature structures comprising the annotation. The annotation document header is brief; it provides four pieces of information:

1. a list of the annotation labels used in the document and their frequencies;
2. a list of documents required to process the annotations, which will include a segmentation document and/or any annotation documents directly referenced in the document;
3. a list of annotationSpaces referenced in the document, one of which may be designated as a default for annotations in the document;
4. (optional) The root node(s) in the graph, when the graph contains one or more graphs that comprise a well-formed tree.

Information about references to other documents is intended for use by processing software, to both validate the resource (ensure all required documents are present) and facilitate the loading of required documents for proper processing. Information about annotation spaces provides a reference to required information in the resource header. When there is more than one tree in a graph, specification of their root nodes is required for proper processing. An example annotation document header is shown in Figure 12.

4 Annotation documents

Following the header, annotation documents contain a graph or graphs and associated annotations. LAF recommends that each annotation type or layer be placed in a separate annotation document, although in the absence of a standard definition of layers it is likely that there will be considerable variation in how this is implemented in practice. A newly-proposed ISO work item will address this and other organization principles in the near future (see Section 9).

GrAF defines the XML serialization of the data model, for which the fundamental data structure is a graph consisting of nodes and edges. An *annotation* is defined as a label and a feature structure that is associated with a node or an edge in the graph. A feature structure is a list of features or nested feature structures, using the XML representation defined in ISO Document ISO/DIS 24610-1(ISO, 2005).

Nodes may be associated with regions in the primary document defined in a base segmentation document, or connected to other nodes in the same or another annotation document by one or more edges. The `node` element is empty when connected by an `edge` element to another node in the graph (i.e., when the node is a non-terminal node). A child `link` element is used when the node refers to a region or regions of primary data (i.e., when the node is a terminal/leaf node).

Annotations associated with a node are represented with `a` elements that appear at the same level in the XML hierarchy, which have a @ref attribute that provides the id of the associated node. The @label attribute on an `a` element gives the main category of the annotation; this may be the string used to identify the annotation as described by the annotation documentation[12], a category identifier from a data

[12] The annotation documentation would be referenced in the annotation type declaration in the resource header.

category registry such as ISOCat, an identifier from a feature structure library, or any PID reference to an external annotation specification. The LAF recommendation is to use PID references to ISOCat categories wherever possible, in order to move toward greater standardization of category definitions.

If the only annotation information is the label, the `a` element is empty. Otherwise, it contains the feature structure or feature structures that provide detailed linguistic information. The ISO specification for representing feature structures allows for feature structures of any complexity and supports the full range of operations over feature structures (subsumption, unification, etc.). It also provides a simplified format that may be used for features consisting of simple name-value pairs, for example (see also Figure 6):

```
<f name="category" value="NP"/>
```

Edges connect two nodes with @from and @to attributes referring to the node ids, and may themselves be labeled with annotations, using the same mechanism described above. By default, edges from a node represent an ordered set of constituents, where the order is determined by the order in which they are defined in the annotation document. Other relationships may be specified by associating an annotation that provides the relational information with the edge, for example, coreference relations (antecedent, etc.) or temporal links. Like any annotation, annotations providing relational information may include a feature structure with more detailed information, as shown in Figure 6.

```
<edge xml:id="tml-e4" from="tml-n1"  to="tml-n2"/>
<a label="TIME-ANCHORING" ref="tml-e4" as="TimeML">
  <fs>
    <f name="relType" value="FOR"/>
  </fs>
</a>
```

Fig. 6: Edge with annotation for a temporal relation

5 Primary data documents

Primary data in a LAF-compliant resource is frozen as read-only to preserve the integrity of references to locations within the document or documents. Thus, a primary data document will contain only the data that is being annotated. Corrections and modifications to the primary data are treated as annotations and stored in a separate annotation document.

In the general case, primary data does not contain markup of any kind. If markup appears in primary data (e.g., HTML or XML tags), it is treated as a part of the data stream by referring annotations; no distinction is made between markup and other characters in the data when referring to locations in the document. Although LAF does not recommend anchoring annotations in primary data by referencing markup, when necessary, XML elements in a document that is valid XML may be referenced by defining a medium type as XML and defining the associated anchor type as an XPath expression. References to locations within these XML elements (i.e., XML element content) can be made using standard offsets, which will be computed by including the markup as part of the data stream; in this case, two media types would be associated with the primary document's file type, as shown in Figure 7.

```
<fileType xml:id="f.primary" medium="text xml"/>
<medium xml:id="text" type="text/plain" encoding="utf-8" extension="txt"/>
<medium xml:id="xml" type="xml" encoding="utf-8" extension="xml"/>
<anchorType medium="xml" default="true"
            lnk:href="http://www.w3.org/TR/xpath20/"/>
<anchorType medium="text"
            lnk:href="http://www.xces.org/ns/GrAF/1.0/#character-anchor"/>
```

Fig. 7: Referencing XML elements in primary data

6 Segmentation: regions and anchors

Segmentation information is specified by defining *regions* over primary data. Regions are defined in terms of *anchors* that directly reference locations in primary data. All anchors are typed; anchor types used in the resource are each defined with an `anchorType` element in the resource header. The type of the anchor determines its semantics and therefore how it should be processed by an application. Figure 8 shows a set of region definitions and the associated anchor type and medium definitions from the resource header.[13]

Anchors are first-class objects the LAF data model (see Figure 1) along with regions, nodes, edges, and links. The anchor is the only object in the model that may be represented in two alternative ways in the GrAF serialization: as a the value of an @anchors attribute on the `region` element, or with an `anchor` element. When anchors are represented with the `anchor` element, the `region` element will include a @refs attribute (and must not include an @anchors attribute) providing the ids of the associated anchors. For example, an alternative representation for region "r2" in Figure 8 is given in Figure 9.

In general, the design of GrAF follows the principle of orthogonality, wherein there is a single means to represent a given phenomenon. The primary reason for allowing alternative representations for anchors is that the proliferation of `anchor` elements in a segmentation document is space-consuming and potentially error-prone. As shown in Figure 8 as well as Section 7, the attribute representation can accommodate most references into text, video, and audio; the only situation in which use of an `anchor` element may be necessary is one where a given location in a document needs to be interpreted in two or more ways, as, for example, a part of two regions that should not be considered to have a common border point. In this case, multiple `anchor` elements can be defined that reference the same location, and each anchor may then be uniquely referenced. Because of its brevity and in the interests of orthogonality, the attribute representation is recommended in LAF.

[13] Note that the @type attribute on the `region` element specifies the anchor type and not the region type.

```
<!-- Definitions in the resource header -->
<medium xml:id="text" type="text/plain" encoding="utf-8" extension="txt"/>
<medium xml:id="audio" type="audio" encoding="MP4" extension="mpg"/>
<medium xml:id="video" type="video" encoding="Cinepak" extension="mov"/>
<medium xml:id="video" type="image" encoding="jpeg" extension="jpg"/>
...
 <anchorType xml:id="text-anchor" medium="text" default="true"
        lnk:href="http://www.xces.org/ns/GrAF/1.0/#character-anchor"/>
<anchorType xml:id="time-slot" medium="audio"
        lnk:href="http://www.xces.org/ns/GrAF/1.0/#audio-anchor"/>
<anchorType xml:id="video-anchor" medium="video"
        lnk:href="http://www.xces.org/ns/GrAF/1.0/#video-anchor"/>
<anchorType xml:id="image-point" medium="image"
        lnk:href="http://www.xces.org/ns/GrAF/1.0/#image-point"/>

<!-- Regions in the segmentation document -->
<region xml:id="r1" anchor_type="time-slot" anchors="980 983"/>
<region xml:id="r2" anchor_type="image-point"
        anchors="10,59 10,173 149,173 149,59"/>
<region xml:id="r3" anchor_type="video-anchor"
        anchors="frame1(10,59) frame2(59,85) frame3(85,102)"/>
<region xml:id="r4" anchor_type="text-anchor"
        anchors="34 42"/>
```

Fig. 8: Region and anchor definitions

```
<anchor xml:id="a1" value="10,59"/>
<anchor xml:id="a2" value="10,173"/>
<anchor xml:id="a3" value="149,173"/>
<anchor xml:id="a4" value="149,59"/>

<region xml:id="r2" refs="a1 a2 a3 a4" anchor_type="image-point"/>
```

Fig. 9: Region and anchor definitions

6.1 Segmentation documents

An annotation document is called a *segmentation document* if it contains only segmentation information–i.e., only `region` and `anchor` elements. Although regions and anchors may also be defined in an annotation document containing the graph of annotations over the data, LAF strongly recommends that when a segmentation is referenced from more than one annotation document, it appears in an independent document in order to avoid a potentially complex jungle of references among annotation documents.

A *base segmentation* for primary data is one that defines minimally granular regions to be used by different annotations, usually annotations of the same type. For example, it is not uncommon that different annotations of the same text–especially annotations created by different projects–are based on different tokenizations. A base segmentation can define a set of regions that include the smallest character span isolated by any of the alternative tokenizations–e.g., for a string such as “three-fold”, regions spanning “three”, ”-”, and ”fold” may be included; a tokenization that regards “three-fold” as a single token can reference all three regions in the @targets attribute on a `link` element associated with the node with which the token annotation is attached, as shown in Figure 10.[14]

```
<region xml:id="seg-r770" anchors="2211 2216"/> <!-- "three" -->
<region xml:id="seg-r771" anchors="2216 2217"/> <!-- "-" -->
<region xml:id="seg-r772" anchors="2217 2221"/> <!-- "fold" -->

<node xml:id="n1019">
        <link targets="seg-r770 seg-r771 seg-r772"/>
</node>
<a label="tok" ref="n1019" as="xces">
        <fs>
            <f name="msd" value="JJ"/>
        </fs>
</a>
```

Fig. 10: Referencing multiple regions

Multiple segmentation documents may be associated with a given primary data document. This is useful when annotations reference very different regions of the data; for example, in addition to the base segmentation document containing the minimal character spans that is partially shown in Figure 10, there may also be a segmentation based on sentences, which may in turn be referenced by annotations for which this unit of reference is more appropriate.[15] Alternative segmentations for different granularities, such as phonetic units, may also be useful for some purposes.

7 Examples

Extensive examples of several types of annotations over text are provided elsewhere (see for example (Ide and Suderman, 2007), (Ide and Bunt, 2010), (Ide et al, 2011)). Here, we provide one example for text together with examples for multi-media.

Figure 11 shows an original FrameNet (Baker et al, 1998) annotation;[16] its GrAF rendering is given in Figure 12. Figure **??** shows a fragment of the associated document defining the tokens referenced in Figure 12. The FrameNet conceptualization specifies a "layer" for each type of information (frame element (FE), grammatical function (GF), phrase type (PT), etc.) in a FrameNet *annotationSet*, that is, a set of annotations for a frame and its slot fillers over a sentence. This requires re-specifying the start and end locations of the annotated region. The GrAF rendering instead groups the elements of an annotation set as children of a node with the annotation label *annotationSet*, which are in turn linked to the tokens defined over the text, as shown graphically in Figure 14.

[14] Note that anchors into character data refer to locations between characters, not to the position of the characters themselves.
[15] Sentences may also be represented as annotations defined over tokens, but for some purposes it is less desirable to consider a sentence as an ordered set of tokens than as a single span of characters.
[16] Some detail concerning the html display has been omitted for brevity.

```
<annotationSet lexUnitRef="11673"  luName="provide.v" frameRef="1346"
         frameName="Supply"  status="MANUAL" ID="2022935">
    <layer rank="1" name="Target">
       <label end="109" start="103"  name="Target"/>
    </layer>
    <layer rank="1" name="FE">
       <label bgColor="0000FF" ... end="138" start="111" name="Recipient"/>
       <label bgColor="FF0000"... end="84"  start="83" name="Supplier"/>
       <label bgColor="FF00FF"... end="79" start="0" name="Means"/>
    </layer>
    <layer rank="1" name="GF">
       <label end="138" start="111"  name="Obj"/>
       <label end="84" start="83"  name="Ext"/>
       <label end="79" start="0"  name="Dep"/>
   </layer>
   <layer rank="1" name="PT">
       <label end="138" start="111"  name="NP"/>
       <label end="84" start="83"  name="NP"/>
       <label end="79" start="0" name="PP"/>
   </layer>
...
</annotationSet>
```

Fig. 11: Original FrameNet standoff annotation in XML

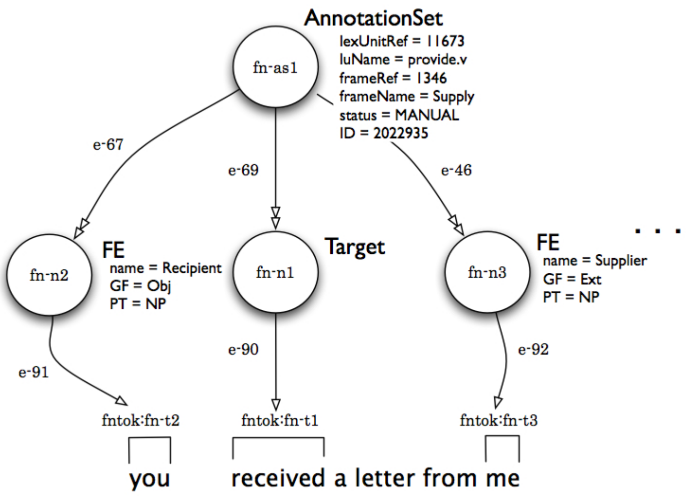

Fig. 14: Visualization of GrAF rendering in Figure 12

The multi-media annotations in Figures 15 and 16 show a segment of gesture annotation as represented in the video and audio annotation tool ELAN[17] and its GrAF rendering. ELAN's internal representation defines *time-slots* that specify a temporal offset (anchor) in the video or audio stream and then defines regions bounded by a start ("time_slot_ref1") and end ("time_slot_ref2") timeslot. This translates naturally into the

[17] http://www.lat-mpi.eu/tools/elan/

GrAF serialization, using anchors as timeslots and regions as "alignable_annotations", and associating the appropriate annotations with nodes that reference these regions.

Figures 17 and 18 similarly show a segment of spatial annotation of video represented using Anvil (Kipp, 2001) and its GrAF rendering. Anvil video anchors may consist of a time (frame) reference and a set of *x, y* coordinates. In the GrAF rendering, the anchor values are given as features of an *element* annotation, rather than being represented as actual GrAF anchors. This is done to remain consistent with the Anvil XML representation, in which the region being annotated and the trajectory are defined using different mechanisms; and in particular to conform to the Anvil data model, which represents a trajectory as an annotation and feature structure, not directly as links into the media. An alternative representation using GrAF regions and anchors similar to the definition for region "r3" in Figure 8 could also be used.

8 GrAF Support Tools and Environment

All GrAF schemas and full documentation of all elements and attributes is available at http://www.anc.org/graf. An API for GrAF is available at http://www.anc.org/graf-api/apidocs/index.html. It provides methods for adding nodes, edges, and annotations to a graph in GrAF format as well as retrieving annotations, features, etc. from the graph. Methods also exist that render annotations in GrAF format in a variety of output formats, such as input to the GraphViz Graph Visualization Software[18].

Two implementations of GrAF in major corpora have been used to inform the GrAF development process, and are freely available via download from the American National Corpus (ANC) website for any use: (1) the Open American National Corpus (OANC)[19] and the Manually Annotated Sub-Corpus (MASC)(Ide et al, 2010a).

The ANC project provides a web application "ANC2Go"(Ide et al, 2010b) that comprises a suite of web services for transducing annotations in GrAF to a variety of other formats, including inline XML (suitable for input to XML-aware software); token / part of speech (with choice of separation character), a common input format for general-purpose concordance software, numerous parsers, and the Natural Language Toolkit (NLTK)[20]; CONLL IOB format, used in the Conference on Natural Language Learning[21] shared tasks; input to the GraphViz[22] graph visualization program, for display of the graphs; and the W3C Resource Description Framework (RDF). The ANC project also provides plugins for the General Architecture for Text Engineering (GATE) (Cunningham et al, 2002) to input and/or output annotations in GrAF format, a CAS Consumer to enable using GrAF annotations in the Unstructured Information Management Architecture (UIMA) (Ferrucci and Lally, 2004), and an NLTK corpus reader. An independent effort within the European project CLARIN[23] has developed a Python implementation of GrAF[24] and an API for mapping data formats used in language documentation into GrAF and back[25] (Blumtritt et al, 2013). Most recently,

[18] http://www.graphviz.org/
[19] http://www.anc.org/OANC
[20] http://nltk.org
[21] http://ifarm.nl/signll/conll/
[22] http://www.graphviz.org
[23] http://www.clarin.eu
[24] https://pypi.python.org/pypi/graf-python/0.3.0
[25] https://poio-api.readthedocs.org/en/latest/

researchers at Universität Potsdam developed a GrAF importer for ANNIS[26] (Zeldes et al, 2009), a powerful corpus query and visualization application (Neumann et al, 2013).

The ANC project has also developed a *GrAF Compact Syntax* (GCS), which represents the information in a GrAF XML serialization as a series of triples. The general format of the GCS is:

```
Regions:
r <id> ["text" | @start @end]  (the region definition may include
                                the text from the document or anchors)
Nodes:
n <id> <region_id> <feature_structure>
Edge:
e <id> <source_id> <target_id>
```

The GCS provides a means to represent the verbose XML representation of GrAF annotations in a compact way. Information and GrAF-to-GCS and GCS-toGrAF converters are available at http://www.anc.org/graf/gcs.

9 Next Steps

LAF and GrAF have been designed to provide a basic scaffolding for linguistic annotations. In principle, GrAF provides no guidelines for naming linguistic categories or organizing or relating specific categories in any way–this principle enabled us to identify and focus on the basic mechanisms required to accommodate the structural and referential properties of these annotations. The result is a generic mechanism that can be used as a pivot for interchanging and combining annotations that has proven to satisfy the many requirements for a Linguistic Annotation Framework outlined in the earliest work on LAF (see for example (Ide and Romary, 2001, 2003, 2004b)).

Complete standardization for linguistic annotation, however, requires much more than the scaffolding that GrAF provides. In addition to standardization of linguistic category semantics, which is the work now being undertaken by ISOCat, it is necessary to establish the inventory and at least a coarse ontology of linguistic objects and features. This is especially urgent in the light of the movement toward building language applications from minimally granular modules, implemented as web services, that provide and ultimately integrate various layers of linguistic annotation. These services must necessarily exchange the same object types and know about the features associated with these objects. As a simple example, the object representing a word with its part of speech could be represented as a "token" object with features "part-of-speech" and "lemma", or as a "noun" object (for example) with a feature "lemma".[27]

Even a simple set of standard linguistic objects has yet to be widely accepted, but it is essential to establish some basis for communication among web services and other language processing tools in order to advance the field. To this end, a new work item has been proposed within ISO TC37 SC4 WG1 to develop at least a basic set of linguistic object/feature descriptors, by working from existing proposals developed or

[26] http://www.sfb632.uni-potsdam.de/annis/

[27] Note that the names of the object and features are much less important than the types of the objects and associated features.

under development in a number of recent projects (e.g., Panacea[28], Language Grid[29], CLARIN[30], etc.), and LAPPS[31], together with best practice in the field as shown in, for example, the design of UIMA type systems. Given that there is now a wide base of recommendations and experience together with increasing convergence of practice, this group should be able to develop at least a basic scheme relatively rapidly that can serve the burgeoning development of modular web services for NLP.

10 Conclusion

This paper provides an overview of the final version of LAF and GrAF, together with a description of the development process that led to the final standard. Despite only recently being finalized, GrAF has already been adopted by many projects, including major European projects such as KYOTO[32], The Australian National Corpus project[33], and several projects in the BioNLP area. Other projects have relied heavily on GrAF to inform development of standards and resources. Even when GrAF is not adopted wholesale, the work on LAF and GrAF has had an enormous impact on the way people think about representing annotation information associated with language data and multi-media. As a result, most if not all newly-developed annotation schemes and formats are based on the LAF abstract data model, and are thus mappable to GrAF–which is in fact all that LAF requires.

Acknowledgments

This work was supported by National Science Foundation grant INT-0753069.

References

Baker CF, Fillmore CJ, Lowe JB (1998) The Berkeley FrameNet project. In: COLING-ACL '98: Proceedings of the Conference, pp 86–90

Bird S, Liberman M (2001) A formal framework for linguistic annotation. Speech Communication 33(1-2):23–60

Blumtritt J, Bouda P, Rau F (2013) Poio API and GraF-XML: A radical stand-off approach in language documentation and language typology. In: Proceedings of Balisage: The Markup Conference 2013, Montreal, Canada, Balisage Series on Markup Technologies, vol 10, DOI 10.4242/BalisageVol10.Bouda01

Cunningham H, Maynard D, Bontcheva K, Tablan V (2002) GATE: A framework and graphical development environment for robust nlp tools and applications. In: Proceedings of ACL'02

[28] http://www.panacea-lr.eu/
[29] http://langrid.nict.go.jp/
[30] http://www.clarin.eu
[31] http://lapps.anc,org
[32] http://www.kyoto-project.eu/
[33] http://www.ausnc.org.au/

Ferrucci D, Lally A (2004) UIMA: An architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering 10(3-4):327–348

Ide N, Bunt H (2010) Anatomy of Annotation Schemes: Mapping to GrAF. In: Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV), Association for Computational Linguistics, Uppsala, Sweden, pp 247–255

Ide N, Romary L (2001) Standards for Language Resources. In: Proceedings of IRCS Workshop on Linguistic Databases

Ide N, Romary L (2003) Outline of the International Standard Linguistic Annotation Framework. In: Proceedings of ACL'03 Workshop on Linguistic Annotation: Getting the Model Right, pp 1–5

Ide N, Romary L (2004a) A Registry of Standard Data Categories for Linguistic Annotation. In: Proceedings of the Fourth International Language Resources and Evaluation Conference (LREC'04), Lisbon, Portugal, pp 135–138

Ide N, Romary L (2004b) International Standard for a Linguistic Annotation Framework. Journal of Natural Language Engineering 10(3-4):211–225

Ide N, Romary L (2007) Towards International Standards for Language Resources. In: Dybkjaer L, Hemsen H, Minker W (eds) Evaluation of Text and Speech Systems, Springer, pp 263–84

Ide N, Suderman K (2007) GrAF: A Graph-based Format for Linguistic Annotations. In: Proceedings of the Linguistic Annotation Workshop (LAW), Association for Computational Linguistics, pp 1–8

Ide N, Bonhomme P, Romary L (2000) XCES: An XML-based encoding standard for linguistic corpora. In: Proceedings of the Second International Language Resources and Evaluation Conference (LREC'00)

Ide N, Baker C, Fellbaum C, Passonneau R (2010a) The manually annotated subcorpus: A community resource for and by the people. In: Proceedings of the ACL 2010 Conference Short Papers, Association for Computational Linguistics, Uppsala, Sweden, pp 68–73

Ide N, Suderman K, Simms B (2010b) ANC2Go: A Web Application for Customized Corpus Creation. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC), Valletta, Malta

Ide N, Prasad R, Joshi A (2011) Towards Interoperability for the Penn Discourse Treebank. In: Proceedings of the Sixth Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation, pp 49–55

ISO (2005) Language Resource Management - Feature Structures - Part 1: Feature Structure Representation. ISO Document ISO/DIS 24610-1

ISO (2012) Language Resource Management - Linguistic Annotation Framework. iso 24612

Kemps-Snijders M, Windhouwer M, Wittenburg P, Wright SE (2009) ISOCat: Remodelling Metadata for Language Resources. International Journal of Metadata, Semantics and Ontologies 4:261–276

Kipp M (2001) ANVIL - A Generic Annotation Tool for Multimodal Dialogue. In: INTERSPEECH'01, pp 1367–1370

Marcus MP, Santorini B, Marcinkiewicz MA (1993) Building a large annotated corpus of english: The penn treebank. Computational Linguistics 19(2):313–330

Neumann A, Ide N, Stede M (2013) Importing MASC into the ANNIS linguistic database: A case study of mapping GrAF. In: Proceedings of the Seventh Linguistic Annotation Workshop (LAW), Sofia, Bulgaria, pp 98–102

Pustejovsky J, Lee K, Bunt H, Romary L (2010) ISO-TimeML: An International Standard for Semantic Annotation. In: Proceedings of the Seventh International Language Resources and Evaluation Conference (LREC'10)

Thompson HS, McKelvie D (1997) Hyperlink semantics for standoff markup of read-only documents. In: Proceedings of SGML Europe 97: The next decade–Pushing the Envelope, pp 227–229

Zeldes A, Ritz J, Lüdeling A, Chiarcos C (2009) ANNIS: A Search Tool for Multi-Layer Annotated Corpora. In: Proceedings of Corpus Linguistics

```
<graph xmlns="http://www.xces.org/ns/GrAF/1.0/">
 <header>
    <labelsDecl>
        <labelUsage label="fullTextAnnotation" occurs="1"/>
        <labelUsage label="Target" occurs="171"/>
        <labelUsage label="FE" occurs="372"/>
        <labelUsage label="sentence" occurs="32"/>
        <labelUsage label="annotationSet" occurs="171"/>
        <labelUsage label="NamedEntity" occurs="32"/>
    </labelsDecl>
    <dependencies>
        <dependsOn file_type.id="fntok"/>
    </dependencies>
    <annotationSpaces>
      <annotationSpace as.id="FrameNet"
            type="http://framenet.icsi.berkeley.edu" default="true"/>
    </annotationSpaces>
 </header>
...
<node xml:id="fn-as1"/>
<a label="annotationSet" ref="fn-as1"  as="FrameNet">
 <fs>
  <f name="lexUnitRef" value="11673"/>
  <f name="luName" value="provide.v"/>
  <f name="frameRef" value="1346"/>
  <f name="frameName" value="Supply"/>
  <f name="status" value="MANUAL"/>
  <f name="ID" value="2022935"/>
 </fs>
</a>

<node xml:id="fn-n1"/>
<a label="Target" ref="fn-n1"  as="FrameNet"/>

<edge xml:id="e69" from="fn-as1"  to="fn-n1"/>
<edge xml:id="e90" from="fn-n1"  to="fntok:fn-t1"/>
<!-- ids fntok:fn-t1 - t4 refer to nodes in the associated tokenization file,
       partially shown in Figure 13 -->

<node xml:id="fn-n2"/>
<a label="FE" ref="fn-n2"  as="FrameNet">
 <fs>
  <f name="name" value="Recipient"/>
  <f name="GF" value="Obj"/>
  <f name="PT" value="NP"/>
 </fs>
</a>
<edge xml:id="e67" from="fn-as1"  to="fn-n2"/>
<edge xml:id="e91" from="fn-n2"  to="fntok:fn-t2"/>

<node xml:id="fn-n3"/>
<a label="FE" ref="fn-n3"  as="FrameNet">
<fs>
 <f name="name" value="Supplier"/>
 <f name="GF" value="Ext"/>
 <f name="PT" value="NP"/>
</fs>
</a>
<edge xml:id="e46" from="fn-as1"  to="fn-n3"/>
<edge xml:id="e92" from="fn-n3"  to="fntok:fn-t3"/>

<node xml:id="fn-n4"/>
<a label="FE" ref="fn-n4"  as="FrameNet">
 <fs>
  <f name="name" value="Means"/>
  <f name="GF" value="Dep"/>
  <f name="PT" value="PP"/>
 </fs>
</a>
<edge xml:id="e10" from="fn-as1"  to="fn-n4"/>
<edge xml:id="e93" from="fn-n4"  to="fntok:fn-t4"/>
```

Fig. 12: GrAF rendering of FrameNet example in Figure 11

```
<!-- A token node and its annotation in the associated ‘‘fntok" file -->
<node xml:id="fn-t1">
   <!-- seg-r14 is a region defined in the base segmentation file,
         covering the token ‘‘received" -->
   <link targets="seg-r14"/>
</node>
 <a label="tok" ref="fn-n10" as="FrameNet"
    <fs>
      <f name="msd" value="VVD"/>
   </fs>
 </a>

<!-- The region definition in the base segmentation file -->
<region xml:id="seg-r14" anchors="73 77"/>
```

Fig. 13: A token node referenced in Figure 12 and its associated region definition

```
<TIME_SLOT TIME_SLOT_ID="ts1" TIME_VALUE="980"/>
<TIME_SLOT TIME_SLOT_ID="ts3" TIME_VALUE="993"/>
<TIME_SLOT TIME_SLOT_ID="ts183" TIME_VALUE="9190"/>

<ANNOTATION>
   <ALIGNABLE_ANNOTATION ANNOTATION_ID="a232" TIME_SLOT_REF1="ts1"
           TIME_SLOT_REF2="ts183">
       <ANNOTATION_VALUE>R Gesture Unit 1</ANNOTATION_VALUE>
   </ALIGNABLE_ANNOTATION>
</ANNOTATION>
<ANNOTATION>
   <ALIGNABLE_ANNOTATION ANNOTATION_ID="a233" TIME_SLOT_REF1="ts1"
           TIME_SLOT_REF2="ts3">
       <ANNOTATION_VALUE>preparation</ANNOTATION_VALUE>
   </ALIGNABLE_ANNOTATION>
</ANNOTATION>
```

Fig. 15: Original ELAN annotation for gesture

```
<!-- Segment of gesture annotation (based on ELAN) -->

<region xml:id="r1" anchors="980 9190"/>
<region xml:id="r2" anchors="980 993"/>
<!-- Each anchor corresponds to an ELAN timeslot -->

<node xml:id="a232">
   <link targets="r1"/>
</node>

<node xml:id="a233">
   <link targets="r2"/>
</node>

<a label="R Gesture Units 1" ref="a232"/>
<a label="preparation" ref="a233"/>
```

Fig. 16: GrAF rendering of ELAN annotation in Figure 15

```
<track-spec name="points" type="primarypoint">
   <attribute name="spatial" valuetype="TimestampedPoints"/>
</track-spec>

<track name="points" type="primary">
     <el index="0" start="1.56" end="1.6">
          <attribute name="traj">
               <point time="1.6" x="698" y="411" />
               <point time="1.6" x="673" y="382" />
               <point time="1.6" x="718" y="379" />
               <point time="1.6" x="684" y="431" />
               <point time="1.6" x="717" y="426" />
         </attribute>
     ...
```

Fig. 17: Original Anvil annotation

```
<!-- Segment of video/spatial annotation (based on Anvil) -->
<region xml:id="r1" anchors="1.56 1.60"/>
<!-- Each anchor corresponds to an Anvil TimeStampedPoint -->

<node xml:id="element-node">
<link targets="r1"/>
</node>
<a xml:id="a1" ref="element-node" label="element" as="anvil">
   <fs>
      <f name="index" value="0"/>
      <f name="traj">
          <fs>
              <f name="point">
                 <fs>
                    <f name="time" value="1.6"/>
                    <f name="x" value="698"/>
                    <f name="y" value="411"/>
                 </fs>
              </f>
              <f name="point">
                 <fs>
                    <f name="time" value="1.6"/>
                    <f name="x" value="673"/>
                    <f name="y" value="382"/>
                 </fs>
                 . . .
</a>

<node xml:id="track-node"/>
<a xml:id="a2" ref="track-node" label="track" as="anvil">
   <fs>
      <f name="type" value="primary"/>
   </fs>
...
</a>

<edge xml:id="e1" from="track-node" to="element-node"/>
```

Fig. 18: GrAF rendering of Anvil annotation in Figure 17