An RDF Realisation of LAF in the DADA Annotation Server

Steve Cassidy
Department of Computing
Macquarie University
Steve.Cassidy@mq.edu.au

Abstract

The Linguistic Annotation Framework defines a generalised graph based model for annotation data intended as an interchange format for transfer of annotations between tools. The DADA system uses an RDF based representation of annotation data and provides a web based annotation store. The annotation model in DADA can be seen as an RDF realisation of the LAF model. This paper describes the relationship between the two models and makes some comments on how the standard might be stated in a more format-neutral way.

1 Introduction

The Linguistic Annotation Framework (Ide and Suderman, 2007) is currently being developed as part of the ISO/TC 37/SC 4 standardisation process. LAF and the ISO-GrAF serialisation is designed as an interchange format that can be used for exchange of annotation data between different tool sets. Hence the data structures that it defines are able to be mapped to those of a number of other annotation formats. Similar efforts to define a *lingua franca* for annotations have been made in the past and the design of LAF has a lot in common with, for example, the Annotation Graph model (Bird and Liberman, 2001) and the Emu Speech Database system (Cassidy and Harrington, 2001). Each of these models defines a graph structure of annotations although they differ in what is defined as a node or edge in the graph.

In our recent work on the DADA annotation store (Cassidy and Johnston, 2009) we have been using models based on the Resource Description Framework (RDF) as the basis for an annotation database. RDF is itself a graph based data model used to represent meta-data and structured information in the Semantic Web effort (Berners-Lee et al., 2001). DADA stores annotations as RDF in a dedicated database called a *triple store* and uses semantic web technologies to manipulate and present data. We define an RDF *ontology* to represent annotations; an ontology is a set of object types, properties and relations that together form the data model. DADA is a working system that is being developed as a general purpose store for linguistic annotation data; a demonstration can be viewed at `http://purl.org/dada/demo`.

This paper briefly describes the DADA RDF ontology and how it implements the core data model of ISO-GrAF. In doing so, we make some suggestions about how the specification could be clarified. Finally, we discuss the advantages of using an RDF based data store for representing annotations on language resources.

2 Summary of ISO-GrAF

We include a brief summary of the GrAF Format here for reference but also to ensure that we record the details of the version under discussion since it is at present a draft standard. We refer to the description of GrAF in (rev00, 2008) although we are aware that changes have been made to the proposal since the publication of that draft. Unfortunately the proposal document is somewhat ambiguous and in some cases we have referred to the published source code and examples from the GrAF project for clarification. The examples and description here are hopefully close to the intended structure of GrAF.

GrAF represents a collection of annotations on a single source document as a *graph* which contains a set of *nodes* and *edges*. Each node represents a single annotation, such as the label *noun* applied to a word or a dialogue class label applied to a speaker turn. A special kind of node, called a *span* denotes a region in the source document via *start* and *end* attributes.

A node in the graph can contain a feature structure as defined by the existing ISO standard

(24610-2, 2009). Feature structures are collections of feature-value pairs such that the values of some features can themselves be feature structures. In this way, GrAF is able to represent complex structured annotation values inside the nodes of the graph.

Nodes in the graph are joined by *edges* which can also be associated with feature structures. The specification document is not clear on what such feature structures would be used for and no examples are given. It is suggested that edges might have a *role* (such as ISOTimeML's *tlink* which relates two temporal events with one of a closed set of relations such as *INCLUDES* or *DURING*). The default meaning of an edge between nodes is said to be a link between a container and its constituents.

The ISO-GrAF standard also includes elements such as *nodeSet* and *edgeSet* which can collect together nodes and edges to allow assertions to be made about them.

```
<span id="s1" start="1252" end="1270">
    <as type="biber">
        <fs label="tok">
            <f n="msd" v="cc++++"/>
            <f n="base" v="and"/>
        </fs>
    </as>
</span>
...
<edge id="edge1" from="s1" to="s32">
     <fs>
       <f n="tlink" v="INCLUDES"/>
     </fs>
</edge>
```

Figure 1: An example GrAF XML annotation fragment showing a single span with associated feature structure and an edge connecting this span to another indicating an ISOTimeML INCLUDES relation.

In summary then, ISO-GrAF defines a graph of annotation nodes which contain feature structures linked by edges that may themselves have associated feature structures. The example in Figure 1[1] shows a single span over a region of the source document with an associated feature structure containing two features, *msd* and *base*. The feature structure is contained in an *annotation set* that denotes these features as *biber* part of speech tags. A single edge is shown that links this span with another via the ISOTimeML INCLUDES relation.

3 Mapping GrAF to RDF

To implement an RDF version of GrAF, we need show that the structures in GrAF can be mapped to equivalent structures in RDF.

Nodes in an RDF graph are either *literals* or *resources*. Literals are simple values, strings by default but optionally associated with a data-type such as integer or date. Resources are denoted by URIs and are effectively just atomic names: they have no internal structure. The RDF graph is formed by defining *triples* of *subject-predicate-object* where the subject and object are nodes and the predicate (also called a property) is the named, directed edge connecting them. The subject of a triple must be a resource, while the object can be either a resource or a literal.

RDF triples can be seen as listing the properties of resources (for example, ***this paper*** *has an* ***author*** *of* ***Steve Cassidy***) but because the object of the triple can also be a resource with it's own properties, an arbitrary graph can be built. The RDF structure maps very clearly onto ISO Feature Structures while the graph of connected nodes corresponds closely to the GrAF graph structure.

The main point of difference between GrAF and RDF is the association of properties (feature structures) with edges. GrAF allows this, but RDF edges or properties generally describe classes of property than individual edges. So, one might define a property corresponding to the ISOTimeML *INCLUDES* relation and use it to relate all instances of temporal inclusion. While one could define properties on an individual edge in RDF, we believe that the use of typed edges covers the only demonstrated use cases for descriptions of edges in the graph, and so will be adequate for representing annotation structures.

4 The DADA Ontology

The DADA ontology defines a number of object types and a set of relations that can hold between them. At the high level, it defines the *corpus* as an overall container object and the *annotation set* as a container for all of the annotations on a source media file (or set of files). The usage of annotation set conflicts with that in GrAF (a set of feature sets associated with a span) so we will refrain from using it in the rest of this discussion.

[1] Note that this example does not match the most up to date specification, however, the points of difference are not significant. Since it has been difficult to pin down the correct form, this version is maintained in this paper.

At the lower level, an *annotation* corresponds to the GrAF *node* in that it can refer to a region of the source media and have associated properties and relations. Unlike GrAF, locations in the source media are not represented directly by offsets but are rather stored as separate *anchor* objects. The primary motivation for anchors is to allow annotations to share endpoint – for example, in the case where one segment starts where the previous one ends. This is a common requirement in multi-modal annotation and must be supported if these annotations are to be represented in this data model.

Each of these object types is defined within the DADA namespace (`http://purl.org/dada/schema/0.1#`) and are written in this paper as, for example, `dada:Annotation` which is short for `http://purl.org/dada/schema/0.1#Annotation`. Each of these object types can have arbitrary properties and relations defined from the DADA or other ontologies. DADA properties define the basic structure of annotations; an example is given in Figure 2 which mirrors the GrAF example in Figure 1.

For a single annotation, the RDF structure is relatively simple. An annotation (`:s1`) has `dada:start` and `dada:end` relations to anchors (`anch1` and `anch2`) which denote the left and right boundaries of the annotation. The offset locations of the anchors are defined by relations denoting the units of measurement, `dada:utf16char` in this example meaning an offset in utf16 characters. The data associated with the annotation is encoded by one or more property links, these can either be direct properties of the annotation object or via an intermediate node representing a feature set. In this example we model the *biber* annotation values in the *oanc* namespace illustrating the ability to extend the RDF ontology via new sets of property names and relations.

The INCLUDES relation in the example is stored as a direct link from this annotation node to another. Again, this relation comes from an external ontology, reflecting the ISOTimeML properties and relations.

5 Using External Ontologies

The ability to make use of externally defined ontologies for both properties and relations is a powerful addition to the standardisation of annotation structures in that it facilitates sharing of annotation ontologies between corpora and projects. Currently there is an established *data category registry* (12620, 2008) for terminology associated with linguistic resources and the infrastructure that has been established around that (ISOCat, `http://www.isocat.org/`) is entirely compatible with its use in RDF (Kemps-Snijders et al., 2008). RDF provides established and standardised mechanisms for inclusion and use of such external ontologies along with a mechanism for defining project specific ontologies via OWL schema.

```
:s1 a dada:Annotation;
    oanc:biber [
        oanc:type  "tok" ;
        oanc:msd   "cc++++" ;
        oanc:base  "and" ;
    ] ;
    dada:start :anch1 ;
    dada:end :anch2 ;
    isotime:INCLUDES :s32 .

:anch1 a dada:Anchor;
    dada:utf16char "1252"^^xmls:int .

:anch2 a dada:Anchor;
    dada:utf16char "1270"^^xmls:int .
```

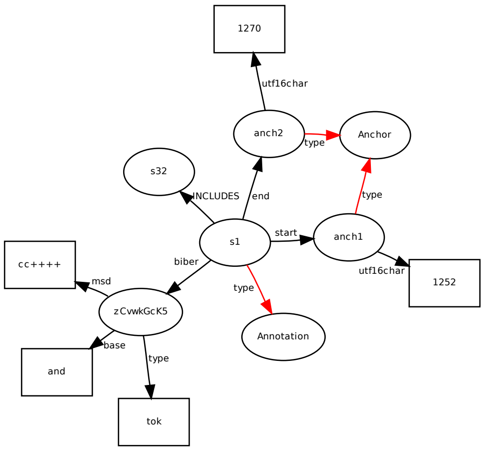

Figure 2: An example annotation structure in RDF in n3 format (top) and in graphical form. Resources (objects) are shown as ellipses while literal values are rectangles.

To give an example of how an external ontology can be integrated into the DADA system we take an example from Kemps-Snijders et al. (2008) which integrates the head word type from the ISOCat registry (Figure 3). Here *headword* is created to reference the ISOCat category DC-258 while *partOfSpeech* is an associated property. The only

```
dcs:headword a dada:AnnotationType;
    dcr:datcat
      <http://isocat.org/datcat/DC-258> ;
    rdfs:label "head word"@en ;
    rdfs:comment "A lemma heading..."@en .

dcs:partOfSpeech a dada:AnnotationType;
    dcr:datcat
       <http://isocat.org/datcat/DC-396> ;
    rdfs:label "part of speech"@en .

:a3 a dada:Annotation;
    dcs:headword [
        dcs:partOfSpeech    "noun";
    ] .
```

Figure 3: An example of defining DADA annotation types referencing the ISOCat data category registry.

addition to the example from the paper is to make both of these an instance of *dada:AnnotationType*. DADA currently does not make the semantic distinction that ISOCat makes between classes and properties, although such a distinction could be supported once the pattern of use by annotation tools is properly understood.

6 Outstanding Issues

One significant issue in the design of the DADA ontology is how annotation types are best represented. There is no clear definition of *type* across the major annotation tools although most offer ways to create different *kinds* of annotation. There is a difference, for example, between word level annotations and part of speech tags, and some tools allow the corpus author to define the way that these are used. We have based much of the design of the DADA type system on the needs of representing multi-modal annotations such as those from ELAN. However, there is a rich set of examples and use-cases in the text annotation world and work developing a clear concept of annotation types is needed before we can progress.

The LAF standard and the GrAF specification are couched in terms of an XML data format. While this format needs to be defined, it would be more useful to have a standard defined in terms of objects and properties and their associated semantics. A specification of a data model that is format-neutral would then allow implementations such as that in DADA to conform to the model even though they do not foreground the XML format. Including a mechanism for making use of externally define vocabularies similar to that in RDF would also facilitate standardisation of these resources. We understand that the most recent drafts of the standards document do move towards this kind of expression of the specification.

7 Summary

We have presented an RDF based annotation data model that implements the core features of the Linguistic Annotation Framework and is functionally compatible with the GrAF interchange format. We argue that the RDF model has a number of advantages including the well defined methods of extending the vocabulary used in the models. The RDF model is implemented in the DADA annotation store that is able to present a web based interface to browsing, querying and updating annotation data.

References

ISO DIS 12620. 2008. Terminology and other language resources data categories specification of data categories and management of a data category registry for language resources. Technical report, International Organization for Standardization, 01.

ISO/DIS 24610-2. 2009. Language resource management – feature structures – part 2: Feature system declaration. Technical report, International Organization for Standardization, Geneva, Switzerland., June.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American*, May. http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html.

S. Bird and M. Liberman. 2001. A Formal Framework for Linguistics Annotation. *Speech Communication*.

S. Cassidy and J. Harrington. 2001. Multi-level Annotation in the Emu Speech Database Management System. *Speech Communication*, 33:61–77.

S. Cassidy and T. Johnston. 2009. Ingesting the Auslan Corpus into the DADA Annotation Store. In *Proceedings of the Third Linguistic Annotation Workshop*, Singapore, July.

N. Ide and K. Suderman. 2007. GrAF: A Graph-based Format for Linguistic Annotations. In *Proceedings of the Linguistic Annotation Workshop, held in conjunction with ACL 2007*, Prague, June. http://www.cs.vassar.edu/~ide/papers/LAW.pdf.

M. Kemps-Snijders, M.A. Windhouwer, and S.E. Wright. 2008. Putting data categories in their semantic context. In *Proceedings of the IEEE e-Humanities Workshop (e-Humanities)*, Indianapolis, Indiana, USA, December.

ISO/TC 37/SC 4 N463 rev00. 2008. Language resource management - linguistic annotation framework. Technical report, International Organization for Standardization, Geneva, Switzerland.