XML Support for Annotated Language Resources

Nancy Ide
Department of Computer Science
Vassar College
Poughkeepsie, New York USA
ide@cs.vassar.edu

Laurent Romary
Equipe Langue et Dialogue
LORIA/CNRS
Vandoeuvre-lès-Nancy, France
romary@loria.fr

LINGUISTIC EXPLORATION
Workshop on Web-Based Language Documentation and Description
Dec 12 - Dec 15, 2000
University of Pennsylvania
Philadelphia, Pennsylvania, USA

Abstract

The XML Corpus Encoding Standard (XCES) is a part of the EAGLES Guidelines developed by the Expert Advisory Group on Language Engineering Standards (EAGLES). XCES is designed to be optimally suited for use in language engineering research and applications, in order to serve as a widely accepted set of encoding standards for corpus-based work in natural language processing applications. The standard specifies a minimal encoding level that corpora must achieve to be considered standardized in terms of descriptive representation (marking of structural and typographic information), provides a suite of DTDs for encoding basic document structure and linguistic annotation, and specifies a corresponding data architecture for linguistic corpora. We are currently extending XCES to support a broad range of annotation types for language data, and providing support for multiple, inter-related annotation levels and co-existence of a multitude of coding schemes and standards. We are also developing "off the shelf" and/or easily modifiable XML support for a broad range of annotation types. XCES is freely available on the web at http://www.xml-ces.org.

The XML Corpus Encoding Standard (XCES)[1] (Ide, *et al.*,2000) is a part of the Guidelines developed by the Expert Advisory Group on Language Engineering Standards (EAGLES)[2]. XCES is designed to be optimally suited for use in language engineering research and applications, in order to serve as a widely accepted set of encoding standards for corpus-based work in natural language processing applications. The standard specifies a minimal encoding level that corpora must achieve to be considered standardized in terms of descriptive representation (marking of structural and typographic information), provides a suite of DTDs for encoding basic document structure and linguistic annotation, and specifies a corresponding data architecture for linguistic corpora. The earlier SGML version of XCES (CES)[3] has been widely adopted by the language processing community (a list of European and US projects using the CES is at http://www.cs.vassar.edu/CES/CES-P.html).

[1] http://www.xml-ces.org
[2] http://www.ilc.pi.cnr.it/EAGLES/home.html
[3] http://www.cs.vassar.edu/CES

We are currently extending XCES to achieve the following:

- support a broad range of annotation types for language data;
- allow multiple annotation levels, where the various annotation levels can be related to each other;
- be open with respect to the information levels and categories within each level;
- allow co-existence of a multitude of coding schemes and standards;
- allow multi-linguality and multi-modality;
- integrate standardization efforts in the US, Europe and Japan;
- provide "off the shelf" and/or easily modifiable XML support for a broad range of annotation types.

The following sections provide an overview of several areas of our work that are relevant to this workshop.

1. Development of an abstract data model for annotations

Reusability of linguistic resources can be achieved only if the data and its annotations are describable using a common data model. The development of XML and related standards, in particular, the XML Schema definition language (Thompson, *et al.*, 2000; Biron and Malhotra, 2000), the Resource Definition Framework (RDF) (Lassila and Swick, 2000), and RDF schemas (Brickley and Guha, 2000), enable description and definition of abstract data models that capture the general form and properties of any object, together with means to interpret, via the model, information encoded using different encoding conventions.

RDF schemas specify the semantics for data based on XML (i.e., objects and their relationships), while XML schemas describe the structure and constrain the contents of XML-encoded documents. RDF definitions are based on well-established data modeling concepts used in diverse areas including knowledge representation (KR), object-oriented design, and database systems, and informs most fundamental data structures in computer science (trees, graphs, etc.), as well models for object-oriented design, database systems (notably, the Entity-Relationship (ER) model [Chen, 1976]). Both XML and RDF schemas rely on a hierarchical class system and offer extensibility through subclass refinement, so that creation of a new XML or RDF schema requires only provision of incremental modifications to the base. Both XML and RDF schemas allow for multiple inheritance to mix definitions, thereby providing multiple views of the data. In addition, one can create instance documents and data based on multiple schemas from multiple sources, thereby "interleaving" different types of annotation.

An abstract model of annotation and a hierarchy of derived types for relevant annotation categories, specifying the semantic roles of the associated data items for each, is defined using RDF schemas.

XML schemas enable document creators to constrain and document the meaning, usage and relationships of the constituent parts of XML documents: datatypes, elements and their content, and attributes and their values. Schemas can also be used to provide default

values for attributes and elements, and/or to specify the data type for these values. XML schemas therefore provide means to define an abstract *structural* model for a class of documents and any number of types, derived types, restricted types, extended types, etc. based on them. The resulting type definition hierarchies provide a powerful tool for describing annotations based on a high-level, general description, at any level of specificity and following precise rules for form and content.

We use XML schemas to instantiate a hierarchically specified structural model for annotations, beginning at the most abstract level and then defining derived types for general classes of annotation (e.g., speech, discourse, morpho-syntax, etc.). Annotation types for different features of each of these types of annotation is defined at the next level of the hierarchy; at the lowest level of the hierarchy, precise annotation values are specified in schemas that can be used "off the shelf" by corpus annotators or modified to suit specific needs. Because types and sub-types are specified in an increasingly precise hierarchy, it is relatively trivial to back up one or more levels of abstraction and define new sub-types. Variant types can also be created from existing ones by defining new derived or extended types.

2. Extension of XCES to additional annotation types.

At present, the XCES provides an XML implementation of the CES's conventions for encoding basic document structure (down to paragraph-level elements), sub-paragraph linguistic structure (sentence, token, named entities, dates, abbreviations, etc.), morpho-syntactic annotation, and alignment among parallel texts, annotations, and primary data. The language processing community has developed additional types of annotation formats since the CES was published in 1996: the Linguistic Data Consortium's annotations page [www.ldc.upenn.edu/annotation] documents the wealth of ongoing activity, the diverse approaches to similar problems, and, conversely, a number of similar approaches to diverse problems. Where common practice and/or a common approach is more or less established, schemas can be provided within XCES, and XSLT scripts can be developed to map between annotation formats (e.g., different POS tagsets). For those areas where common practice is not yet clear, XML schemas can be provided to the relevant level of specificity, and specific annotation formats can be instantiated using this framework.

We are working with various groups to add encoding support (schemas, XSLT scripts for transduction among formats, etc.) to XCES for the following:

- computational lexicons (EAGLES/ISLE, XMELLT)
- discourse and dialogue (MATE, LORIA/CNRS)
- syntactic annotation (several projects)
- speech and its various levels of annotation and representation (ISLE)
- Asian character support (Basis Technology Corp.)
- additional written text features, especially named entities, temporal annotation, etc. (MUC)

3. Creation of a repository of annotation formats and schemas

Because the XML framework provides a powerful retrieval mechanism that enables extraction and transformation of information from one or more XML documents, the use of a precise set of tags for encoding corpora and their annotations has become less critical. The overall aim of XCES is to provide a framework in which annotations can be easily defined (and validated), rather than to dictate the use of specific annotation values, elements, etc. To this end, we have established a repository of existing annotation formats for a variety of linguistic features and, where necessary, and are creating XML schemas to instantiate them together with XSLT scripts to transduce between different formats where appropriate.

XML-encoded annotated corpora are used primarily for interchange--not only interchange between research sites, but also interchange between individual processing and analytic tools. Most corpus-handling tools use one of a variety of internal formats (flat files, database formats, etc.) and import from and export to XML documents. For commonly used tools, XSLT scripts are being developed for mapping, and extraction of annotated data, import/export of (partially) annotated material, and integration of results of external tools into existing annotated data in XML.

4. An example: Syntactic annotation

At its highest level of abstraction, an annotation is a set of data or information (in our case, linguistic information) that is associated with some other data. Typically, the latter is what could be called “primary” data (e.g., a part of a text or speech signal, etc.), but this need not be the case; consider, for example, the alignment of parallel translations, where the "annotation" is a link between two primary data sets (the aligned texts), or syntactic annotation of a text already annotated for part of speech. We can therefore model an annotation as a link associating two undifferentiated data items, each of which may be simple (e.g., a string of characters) or more complex (a set of data points defining some object, or a complex structure with a hierarchy of parts).

In fact, the RDF model is precisely this one. In RDF, all things being described are considered to be *resources*, and resources can be documents, elements or locations within documents, whole web sites, or even objects not accessible via the web such as a book. Furthermore, since a resource is identified by its URI, a resource can be an RDF specification itself. Resources are characterized and related to one another via *properties*, each of which has a specific meaning and a defined set of relations to other properties. The fundamental RDF construct is the RDF *statement*, which is a resource, a named property, and the value of that property. This mirrors the general model of an annotation given above.

The goal in the XCES is to provide a framework for annotation that is theory and tagset independent. For syntactic annotation, this can be accomplished by differentiating the general structure of syntactic annotations and the data category specifications that are used to describe the constituent objects and their relations. In general, syntactic

annotations describe dependencies, for example, modifier/modified, hierarchical grammatical relations represented in syntax trees, etc.

For lack of space, we show here only a simple example of stand-off markup in XCES representing syntactic annotation in the Penn Treebank (Figures 1 and 2).

```
 ((S      (NP-SBJ-1 Jones)
   (VP followed)
          (NP him)
          (PP-DIR       into
                 (NP the front room))
          ,
          (S-ADV (NP-SBJ *-1)
                 (VP    closing
                        (NP the door)
                        (PP    behind
                               (NP him)))))
          .))
```

Figure 1. Penn Treebank in-line annotation for the sentence “Jones followed him into the front room, closing the door behind him.”

```
<chunk xml:base="http://www.loria.fr/doc.xml#">
   <struct id=“s0”>
       <feat type=“CAT”>S</feat>
       <struct id=“s1” xlink:href="xptr(substring(/p/s[1]/text(),1,5))"/>
             <!—Jones -->
             <feat type=“CAT”>NP</feat>
             <rel type=“SBJ” head=“s2”/>
       </struct>
       <struct id=“s2” xlink:href="xptr(substring(/p/s[1]/text(),7,8))"/>
            <!—followed -->
            <feat type=“CAT”>VP</feat>
            <struct xlink:href="xptr(substring(/p/s[2]/text(),16,3))"/>>
                  <!— him -->
                 <feat type=“CAT”>NP</feat>
                 <!-- implicit OBJ relation here -->
             </struct>
             <struct xlink:href="xptr(substring(/p/s[2]/text(),20,4))"/>
                   <!—into -->
                  <feat type=“CAT”>PP</feat>
                  <rel type=“DIR” head=“s2”/>
                  <struct xlink:href=
                            "xptr(substring(/p/s[2]/text(),25,14))"/>
                      <!-- the front room -->
                     <feat type=“CAT”>NP</feat>
                  </struct>
              </struct>
               <struct>
                  <feat type=“CAT”>S</feat>
                  <rel type=“ADV” head=“s2”/>
                  <struct ref=“s1”>
                     <feat type=“CAT”>NP</feat>
                     <rel type=“SBJ” head=“s3”/>
                  </struct>
                  <struct id=“s3” xlink:href=
                      "xptr(substring(/p/s[2]/text(),41,7))"/>
                      <!—closing -->
                     <feat type=“CAT”>VP</feat>
                         <struct>…</struct>
                  </struct>
              </struct>
   …
   </chunk>
```

Figure 2. A possible XCES encoding of the annotation as stand-off markup

Note the following:

- We use a generic `<struct>` element to represent the hierarchical syntactic structure. XML schemas can constrain the types that can be embedded, where appropriate. This provides for implicit marking of dependency relations, where the default is the parent `<struct>` element.
- A generic `<feat>` element is used to specify one or more descriptions of the bracketed segment. Again, schemas can constrain the values of descriptors, allowing, for example, for use of any set of syntactic category labels. Note that instead of explicitly specifying a category, the `<feat>` element could point to a descriptor, possibly including additional information (e.g., number, gender, etc.), in another location.
- We encode explicitly here much information that will ultimately be specified using RDF, which provides built-in support for linkage with labeled relations (links). In particular, information currently specified using the `<feat>` element can be represented as a 3-tuple (resource, property, value) that indicates that the resource (identified by `ID=s1` in the encoding above) has the property `CAT` with value `NP`, follows:

```
<rdf:Description about="http://www.loria.fr/docs/ann1#…">
  <s:Cat>NP</s:Cat>
</rdf:Description>
```

  Alternatively, the `<s:Cat>` element could point to another resource with a fuller description. Similarly, information represented in the encoding above using `<rel>` elements and the *ref* attribute could also be expressed in RDF. Because RDF is still under development and software support is some way off, we provide alternative XML formats in the interim.

We stress that XCES is under development; what we show here is provisional and likely to change as XCES and related standards such as RDF develop. We welcome input on any aspect of our work.

REFERENCES

Biron, P., Malhotra, A., 2000. XML Schema Part 2: Datatypes. W3C Candidate Recommendation, 24 October 2000. http://www.w3.org/TR/xmlschema-2/.

Brickley, D. and Guha, R.V. (2000). Resource Description Framework (RDF) Schema Specification 1.0. W3C Candidate Recommendation, 27 March 2000. http://www.w3.org/TR/rdf-schema/.

Chen. P. (1976). The Entity-Relationship Model--Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1:1, March 1976.

Ide, N., Bonhomme, P., Romary, L. (2000). XCES: An XML-based Standard for Linguistic Corpora.. *Proceedings of the Second Language Resources and Evaluation Conference (LREC)*, Athens, Greece, 825-30.

Lassila, O. and Swick, R. (1999) Resource Description framework (RDF) Model and Syntax Specification. W3C Recommendation, 22 February 1999. http://www.w3.org/TR/REC-rdf-syntax.

Thompson, H., Beech, D., Maloney, M. Mendelsohn, N., 2000. XML Schema Part 1: Structures. W3C Candidate Recommendation, 24 October 2000. http://www.w3.org/TR/xmlschema-1/.