A standardized general framework for encoding and exchange of corpus annotations: the Linguistic Annotation Framework, LAF

Kerstin Eckart
Institut für maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart, Germany
eckartkn@ims.uni-stuttgart.de

Abstract

The Linguistic Annotation Framework, LAF, proposes a generic data model for exchange of linguistic annotations and has recently become an ISO standard (ISO 24612:2012). This paper describes some aspects of LAF, its XML-serialization GrAF and some use-cases related to the framework. While GrAF has already been used as exchange format for corpora with several annotation layers, such as MASC and OANC[1] the generic LAF data model also proved useful as the basis for the design of data structures for a relational database management system.

1 Introduction

Language data and their annotation have been approached from various perspectives. Some approaches are characterized by a layer-wise architecture, where different layers may build on each other, such as morphological, prosodical, syntactic or semantic layers of information. Other approaches concentrate on one specific layer but from various linguistic view points, such as different syntactic frameworks, e.g. dependency and constituency structures. To make reference to these approaches, we speak of vertical vs. horizontal approaches.

In practical work with these approaches, it is often important to pass on, combine or compare annotations for further processing or conjoint querying of different resources. Mostly this includes a lot of work on converting annotation encodings, as processing tools and query engines expect a specific input format and representation. Each of these proprietary formats is useful and adapted for its respective environment and should therefore be used when dealing with this environment. However adding a generic intermediate format – or: exchange format – reduces the overhead in conversion work, and draws the attention to tasks regarding content-related differences.

The *Linguistic Annotation Framework*, LAF, is an ISO standard (ISO 24612:2012) designed for such purposes. It has been developed by ISO's Technical Committee 37, TC 37, *Terminology and other language and content resources*, Sub Committee 4, *Language resource management*. LAF specifies an abstract data model for an exchange format to represent different kinds of linguistic annotations, and provides an XML pivot format for this model, the *Graph Annotation Format*, GrAF.

Section 2 discusses some objectives and examples for exchange formats. Sections 3 and 4 introduce the LAF data model and some aspects of its XML-serialization GrAF. Sections 5 and 6 give an overview of some existing examples for the application of the LAF/GrAF framework from recent publications and experience.

Throughout this article we make use of the example sentence (1) from the DIRNDL Corpus of German radio news, cf. Section 6 on DIRNDL.

(1) Die EU-Kommission bezeichnete das Treffen in Rom als Auftakt für einen neuen Dialog zwischen den europäischen Institutionen und der Jugend.
The European Commission characterized

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

the meeting in Rome as a starting point for a new dialogue between the European institutions and the young people.

2 Exchange formats

Some important objectives of an exchange format for linguistic annotations are to be

- *generic*, such that different types of annotations can be mapped onto it;
- *theory-independent*, such that no preference for a linguistic theory is reflected in the data model;
- *human-readable*, i.e. not only machine-readable or in binary code, such that inspection and surface error tracking is facilitated;
- *stand-off*, i.e. annotation layers are represented separately from each other and from the primary data, such that also single annotation layers can be exchanged and the primary data can consist of different media.

Moreover, a generic exchange format also constitutes some kind of *interlingua* or intermediate representation as known from machine translation (Hutchins and Somers, 1992) and compiler design (Aho et al., 2007). By providing a mapping from a source format $S_1 = A_1$ into the exchange format X, there is automatically a conversion from A_1 into every target format T_i for which a mapping from X to T_i exists. Figure 1[2] visualizes this for the representation formats A_i $(i$ in $1..n)$.

Let a converter be a routine that takes input of one format and produces output of another. Then by making use of an interlingua, $2n$ converters are needed to provide mappings between n representation formats. Without the interlingua $n^2 - n$ converters would be needed for the direct mappings between the representation formats.

In fields adjacent to linguistic annotation, standards for general data models or formats for information encoding and exchange have been proposed, such as the encoding guidelines for digital text from the *Text Encoding Initiative*, TEI[3], also used by libraries and museums. Another example is the W3C standard of the *Resource Description Framework*, RDF[4], to represent linked data in the (Semantic) Web. In the fast growing field of Semantic Web and Linked Open Data[5], an RDF realization of LAF has been proposed by Cassidy (2010) and recently, POWLA has been presented by Chiarcos (2012) which links a generic representation of linguistic annotations with the standard query utilities of RDF[6]. Thereby a connection with other linguistics-related resources from the Linked Open Data cloud is enabled.

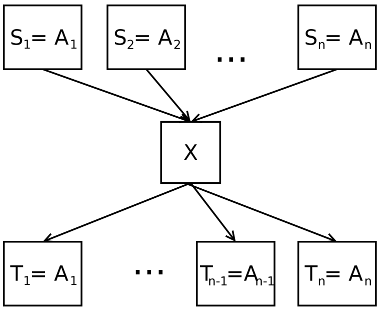

Figure 1: Format conversion with interlingua

Models for the generic representation of linguistic annotations are for example NXT/NITE (Carletta et al., 2003), PAULA (Dipper, 2005), of which the above mentioned POWLA format is a serialization, and Salt, the data model of the converter framework Pepper, cf. (Zipser and Romary, 2010). Also exchange formats for annotations of specific layers have been proposed as ISO standards, such as the SynAF standard ISO 24615:2010, for a *Syntactic Annotation Framework*, the upcoming standard for a *Morpho-syntactic Annotation Framework*, MAF, and the proposals for components of a *Semantic Annotation Framework*, SemAF.

3 The LAF data model

The Linguistic Annotation Framework proposes a generic graph-based data model to represent linguistic annotations. Figure 2 shows the LAF data model and is cited here, with permission of the authors, from (Ide and Suderman, 2012).

The model contains three parts: The annota-

[2] Just another realization of the pictures by Ide and Suderman (2007) and Zipser and Romary (2010).

[3] `http://www.tei-c.org/`

[4] `http://www.w3.org/RDF/`

[5] cf. (Chiarcos et al., 2012) on Linked Data in Linguistics

[6] POWLA also makes use of OWL/DL to introduce controlled vocabulary and constraints for the data model.

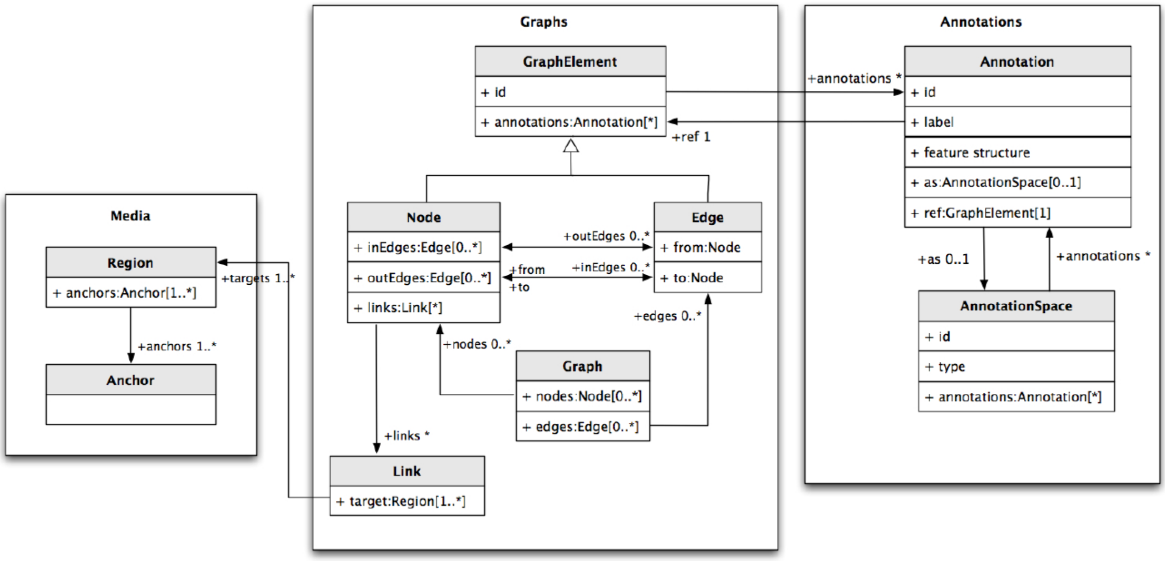

Figure 2: LAF data model by Ide and Suderman (2012), cited with permission

tion part, shown in the rightmost box in Figure 2, which contains data structures to represent annotation labels and complex annotations making use of feature structures[7]. A graphs part, shown in the middle box in Figure 2, allows to represent all annotation structures in a graph based fashion. The graphs consist of nodes and directed edges, where both, nodes and edges can equally be associated with annotations. From the graph structure, links represent the connection to primary data references in the third part which is the media part, shown in the leftmost box in Figure 2.

LAF supports stand-off annotation, which has two effects regarding the data model. Firstly, the primary data is not changed by any annotation. An anchoring mechanism is used to define parts of the primary data to be annotated, i.e. a segmentation which leaves the primary data untouched. The type and encoding of the primary data have to be specified, such that anchors can be interpreted to define virtual positions in between the base units of the encoding.

Let plain text be the medium of sentence (1), UTF-8 its encoding and the sentence the beginning of a primary data set[8], then Figure 3 visualises the anchors in between the characters for the first part of the sentence.

```
|D|i|e| |E|U|-|K|o|m|m|i|s|s|i|o|n|
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
                    1
```

Figure 3: LAF anchoring mechanism

Regions of different granularity can be defined by one or more anchors, e.g. anchors 0 and 3 denote the region *Die*, while anchors 15 and 16 denote the last *o* in *Kommission*. Regions in other media make use of other types of anchors, such as timestamps for audio data, coordinates for image data, etc. The defined regions can be referenced by the links from the annotation graphs, and therefore be related to an annotation.

The second aspect of stand-off annotation is that annotations are layered one on top of the other, making reference to the layers beneath them. In LAF this is represented by edges between nodes belonging to different annotation layers. A node from a syntactic annotation does probably not link the related regions directly, but has edges to one or more nodes from a morpho-syntactic annotation of the respective regions.

LAF also proposes an architecture for corpus resources containing primary data, annotations and metadata, by dividing the data conceptually into different types of documents and headers.

Primary data documents can contain any type of primary data, e.g. text, audio or video data.

[7] cf. (Ide and Suderman, 2012) on `AnnotationSpace`

[8] So the first visualised anchor is 0.

As these documents provide the reference point for the anchors, they must not be changed. Annotation documents contain the annotation graphs and the segmentations related to the primary data. Ide and Suderman (2012) state that creating independent segmentation documents, i.e. annotation documents, that only contain segmentation information, is recommended, when the same segmentation is directly referenced from different annotation documents.

Metadata is represented via headers, i.e. a resource header for the combined resource, a header for each primary data document[9], and a header for each annotation document.

A central concept of LAF is the separation of representation and content. A given annotation content, e.g. a syntactic constituent tree according to a specific linguistic theory or annotation scheme[10], can be represented in an XML format, like TIGER-XML, cf. (König et al., 2003) or in a bracketing format like that of the Penn Treebank, cf. (Bies et al., 1995), while still containing the same annotation content. Therefore LAF separates representation and content, and provides an exchange format with respect to representation. This is also often called syntactic interoperability with respect to different annotation formats. Its complement, semantic interoperability, i.e. the (linguistic) specification and the exchange of data categories and concepts is not in the scope of LAF, but is tackled by other frameworks such as the ISO standard on a Data Category Registry for language resources, ISO 12620:2009 and its implementation ISOcat[11]. Nevertheless LAF provides structures in the resource header and the annotation documents, where reference can be made to the utilized annotation schemes or to single data categories (Ide and Suderman, 2012).

4 The GrAF XML format

The Graph Annotation Format is an XML-serialization of the LAF data model and architecture, and has been proposed as a part of the standard.[12].

GrAF provides XML structures to represent annotations, as well as for headers containing metadata and data on corpus organisation, as presented by Ide and Suderman (2012).

An example of the corpus organisation information is the `fileStruct` element in the resource header, which mirrors the directory structure and indicates the root directory of the resource. Other metadata such as a source description containing the publisher, and text classification elements referring to the domain of the primary data are included in the primary data document header. Relevant for processing is the information on dependencies and anchor types which can be found in the resource header and in the annotation document headers. It states, on which files an annotation document is dependent, i.e. which files have to be present, so that the respective annotation can be correctly processed. The utilized anchor type depends on the medium and encoding of the respective primary data document. The media used in the resource are listed in the resource header and the anchor types are related to the respective medium they apply to. With the LAF anchor mechanism, which abstracts over the different anchor types, anchors in GrAF can be consistently represented, the information on the respective anchor type being accessible via the resource header, cf. Ide and Suderman (2012).

The graph structure specified in LAF is instantiated in GrAF by `node`, `edge` and `link` elements and the annotation structure by `a` elements containing a label and/or a feature structure element `fs`. An annotation element `a` makes reference to an `edge` or `node` element and therefore attaches annotation features to the respective graph element. A contained feature structure can be complex, as defined in ISO 24610-1:2006, the ISO standard on *Feature Structure Representation* (FSR). Edges exist between a start and an end node[13], and nodes which reference regions in the primary data contain `link` elements, stating which regions of the primary data are referenced by the respective node. Primary data regions as defined in the LAF data model, are specified by

[9] Represented as a stand-alone header because the primary data document should not be changed and should only contain primary data.

[10] For an example tree see the unfilled nodes and continuous lines in Figure 8.

[11] `http://www.isocat.org/`

[12] The schema and an API can be found at `http://www.xces.org/ns/GrAF/1.0/`

[13] Therefore GrAF implements directed graphs.

a particular number of anchors and a specific anchor type, both depending on the medium type and encoding of the primary data. In GrAF they are represented by `region` elements. Figures 4 to 7 show parts of a GrAF standoff annotation for sentence (1) and anchors as specified in Figure 3. The same annotation is displayed in Figure 8, by the unfilled and dark gray nodes and the respective edges.

```
<!-- Die -->
<region xml:id="r1" anchors="0 3"/>
<!-- EU -->
<region xml:id="r2" anchors="4 6"/>
<!-- - -->
<region xml:id="r3" anchors="6 7"/>
<!-- Kommission -->
<region xml:id="r4" anchors="7 17"/>
```

Figure 4: Token segmentation

Figure 4 shows a part of a segmentation into regions. Note, that the segmentation does not cover every base unit, as the first whitespace is not part of any region, and that this segmentation is not the most granular one, but fits the annotation.

```
<node xml:id="n1"> <!-- Die -->
 <link targets="r1"/>
</node>
<node xml:id="n2"> <!-- EU-Kommission -->
 <link targets="r2 r3 r4/>
</node>
<a label="tok" ref="n1">
 <fs>
  <f name="word" value="die"/>
  <f name="pos" value="D[std]"/>
  <f name="seq" value="1"/>
 </fs>
</a>
<a label="tok" ref="n2">
 <fs>
  <f name="word" value="EU-Kommission"/>
  <f name="pos" value="N[comm]"/>
  <f name="seq" value="2"/>
 </fs>
</a>
<a>
</a>
```

Figure 5: Nodes with part-of-speech annotation

The first annotation layer which builds on the segmentation from Figure 4 is displayed in Figure 5. Two nodes are shown, where node `n2` references more than one region to represent the token *EU-Kommission*. The annotations refer to these nodes and contain feature structures denoting the part-of-speech of the token (`pos`), the token string (`word`) and the position of the token in the token sequence composing a sentence (`seq`).

```
<node xml:id="n3"/> <!-- NP -->
<edge xml:id="e1" from="n3" to="n2"/>
<node xml:id="n4"/> <!-- DPx[std] -->
<edge xml:id="e2" from="n4" to="n1"/>
<edge xml:id="e3" from="n4" to="n3"/>
<node xml:id="n5"/> <!-- DP[std] -->
<edge xml:id="e4" from="n5" to="n4"/>
<a label="syn" ref="n3">
 <fs>
  <f name="pos" value="NP"/>
 </fs>
</a>
<a label="syn" ref="n4">
 <fs>
  <f name="pos" value="DPx[std]"/>
 </fs>
</a>
<a label="syn" ref="n5">
 <fs>
  <f name="pos" value="DP[std]"/>
 </fs>
</a>
```

Figure 6: Nodes denoting syntactic constituents

The syntactic annotation layer in Figure 6 builds on the token layer in Figure 5, by making reference to the nodes `n2` and `n1` with the edges `e1` and `e2`. Nodes `n3`, `n4` and `n5` denote syntactic constituents according to the German LFG grammar by Rohrer and Forst (2006).

```
<node xml:id="n6"/>
<edge xml:id="e5" from="n6" to="n5"/>
<a label="is" ref="n6">
 <fs>
  <f name="is" value="UNUSED-KNOWN"/>
 </fs>
</a>
```

Figure 7: Node denoting information status of a syntactic phrase

The last layer, exemplified here in Figure 7, is an information status annotation according to Riester et al. (2010). Information status is annotated to a phrase and therefore related to the nodes from Figure 6. The node labeled UNUSED-KNOWN is directly related to node `n5`, which stands for the determiner phrase *Die EU-Kommission*.

5 Use-cases for LAF/GrAF representation and exchange

In this section we give a short overview on examples of use-cases where GrAF has been applied as a representation and exchange format. While some of the examples reference an earlier version of the GrAF schema related to a pre-standard LAF draft, the concepts still apply.

Two multi-layer annotated corpora related to the *American National Corpus*, ANC, have been distributed in the GrAF XML format. The *Open American National Corpus*, OANC[14], which is annotated for part-of-speech as well as noun and verb chunks, and the *Manually Annotated Sub-Corpus*, MASC, cf. (Ide et al., 2010), which is annotated for various annotation layers and also with different annotations for some layers. Both corpora exemplify the corpus architecture supported by GrAF and can be seen as a reference example for GrAF format encoding. Interfaces for ANC data in GrAF exist for frameworks like UIMA and GATE, and the GrAF representation of MASC has been input to a conversion into POWLA (Chiarcos, 2012).

Distiawan and Manurung (2012) proposed a speech recognition web service within the Language Grid framework[15] where the output is encoded in a pre-standard GrAF format aiming at the combination of process interoperability provided by the Language Grid with annotation interoperability provided by the LAF/GrAF framework. Here the primary data is audio or video data and the anchors defining the segments to be annotated therefore are timestamps, providing for flexible annotation/transcription of the primary data. An additional advantage described by Distiawan and Manurung (2012), is that multiple segmentation results can be conjointly represented, which allows that (the n-best) alternative recognition results can be provided to the user. An objective of the described web service approach was also to get information back from the user for retraining the actual recognizer: this setup might encourage the user to provide feedback, as he or she does not have to propose a correction, but just has to choose from alternatives.

Hayashi et al. (2010) present a wrapper architecture for GrAF-based encoding of dependency structures produced by different dependency parsers. The input to a wrapper is therefore a propritary format from a dependency parser, such as the XML formats of the Stanford English parser (de Marneffe and Manning, 2008) or the Japanese dependency parser CaboCha (Kudo and Matsumoto, 2002). The implemented wrappers make use of a sub-schema of GrAF, proposing a general representation for dependency structures, where word-like tokens and phrase-like chunks can be equally parts of a dependency relation. As this proposes a subtyping of GrAF for a specific type of annotation, it is close to annotation-layer-based instantiations such as SynAF; obviously, this work also provided feedback to the respective ISO groups.

6 Use-cases for LAF/GrAF-based data structures

The B3-Database, B3DB, cf. (Eckart et al., 2010) is a database to collect and relate data which occur in the workflow of a linguistic project. This comprises primary data, annotations and information about how the annotations were produced, e.g. taking tool versions or annotation schemes into account. The annotations are stored in the database as objects of manually created information or tool output. Then each of these objects can be decomposed and mapped upon one or more graph structures inside the database. Some important design decisions on annotation handling in the database are the following:

All information is annotation All information from the original annotation file is kept as annotation, e.g. identifiers which are unique within the file or technical data such as processing time.

All annotations are graphs Each annotation is mapped to a set of nodes and edges, or to one of them; sequential annotations, as in prosodic transcriptions, are mapped onto trivial graphs.

Graph representations are static Whenever an error is found in a graph representation, or when it should be changed in any way, the complete graph has to be rebuilt.

The B3DB is implemented as a PostgreSQL[16] relational database system. Its data structures for the graph representations are based on the LAF data model and the respective GrAF structures. In the database, tables are implemented for nodes, edges, links, regions, annotations, feature structures, features and different kinds of name/value pairs. On top of that, nodes, edges and annotations are typed, which helps to distinguish the different types of annotations that exist due to the

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

[15] http://langrid.org

[16] http://www.postgresql.org/

first design decision stated above. A table containing the transitive closure of the graphs helps to facilitate queries on graph structures where basic SQL does not provide for recursion. Here the third design decision comes into play. As the graphs may not be changed dynamically, the closure remains static and avoids overhead from closure updates. That way efficient queries can be processed on the database structures, and a GrAF representation of the graph structures can be easily extracted from the database system.

The database has proven useful for different resources, such as DIRNDL, cf. (Eckart et al., 2012), a German radio news corpus based on two primary data sets. For this corpus there had been two parallel workflows on two closely related primary data sets: a textual version of the radio news, probably the manuscript read by the speakers, and the audio version of the actually spoken news text. The two primary data sets deviated slightly, due to slips of the tongue or minimally modified statements. However, the phonetic processing could only be applied to the primary data based on the audio files, while the syntactic and semantic processing could only be applied to the textual version, cf. (Eckart et al., 2012).

Automatic transcription using forced alignment (Rapp, 1995) was utilized with the audio version, and it was manually annotated with prosodic phrase boundaries and pitch accents according to GToBIS (Mayer, 1995). The textual version was parsed by an LFG-parser[17] with a German grammar by Rohrer and Forst (2006), and the constituency trees were manually annotated for information status according to the annotation scheme by Riester et al. (2010).

To relate and conjointly query the two annotation sets, a linking of the two token layers was semi-automatically created and represented by edges between the two graph representations in the database. Here the LAF edges, which serve as a connection between annotation layers could also bridge the gap between the slightly deviating data sets.

Figure 8 shows an excerpt from the DIRNDL annotations of sentence (1). The information status graph is denoted by the dark gray nodes, the prosody graph by the light gray nodes, and the syntactic constituent tree by the unfilled nodes and the continuous lines[18]. The dashed lines denote edges between the different annotation graphs. While *EU-Kommission* is one token in the syntactic graph, it was divided into two tokens for the prosodic processing[19]. Therefore the two tokens from the prosodic graph are related to the single token of the syntactic graph. While in this case a fine-grained segmentation of the transcription as a common document would have done the same, the cases of slips of the tongue or statement modifications found in the DIRNDL sample make clear that there are actually two primary data sets in the corpus[20].

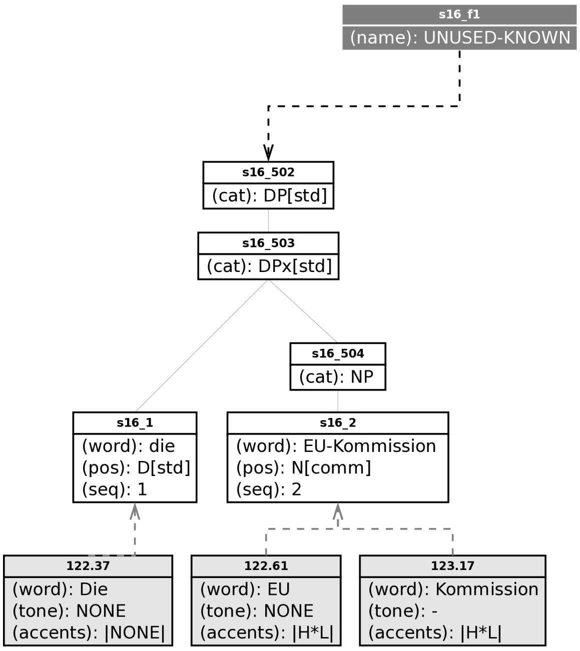

Figure 8: Annotation subgraph for sentence (1) in DIRNDL

The edges in the database are also directed. Nevertheless this has no impact on the query, as the information extraction can follow and edge from source to target or from target to source. Therefore all directions between information status, syntax and prosody can be equally processed.

[17] XLE parser with Lexical Functional Grammar.

[18] DP: determiner phrase, NP: noun phrase; D[std]: determiner, N[comm]: noun; |H*L|: falling accent; -: intermediate phrase boundary

[19] The hyphen is omitted as it is not pronounced.

[20] While one layer could have been marked as the main primary data set and the deviations as a separate annotation layer of 'corrections', we did not want to introduce a weighting wrt the different primary data sets.

As audio data is not directly stored in the database, we decided to represent the timestamps of the prosodic annotation as annotations. As can be seen in Figure 8, each prosody node is annotated with a timestamp denoting the end of the utterance of the respective token in the audio files. Figure 9 however shows both encodings in GrAF, the timestamps as annotations to the phonetic transcription, and as anchors to the database-external audio files.

```
<!-- timestamp annotation -->
<region xml:id="r2" anchors="4 6"/>
<node xml:id="n12">
 <link targets="r2">
</node>
<a label="prosody" ref="n12">
 <fs>
   <f name="timestamp" value="122.61">
   ...
 </fs>
</a>

<!-- timestamp anchors -->
<region xml:id="r52" anchors="122.37 122.61"/>
```

Figure 9: Timestamps as annotations, and as anchors

Other use-cases related to the B3DB have been described by Haselbach et al. (2012) and Eckart et al. (2010). They take among other things different syntactic dependency structures into account. Figure 10 shows abstract dependency relations[21] for the part *Die EU-Kommission bezeichnete das Treffen* of sentence (1).

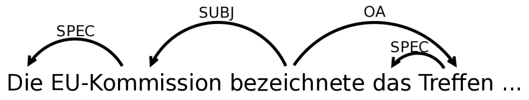

Figure 10: Dependency relations

Note that in the LAF concept of nodes, a node which has a direct link to a region, cannot be the starting point of an edge. So in case no intermediate layer is given, this has to be handled e.g. by introducing additional nodes as start and end points of a dependency relation, cf. Figure 11[22].

Hayashi et al. (2010) propose another representation of dependency structures in GrAF, where an additional dependency node is introduced. This additional node is the start node of two edges, one to the node representing the head and one to the node representing the dependent. The annotation elements of the dependency nodes contain information regarding the dependency relation. Hayashi et al. (2010) argue that a single edge between head and dependent nodes introduces additional semantics by combining the `from` and `to` attributes of an edge with the head and dependent notions from dependency structure. In our understanding, this does not deviate from the usual disposition of GrAF, as edges in a constituency tree also carry semantics, and also some dependency representations tend to introduce edges from dependent to head. However, all these cases can directly be represented in GrAF. Nevertheless the approach of Hayashi et al. (2010) goes beyond the idea of simply representing an existent annotation format in GrAF and aims at a common layer representation for dependency structures. Therefore it is important to choose a representation which makes the common semantics explicit.

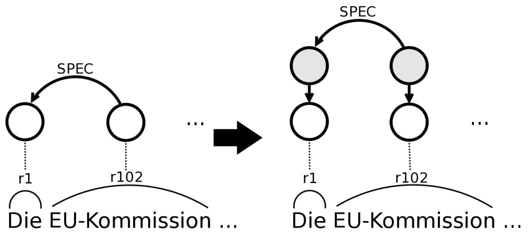

Figure 11: Introducing additional dependency nodes

7 Conclusion

In this paper we reported on the Linguistic Annotation Framework, LAF, and some use-cases of its serialization and data model. Since the early versions of the LAF proposal, the model itself and related use-cases have been discussed and implemented, extensions (Kountz et al., 2008) and derivations (Hayashi et al., 2010) have been proposed and upcoming fields have been included (Cassidy, 2010; Chiarcos, 2012). In 2012 LAF has become an ISO standard. As Chiarcos (2012) concludes, the different but related approaches should be utilized in a complementary fashion, making use of the respective properties and advantages coming with the respective data models and serializations.

[21] SPEC: specifier, SUBJ: subject, OA: object, accusative case

[22] This example applies an alternative segmentation to the one given in the examples in Chapter 4.

An important concept of LAF is that it embodies clear distinctions. Distiawan and Manurung (2012) sum this up very nicely as separating 'user format from exchange format', 'primary data from annotation' and 'representation from content'. In conclusion, one could also add the separation of the 'data model from its serialization', and be prepared for more LAF/GrAF related use-cases to come.

References

Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. 2007. *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison Wesley, August.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre, 1995. *Bracketing Guidelines for Treebank II Style Penn Treebank Project*.

Jean Carletta, Stefan Evert, Ulrich Heid, Jonathan Kilgour, Judy Robertson, and Holger Voormann. 2003. The NITE XML toolkit: Flexible annotation for multimodal language data. *Behavior Research Methods*, 35:353–363.

Steve Cassidy. 2010. An RDF realisation of LAF in the DADA annotation server. In *Proceedings of ISA-5*, Hong Kong, January.

Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann, editors. 2012. *Linked Data in Linguistics. Representing and Connecting Language Data and Language Metadata*. Springer, Heidelberg.

Christian Chiarcos. 2012. A generic formalism to represent linguistic corpora in RDF and OWL/DL. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The stanford typed dependencies representation. In *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*.

Stefanie Dipper. 2005. XML-based stand-off representation and exploitation of multi-level linguistic annotation. In *Berliner XML Tage*, pages 39 – 50.

Bayu Distiawan and Ruli Manurung. 2012. A GrAF-compliant indonesian speech recognition web service on the language grid for transcription crowdsourcing. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 67–74, Jeju, Republic of Korea, July. Association for Computational Linguistics.

Kerstin Eckart, Kurt Eberle, and Ulrich Heid. 2010. An infrastructure for more reliable corpus analysis. In *Proceedings of the Workshop on Web Services and Processing Pipelines in HLT: Tool Evaluation, LR Production and Validation (LREC'10)*, pages 8–14, Valletta, Malta, May.

Kerstin Eckart, Arndt Riester, and Katrin Schweitzer. 2012. A discourse information radio news database for linguistic analysis. In Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann, editors, *Linked Data in Linguistics. Representing and Connecting Language Data and Language Metadata*, pages 65–75. Springer, Heidelberg.

Boris Haselbach, Wolfgang Seeker, and Kerstin Eckart. 2012. German "nach"-particle verbs in semantic theory and corpus data. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May.

Yoshihiko Hayashi, Thierry Declerck, and Chiharu Narawa. 2010. LAF/GrAF-grounded representation of dependency structures. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May.

William J. Hutchins and Harold L. Somers. 1992. *An introduction to machine translation*. Academic Press.

Nancy Ide and Keith Suderman. 2007. GrAF: A graph-based format for linguistic annotations. In *Proceedings of the Linguistic Annotation Workshop*, pages 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.

Nancy Ide and Keith Suderman. 2012. A model for linguistic resource description. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 57–66, Jeju, Republic of Korea, July. Association for Computational Linguistics.

Nancy Ide, Collin Baker, Christiane Fellbaum, and Rebecca Passonneau. 2010. The Manually Annotated Sub-Corpus: A community resource for and by the people. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 68–73, Uppsala, Sweden, July. Association for Computational Linguistics.

Esther König, Wolfgang Lezius, and Holger Voormann, 2003. *TIGERSearch 2.1 User's Manual. Chapter V - The TIGER-XML treebank encoding format*. IMS, Universität Stuttgart.

Manuel Kountz, Ulrich Heid, and Kerstin Eckart. 2008. A LAF/GrAF-based encoding scheme for underspecified representations of dependency structures. In *Proceedings of the 6th Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, May.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of CoNLL-2002*, pages 63–69. Taipei, Taiwan.

Jörg Mayer. 1995. Transcription of German Intonation. The Stuttgart System. ms.

Stefan Rapp. 1995. Automatic phonemic transcription and linguistic annotation from known text with hidden markov models – an aligner for German. In *Proceedings of ELSNET Goes East and IMACS Workshop "Integration of Language and Speech in Academia and Industry" (Russia)*.

Arndt Riester, David Lorenz, and Nina Seemann. 2010. A recursive annotation scheme for referential information sta tus. In *Proceedings of the Seventh International Conference on Languag e Resources and Evaluation (LREC)*, pages 717–722, Valletta, Malta.

Christian Rohrer and Martin Forst. 2006. Improving coverage and parsing quality of a large-scale LFG for German. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy.

Florian Zipser and Laurent Romary. 2010. A model oriented approach to the mapping of annotation formats using standards. In *Proceedings of the Workshop on Language Resource and Language Technology Standards, LREC 2010*, Malta.