Resources for Lexicalized Tree Adjoining Grammars and XML encoding: TagML

Patrice Bonhomme* and Patrice Lopez[†]

*LORIA BP 239 F-54506 Vandoeuvre-lès-Nancy bonhomme@loria.fr

[†]DFKI GmbH

Stuhlsatzenhausweg 3 D-66123 Saarbrücken lopez@dfki.de

Abstract

This work addresses both practical and theorical purposes for the encoding and the exploitation of linguistic resources for feature based Lexicalized Tree Adjoining grammars (LTAG). The main goals of these specifications are the following ones:

- 1. Define a recommendation by the way of an XML (Bray et al., 1998) DTD or schema (Fallside, 2000) for encoding LTAG resources in order to exchange grammars, share tools and compare parsers.
- 2. Exploit XML, its features and the related recommendations for the representation of complex and redundant linguistic structures based on a general methodology.
- 3. Study the resource organisation and the level of generalisation which are relevant for a lexicalized tree grammar.

1. Introduction

A working group gathering people, mainly from TA-LaNa (University of Paris 7, France), ENST (Paris, France), INRIA (Rocquencourt, France), LORIA (Nancy, France) and DFKI (Saarbrücken, Germany) who are currently working on this formalism, made it necessary to define a shared and common representation of grammars with the aim of exchanging both grammars and associated resources, developing normalised parsers and specifying generic tools. Our proposal, TagML (Tree Adjoining Grammars Markup Language) is a general recommendation for the encoding and exchange of the resources involved in LTAG. This paper presents a model and a syntax to represent, encode and maintain LTAG grammars independently of any development, software and architecture.

A significant number of works are based on the TAG (Tree Adjoining Grammar) formalism (Joshi et al., 1975). Still for the moment, none has led to a common representation format of the grammars which would facilitate the exchange of TAG grammars and associated data, as well as for developing normalised parsers and specifying generic tools with a full compatibility. Research and work around the formalism of Lexicalized TAG (LTAG) (Abeillé, 1991) increased during last ten years both for the linguistic point of view and for the computational level. Based on solid mathematical foundations, the linguistic choices associated to the LTAG formalism remain relatively free and contribute to the variety of results and to the important number of developments and applications.

The XTAG system, developed in the early nineties, offers the first workbench dedicated to LTAG grammar design and a Earley-like parser. However, the integrated parser provides only a binary answer (accepted or rejected sentence) hardly compatible with the test of a large grammar. Partial results and diagnostics about errors are necessary to test a grammar and to identify the step involved in the failure of a parse during grammar debugging. Thus, designing a new parser is justified but integrating new components to the XTAG system is technically very difficult for someone that has not been involved in the initial development of the system. More generally, this system has not been developed technically to be distributed since it is based on proper and non specified formats. It requires a narrowly-specialised skill for its installation, its usage and its maintenance.

In this introduction, we describe our approach for the definition of a generic architecture for encoding and managing TAG grammars, the contribution of XML and a global structure for an LTAG grammar. The remainder of this paper is organised as follows. In the section 2, we give an overall view of the TagML architecture and we start with a presentation of the elementary tree encoding principles including a description of the phrase structure components and the feature structures and their place within the TagML architecture. We complete the section with the notion of tree families allowing a meta-description and organisation of elementary trees. In section 3, we tackle the problems connected to the lexicon management and their links with the remainder of the resources. We propose an organisation of these resources within an abstract relational model. Section 4 is concerned with managing of the parsing result and output, which means representing all derived trees (the phrase structures of a sentence) and in parallel all derivation trees (structures closed to semantic dependency tree of a sentence).

1.1. Towards a generic architecture

The definition of a generic tool for parsing and managing LTAG grammars supposes a common language specification, shared by the concerned community. The first step toward a more generic and flexible tool undergoes the definition of an appropriate encoding for the management of large-size linguistic resources. This encoding should be able to structure possibly heterogeneous data and give the possibility of representing the inevitable redundancies between lexical data. Consequently, we decided to define TagML as an application of the XML recommendation.

Derived from the SGML, a standard (ISO, 1986) for encoding electronic texts with information about the structural layout and content of the document, the XML recommendation stands out as one of the best encoding schema intended for structuring information and providing interesting possibilities for managing and accessing textual data components. These aspects have been exploited for managing linguistic resources within the Text Encoding Initiative or TEI guidelines (Sperberg-McQueen and Burnard, 1994). The normalisation of resource associated to a LTAG grammar is a necessity first to interchange data between members of the community working on this formalism, then to share tools with the aim to evaluate and compare our results. This normalisation process will offer to the community the opportunity to take benefit of some existing tools (editors, grammar design workbench, tools for testing and comparing different parsers, etc.) and also to exploit reusable software components. Anyone implementing a tool on the basis of the TagML encoding can guarantee its interoperability with existing ones.

The initial motivations for this encoding proposition are mainly centred on the notion of grammar re-usability as well as the software independence and perenniality on the whole. It should be noted that the choices we proposed in this paper are complementary to a set of tools intended to be easily and freely distributed to the community: we could mention an XML parser, graphical editors and a parsing workbench. The developments are based on Java which ensures them reliability and portability.

1.2. Why XML for encoding LTAG grammars?



Figure 1: Basic example of virtual resources and reusability

Although it is still young, the motivation for using XML for encoding LTAG resources comes from the following properties that appear to be particularly relevant for our needs:

- XML is a meta-language for defining markup languages. It provides a common syntax for structuring resources according to their content, meaning, and above all their logical structure. It provides a means to encode and exchange linguistic resources in an independent way, between applications for display, manipulation and processing.
- The virtual resources principle (several view and/or level of annotation onto the same data) can be exploited for the management of lexicon by offering different entry points to the same data (see the example on figure 1). For example, one could reverse a morpho-syntactic lexicon designed first for parsing (entries are the inflected forms) to a morpho-syntactic lexicon dedicated for the generation task (entries are the lemma and a set of morphological features). This notion of virtual resources avoid notably the duplication of data at the physical level and make the maintenance of the resources easier.
- The consistency of an LTAG grammar is very important for developing a broad-covering grammar which supposes several developers, several lexical components, etc. In our case, the consistency of a grammar is a consequence of the validation of its XML encoding with a specific DTD defined by the concerned research community.
- Loading a whole Lexicalized TAG with a system such as XTAG (Doran et al., 1994) is time-costly and resource consuming. In term of implementation, it means that some important efforts have been made to normalise and especially optimise the input reading and access to XML data. An interesting property of XML is that it is no more necessary to load the whole XML encoded lexicon to search for some particular entries. Some normalised software components, as the SAX (Simple API for XML) interface, provides this kind of functionalities in a plain manner.
- The requirements in data typing and preprocessing (for example typing in term of left or right auxiliary tree) can be easily solved at two different levels, either at the description level or at the application level. The first level means that we can describe the propriety to test with a restricted DTD. The second level is handled by the XML application and the propriety is tested by the implementation. Both solutions are of course combinable.
- Finally the semi-structured data model, underlying to XML, allows the use of extended queries based both on the hierarchical structure and the contents of the resources.

1.3. Structure of an LTAG grammar

The exploitation of virtual resources for the encoding of a LTAG grammar is promising but supposes to identify explicitly within a whole LTAG grammar the various resources involved first in the morphological component, in the syntactical lexicon and in the set of elementary trees, but also in the shared forests corresponding to derived and derivation trees resulting from a LTAG parsing.

Writing a *global DTD* for all these resources supposes to identify the constraints on these different data. The writing of a *property DTD* allow complementary to exploit the descriptive power of XML to check specific properties and consistency constraints in a LTAG grammar.

The application of an XML encoding can be view as a linguistic engineering work, but the researches needed to define the encoding principles suppose a deep study of the LTAG formalism and its properties. We will see that this study has also opened interesting issues for the parsing point of view. More generally, we think that this work shows the relevance of the XML formalism for the representation of complex heterogeneous data.

2. Encoding of an elementary tree schema

2.1. Principle

We call *elementary tree schema* a non-lexicalized elementary tree which is the classical tree used in existing LTAG lexicon to factorize complete elementary tree representations. The term *schema* can be also used, see (Candito, 1999). In an elementary tree schema, we can distinguish:

- The structural part, i.e. a partial phrase structure or a partial parsing tree.
- The set of feature equations constraining top and bottom feature structures.

One can note that these two parts present many redundancies in the different elementary trees due to the lexicalization and the extended domain of locality property. We want to be able to encode these redundancies in order to exploit them to improve the parsing process.

Some specifications for the encoding of elementary tree families has been proposed on the basis of the SGML norm in (Issac, 1998). A tree family gathers the elementary tree schemas that can be considered as the syntactic realization of the same predicate-argument schema. This kind of structure for the set of elementary trees is frequent because it makes the development of a grammar easier. Still, by associating a tree family to a lemma, the entry can really anchor only a subset of the elementary tree schemas of this family. This subset can be small for inflected languages as French, Spanish or Korean. The selection is proceeded with filtering features during the lexicalization stage. Such an unification operation is costly while it is possible to indicate statically in the lexicon the exact set of elementary tree schemas that can anchor a precise inflected entry. Our choice is to consider the elementary tree schema description as the document to encode. A tree family is just a particular and optional view on a set of these elementary tree schemas.

An example of the representation of a schema proposed by (Issac, 1998) is given on figure 2. We can note that the encoding of the features is basic and just correspond to introduce common labels for shared feature values. We exploit XML first to encode feature equations without these labels, secondly to avoid redundancies.

We keep from (Issac, 1998) most of the elements involved in the elementary tree schema structure encoding:

```
/* ... */
<n id="n1">
    <val>&amp; P</val>
    <fs type="b">
        <fs type="b">
        <fs type="b">
        <fs type="b">
        <fs type="b">
        </fs type="b">
        </fs type="b">
        </fs
        </fs
        </fs
        </fs>
        // * ... */
</n>
```

Figure 2: Node representation in (Issac, 1998)

- < t > : elementary tree, document that we specify in this part.
- < n > : general node, the attribute *cat* gives the category of this node and the attribute *type* distinguishes foot node, substitution node and anchor.
- < fs > : feature structure, of type bottom or top
- < f > : typed feature (attribute-value) similarly to the TEI. For typed feature equation, we introduce the element *linkGrp* specified in the TEI specifications to group internal or external links (element *link*) and their re-usability.

2.2. Structural component

Similarly to (Issac, 1998) proposal, we represent straightforwardly by an isomorphy the tree structure of an elementary tree schema and the XML tree structure (see figure 3).



Figure 3: Isomorphy between elementary tree schema and XML tree structure

In practice in a broad-covering lexicalized grammar, the redundancy of common substructures is very important. For instance, the subtree dominated by a V category with a depth of 1 (the anchor and the pre-terminal category) is shared by most of the trees describing verbal syntactical context (several hundred of trees for the English XTAG grammar, several thousand for the French LTAG grammar). This redundancy can be very useful to encode for linguistic or efficiency issues. In order to represent these redundancies, we propose to use the XML links and to identify systematically every nodes. We use the principle of virtual resources systematically to obtain only one representation of the different nodes within the whole grammar. Consequently each structure or complete elementary tree is a particular structuring view of these XML documents.

2.3. Feature equations

The TEI proposes a recommendation for the encoding of feature structures that we propose to integrate to TagML.

This normalisation allow to type the features and to represent explicitly feature percolation. The features used in the LTAG formalism are only with atomic value thanks to the extended domain of locality principle.

The feature equations of an elementary tree schema can be view as a global term for a complete elementary tree, or as several terms distributed in the various nodes of an elementary tree sharing common variables. We propose to link directly the shared features in order to avoid the necessity to manage shared labels during the parsing of the features structures. These links are specified in *linkGrp*.

We have the possibility to give a type to a *linkGrp*, i.e. for a feature equation, for instance *subject-verb agreement*, then by identifying this *linkGrp* to share the corresponding feature equation to several elementary tree schemas. If we still consider the example of *subject-verb agreement* feature equation, the corresponding *linkGrp* will be shared by all elementary tree schemas that include this kind of agreement. The nodes that carry the features linked by percolation can be identified given the two following ways:

- By the definition of global and unique identifiers for the nodes for all the elementary tree schema belonging to the a unique tree family (all the nodes that represent a subject are identified by the same *id*).
- By a special attribute which identify the function of a given node involved in the feature equation. The access to these specific nodes are obtained with the selection language proposed both for XSL Tranformation Language (Clark, 1999) and for the XML pointers called XML Paths (Clark and DeRose, 1999).

As we can see in figure 4, the percolated feature is linked to the *linkGrp* corresponding to the feature equation, so it is straightforward to access with this link all the other features which shared the same value, without dealing with any labels and table of labels.

```
<n cat="P" id= "n0">
   'f()' >
         <link
                 xlink:type=
                                  "simple"
                                  "doc#id(IO')/>
           xlink:href=
      </f>
      <f name="det" id= "f1"><minus/></f>
   </fs>
   <fs type= "bottom id= "fs1">
   /* ... */
</fs>
</n>
 /* External document */
<linkGrp type="accora >
<link targets="
id(n0)/fs[1][@type,top]/f[1][@name,num]
id(n2)/fs[1][@type,bottom]/f[1][@name,num]"
id= "I0"/>
```

Figure 4: Shared features and factorisation of common feature equation

These identifications of nodes are fully compatible with the automatic generation system of elementary tree schemas of (Candito, 1996) and both works are complementary. Such a system can identify the unique function associated to the different nodes of a given elementary tree schema. Since the feature equations are shared and typed, we can apply on them a specific treatment in order to shared computation and consequently decrease significantly the number of unification. This optimisation is important because the worst case complexity of the unification in LTAG is exponential.

2.4. Morpho-syntactic lexicon

2.5. Global structure of a TagML document

The TagML DTD is quite simple since it defines only 15 elements and 16 attributes (without counting the XML Link attributes). A < tag > document is composed of < tlist > elements, each one containing a list of generic trees < t > (see figure 6).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<desc>Generic trees for determiners</desc>
<t id= "A1_determiner1 n="1" name="determinet"></desc>
       <desc>Tree description goes here</desc>
       <sample>A sample</sample>
       < n > / \frac{1}{2}
                   */</n>
              . . .
     </t>
     <t>
       <n>/* ... */</n>
     </t>
  </tlist>
  <tlist>
            * /
     /*
         . . .
  </tlist>
/* ... */
</tag>
```

Figure 6: Global structure of a TagML document containing a set of generic elementary trees

2.6. Tree family

In order to manage efficiently a set of elementary trees that could be quite large, TagML provides a mechanism allowing to gather elementary trees sharing a same subcategorisation frame and corresponding to different syntactic structures. A possibility to describe a tree family (indicated by the tag < tfamily >) from a set of elementary tree schemas is obtained by defining a set of links to a subset of elementary tree schemas.

The figure 7 presents an example of tree family definition (in this example *I1_VTA_0* and *I2_VTD_1B* refers to two elementary tree schemas for transitive verbs and *I2_adjectif6* and *I1_adjectif1* to two elementary tree schemas for adjective).

3. Lexicon

For a Lexicalized Tree Grammar, lexicon and grammar are merged into a syntactic lexicon, but we usually consider three kinds of data bases:

- a morpho-syntactic lexicon
- a syntactic lexicon



Figure 5: RROM for morphological lexicon.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<! DOCTYPE tag SYSTEM "tagml.dtd>
<! DOCTYPE tag SYSTEM
<tag xmlns:xlink= "http:
<descs.c
                          "http://www.w3.org/XML/XLink"/0.9
   <desc>Our tree families</desc>
<tfamily name="transitive verb>
      cdesc> Tree family for transitive verbs</desc>
<t xlink:type= "simple"</pre>
                            "I1_VTA_0.xml"
      xlink:href
      xlink:show=
                            "replace
                                 'auto"/>
      xlink:actuate=
                            " simple"
" I2_VTD_1B.xml "
      <t xlink:type=
      xlink:href
      xlink.show=
                            "replace"
       xlink:actuate=
                                 'auto"/>
   </tfamily>
   <tfamily
                name="adjective">
      <desc>Tree family for adjectives</desc>
      <t xlink:type=
                            "simple"
      xlink:href
                            'A1_adjectif1.xmľ
      xlink:show=
                            "replace
                                 'auto"/>
      xlink:actuate=
      <t xlink:type=
                            'simple"
                            "I2_adjectif6.xm"l
      xlink href
      xlink:show=
                            "replace'
       xlink:actuate=
                                 'auto"/>
   </tfamily>
/* ... */
</tag>
```

Figure 7: Sample of a TagML document and two tree families

• a set of elementary tree schemas

The encoding of the syntactic grammar is more complex that single elementary tree schemas. The role of this lexicon is to link lexical entries to the right set of schemas. Figure 8 proposes an example of a very simple encoding for this lexicon, which only consist in an enumeration of the correct schema for all valid inflected entries. The complexity is the consequence of the fact that many pieces of information are in relation and are distributed in these three kind of data.

Our first attempt to define encoding principles for these lexicon was done directly on the basis of the XML formalism without any special regards on the abstract organisation of the data. This first result was not satisfactory for two main weaknesses: the limited possibility for extending the encoding principles and the limited sharing of distributed

```
<lexicon
           type= "syntax" >
  <entry flex= "voile">
     <lemma form="voiler">
           att= "cat">v</f>
       <f
          att= "num">sing</f>
       <f
           att= "mode">ind</f>
       <f
           xlink:type=
                           simple'
        <†
                          "replace"
           xlink:show=
           xlink:href=
                          "I1_VTA_4#id(n0)"/>
           xlink:type=
                          "simple
        <†
                          "replace"
           xlink:show=
                          "I1_VTC_B#id(nO)"/>
           xlink:href=
           . . .
     </lemma>
     /*
              * /
         . . .
  </entry>
  /*
           *
      . . .
</lexicon>
```

Figure 8: A basic encoding of a syntactic lexicon with links to elementary tree schemas

resources according to the virtual resource principle. The main reason is that XML does not offer an abstract view on the logical organisation of resources that would allow to define directly general encoding principles. To model these resources and their global organisation we have then used an abstract relational model which allows the representation of each independent resource and its relation to others. This abstract relational model have a direct realisation in XML.

The relation model for LTAG is presented in the next section and should result in an XML DTD for the syntactic grammar level in future work.

The RROM 3.1.

Our abstract level of representation is called RROM (Relational Resource Organisation Model). A RROM is composed of a set of Resource Entities (RE) and a set of relations between these entities. A RE corresponds to an independent and abstract type of data that is used in a NLP system (for example word, lemma or category). Given a set of resources, Independent data means that this data is not the result of a set of relations between other RE. A RE is represented with a general name and is associated to a data



Figure 9: Simplified RROM for LTAG resources.

type definition. An *instantiation* of a RE is a realization of this RE according to the corresponding data type specifications. In following figures, an RE is graphically represented with a square box.

The relations between entities used in this model are characterised by two couples of integers on each edge. Depending on the direction of the relation, this couple gives the arity of the relation with the RE given by the edge, by analogy to the couples on the edges used in relational databases entity/relation models. Two RE can also be in relation. A RROM can be graphically represented with diagrams describing which REs are related to one another. In these diagrams, a Resource Relation (RR) is represented with ellipsis. We distinguish two kinds of edges: *Unary* edges (single line) which indicate a single link relation and n - edge (double line) which means that a relation can link n instantiations of a RE at the same time.

A morphological lexicon database, as MULTEXT (Ide and Véronis, 1994), usually associates an inflected word to a set of lemmas and a set of features. Reversible access is needed for generation for example. A lemma is an abstract entity that is represented with a normal form of a word (the entry of a dictionary) and can be realized with all possible flexions of a word. We can distinguish as resources entities inflected words, lemma and morphological features (including a category) that will characterise the inflection. An inflection is a relation between one inflected word, one lemma and a set of morphological features. Depending on the sense that one follows this inflection relation (from the lemma or from the inflected word), we obtain a reversible access. Each lemma is characterised by a link to one inflected word which is the normal form that identify this lemma (see figure 5). Respectively, an inflected word is not always the normal form of a lemma.

3.2. The LTAG syntactic lexicon

The previous RROM model for morphological lexicon is extended to the other resources needed at the syntactic level. An inflection (a lemma and a set of morphological features including verb mode for example) corresponds to a set of schemas. This lexicalization relation can include the instantiation of co-anchors (a lemma and a set of possibly under-specified morphological features) and of some additional syntactic features in the schema. Each syntactical instantiation give a complete elementary tree. If we assume that linguistic principles given in (Abeillé et al., 1990) and (Candito, 1999) are fulfilled by the grammar, each syntactical instantiation corresponds to only one semantic instantiation (semantic consistency principle). This model allows an incremental view of the lexicon resources that could be extended easily.

The figure 9 presents the corresponding RROM. To simplify, tree families and structuration of features are not included in this example.

Such an approach based on a relational model to define XML encoding has also been used for the encoding of multilevel annotated textual corpus (Lopez and Romary, May 2000).

4. Parsing forest

4.1. Principle

The result of a parsing based on a LTAG is two packed representations called shared forests representing respectively all derived trees and all derivation trees. The representation of such a forest with XML is possible by using XML links. The resulting structure is equivalent to an acyclic graph representation. Maintaining this kind a shared structure allows a more compact representation according to the size of the data but also more efficient and useful for sharing additional semantic processing.

4.2. Derived tree forest

A derived shared forest is an element corresponding to a tag < ddf >. The trees are then expressed similarly as an elementary tree schema. The nodes can contain only one feature structure *center* resulting from the unification of top and bottom feature structures, or the two non-unified feature structures if we consider a partial derived tree.

4.3. Derivation tree forest

We consider here two kinds of nodes (corresponding to a element with the tag $\langle n \rangle$) for the derivation trees (corresponding to a tag $\langle df \rangle$): node for initial tree (the value of the attribute *type* is *i*) and node for auxiliary tree (the value of the attribute *type* is *a*). For such a node representing a given elementary tree, an additional attribute also represents the Gorn address of the node where the attachment has been realized with the father tree, the name of the elementary tree schema and the lexical string anchoring the tree.

5. Conclusion

We have presented in this paper the first specifications of a general encoding of the various linguistic resources involved in the LTAG formalism called TagML. This work can be view as a generalisation and a normalisation of the XTAG format. It includes first a complete specification for the encoding of elementary tree schemas:

- Used in an implemented graphical workbench for LTAG.
- Associated to a XSL style sheet in order to produce LATEX documentation (on the basis of the *pstricks* package).

We have also proposed some high level specifications for the lexicon based on a relation model called RROM and a straightforward extension to the encoding of results (derivation and derived forests). The lexicalization of the formalism and the complex distribution of the resources in several knowledge sources raise several problems if we want to capture sharing properties. Considering these difficulties, the XML encoding formalism is powerful and relevant to represent complex heterogeneous linguistic resources. Future works on TagML will complete the encoding specification of the lexical components.

Parallel works (Lopez and Romary, May 2000) focus on the efficiency of XML-based processing, including an efficient internal representations directly deduced from XML documents and based on Finite State Techniques. Applied to TagML, our ambition is then to provide a complete and efficient LTAG resource management system based on an XML architecture. We welcome all contributions to the current undergoing development of the TagML specification and we hope that it will appear enough promising to give rise to interests and possible contributions from the whole LTAG community.

6. References

- Anne Abeillé, Kathleen M. Bishop, Sharon Cote, and Yves Schabes. 1990. A Lexicalized Tree Adjoining Grammar for English. Technical Report MS-CIS-90-24, Departement of Computer and Information Science, University of Pennsylvania.
- Anne Abeillé. 1991. Une grammaire lexicalisée d'arbres adjoints pour le français. Ph.D. thesis, Université Paris 7.
- Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen, 1998. Extensible Markup Language (XML) 1.0. W3C, http://www.w3.org/TR/REC-xml, February. W3C Recommendation 10-February-1998.
- Marie-Hélène Candito. 1996. A principle-based hierarchical representation of LTAGs. In *COLING'96*, Copenhagen, Denmark.
- Marie-Hélène Candito. 1999. Structuration d'une grammaire LTAG : application au français et à l'italien. Ph.D. thesis, University of Paris 7.
- James Clark and Steve DeRose, 1999. XML Path Language (XPath) Version 1.0. W3C, http://www.w3.org/-TR/xpath, November. W3C Recommendation 16 November 1999.
- James Clark, 1999. XSL Transformations (XSLT) Version 1.0. W3C, http://www.w3.org/TR/xslt, November. W3C Recommendation 16 November 1999.
- Christy Doran, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. 1994. XTAG System - A Wide Coverage Grammar for English. In *COLING*, Kyoto, Japan.
- David C. Fallside, 2000. XML Schema Part 0: Primer. W3C, http://www.w3.org/TR/xmlschema-0, April. W3C Working Draft, 7 April 2000.
- Nancy Ide and Jean Véronis. 1994. Multext (multilingual tools and corpora). In 14th Conference on Computational Linguistics (COLING'94), Kyoto, Japan.
- ISO, 1986. Information Processing, Text and Office Systems, Standard Generalized Markup Language (SGML) = Traitement de l'information, systemes bureautiques, langage standard gnralis de balisage (SGML). First edition, 1986-10-15. International Organization for Standardization, Geneva, Switzerland. International Standard ISO 8879-1986. Federal information processing standard; FIPS PUB 152.
- Fabrice Issac. 1998. A Standard Representation Framework for TAG. In Fourth International Workshop on

Tree Adjoining Grammars and Related Frameworks (TAG+4).

- A. Joshi, L. Levi, and M. Takahashi. 1975. Tree adjunct grammars. *Journal of the Computer and System Sciences*.
- Patrice Lopez and Laurent Romary. May, 2000. A Framework for Multilevel Linguistic Annotations. In *LREC Workshop on Data Architectures and Software Support for Large Corpora: Towards an American National Corpus*, Athens.
- C.M. Sperberg-McQueen and Lou Burnard. 1994. *TEI Guidelines for Electronic Text Encoding and Interchange (P3).* Chicago and Oxford, http://etext.virginia.edu/TEI.html. ACH/ACL/ALLC Text Encoding Initiative.