Towards an international standard on feature structure representation

Kiyong Lee[1], **Lou Burnard**[2], **Laurent Romary**[3], **Eric de la Clergerie**[4]
Thierry Declerck[5], **Syd Bauman**[6], **Harry Bunt**[7], **Lionel Clément**[4]
Tomaz Erjavec[8], **Azim Roussanaly**[3], **Claude Roux**[9]

[1]Korea University Linguistics, Seoul, Korea
klee@korea.ac.kr
[2]Oxford University Computing Services, UK
lou.burnard@oucs.ox.ac.uk
[3]LORIA, France
{Laurent.Romary—Azim.Roussanaly}@loria.fr
[4]INRIA, France
{lionel.clement—Eric.De-La-Clergerie}@inria.fr
[5]Saarland University & DFKI, Germany
declerck@dfki.de
[6]Brown University, USA
syd_bauman@brown.edu
[7]Tilburg Univeristy, The Netherlands
Harry.Bunt@uvt.nl
[8]Jozef Stefan Institute, Slovenia
tomaz.erjavec@ijs.si
[9]XEROX Research Center Europe, France
claude.roux@xrce.xerox.com

Abstract

This paper describes the preliminary results of a joint initiative of the TEI (Text Encoding Initiative) Consortium and the ISO Committee TC 37SC 4 (Language Resource management) to provide a standard for the representation and interchange of feature structures.

1. Introduction

This paper describes some preliminary results from a joint initiative of the TEI (Text Encoding Initiative) Consortium and the ISO Committee TC 37/SC 4 (Language Resource management), the goal of which is to define a standard for the representation and interchange of feature structures. The joint working group was established in December 2002, and its proposals are now progressing to Draft International Standard status.

1.1. TEI

Initially launched in 1987, the Text Encoding Initiative (TEI) is an international and interdisciplinary effort the goal of which is to help libraries, publishers, and individual scholars represent all kinds of literary and linguistic texts for online research and teaching, using an encoding scheme that is maximally expressive and minimally obsolescent. The TEI has also played a major role in the development of European language engineering standards since the days of EAGLES. Its recommendations, the "TEI Guidelines", underpin such key standards as the Corpus Encoding Standard, and address many other areas of language resource documentation and description, as well as lexicographic and terminological databases. Since 2000, maintenance and development of the TEI has been managed by an international membership Consortium, which announced publication of a complete XML version of the TEI Guidelines, known as P4 in 2002, and is now overseeing production of a major new revision, known as P5.[1]

1.2. TC 37/SC 4

The research areas of ISO/TC 37/SC 4 include computational linguistics, computerized lexicography, and language engineering. Language resources consist of contents represented by linguistic data in various formats (e.g., speech data, written text corpora, general language lexical corpora). Text corpora, lexica, ontologies and terminologies are typical instances of language resources to be used for language and knowledge engineering. In both monolingual and multilingual environments, language resources play a crucial role in preparing, processing and managing the information and knowledge needed by computers as well as humans. With a view to mobile computing and mobile content etc., the availability of language resources, having to be considered as multilingual, multimedia and multimodal from the outset will be one of the key success factors.[2]

1.3. Current topics of the joint group

The joint TEI and ISO activity has focussed on the following topics:

- articulation of a detailed technical proposal for an XML format able to represent a feature structure anal-

[1]See also http://www.tei-c.org/
[2]See also http://www.tc37/sc4.org/

ysis with a precise description of the underlying formal mechanism to ensure the coherence and soundness of the standard in line with major theoretical works in this domain;

- provision of specific mechanisms to deal with reentrant structures, clearly distinguished from a generic pointing mechanism;
- provision of a coherent description of the notion of type, which will enable further development of the standard to include a complementary set of proposal relating to declaration of a Feature System.
- integration of this proposal into the on-going revision of the TEI Guidelines (TEI P5) due for publication in 2004;

2. Goal of the paper

The paper first introduces the basic concepts of the features structure formalism. Section 4 briefly describes the proposal currently being developed as an ISO Standard and its relation to other ongoing work relating to the deployment within ISO TC 37 of a general data category registry for linguistic description. The current proposals will include this and other external sources for use, as a reference, in the declaration of particular feature sets. Finally some conclusions are drawn.

3. Feature structures

Feature structures (FSs) form an essential part of many language processing systems, whether their focus is on the description, enrichment, storage, or management of linguistic data. The FS formalism itself has a formal background in graph theory, and supports powerful unification mechanisms for combining elementary structures, which have facilitated its use in many real-world applications. There are many possible ways of representing FSs, but the basic notions have an intrinsic legibility which make them very useful for representing linguistic information in interchange situations, both between people and between processing systems. To take full advantage of this capability, a standard way of representing such structures in electronic format should be made available so that a) specialists from diverse application fields can share detailed expertise from diverse domains and b) implementers can share basic libraries dedicated to the manipulation of FSs, thus reducing the overall cost of application development.

FSs are uniform objects that can be used to represent a wide range of objects, ranging from very simple structures consisting of simple lists of feature-value pairs, to highly complex typed and nested structures with reentrancy, as found for instance in HPSG (Pollard and Sag, 1994), LFG (Bresnan, 1982), etc. More recently, FSs have also been used as the internal representation for shallow and robust NLP systems based on finite state technologies, or for merging information sources coming from distinct modalities in multi-modal systems.

4. The proposal

This proposal combines a basic set of tags for representing features and feature structures covering in a uniform way the full range of complexity attested by current implementations, together with additional mechanisms to describe libraries of values, feature value pairs and feature structures. As an example, consider the following simple morpho-syntactic annotation for the word ‘vertes’ in French:

```
<fs>
   <f name='token'>
     <string>vertes</string>
   </f>
   <f name='lemma'>
     <string>vert</string>
   </f>
   <f name='pos'>
     <symbol value='adj'/>
   </f>
   <f name='gender'>
     <symbol value='fem'/>
   </f>
   <f name='number'>
     <symbol value='plural'/>
   </f>
</fs>
```

In this XML representation, the element `<fs>` is used to encode a feature structure, and the `<f>` element is used for each of five feature-value pairs making up this structure. Each feature-value pair has a name, given by the name attribute, and contains a primitive or atomic value, marked (in this case) by either a `<string>` or a `<symbol>` element, depending on its datatype. Other possible child elements for the <f> element include <binary> for binary- or boolean-values such as PLUS or MINUS, and <numeric> for various kinds of numeric values and ranges. Complex values can also be represented: collections or multivalues such as lists, sets or multisets (bags) are tagged using a <coll> element; feature structures may also be used as feature-values, thus providing a recursive ability. The components of particular feature structures may be represented directly or referred to by using pointers to previously stored “libraries” of features or feature values. We believe that this XML representation has equivalent expressive power to the classical AVM (Attribute-Value-Matrix) notation, but is more readily processed.

In developing the XML representation, the work group was able to simplify considerably the original TEI proposals as described in (Langendoen and Simons, 1995), by focussing on applications of the formalism in linguistic analysis alone. The availability of new XML-based tools, in particular the relax-NG schema language now used to express the TEI markup scheme, also proved beneficial for developing a powerful and expressive formalism, adequate to the needs of those using feature structure analysis.

Applications for this formalism have demonstrated the need for more complex mechanisms, which are needed to handle elaborated linguistic information structures. Following on from reference works by Shieber (PATR-II)

(Shieber, 1986) or (Carpenter, 1992), there has been a whole range of implementations of FSs in computational linguistics applications. Examples include LOGIN/LIFE (Ait-Kaci and Nasr, 1986), ALE (Carpenter and Penn, 1996), Profit (Erbach, 1995), DyALog (de la Clergerie, 2002), ALEP (Simpkins and Groenendijk, 1994), WAM-like Abstract Machine for TFS (Wintner and Francez, 1995), etc. From another point of view, one can consider the variety of linguistic levels concerned with such representations, e.g. phonology, morpho-syntax, grammars (unification grammars: LFG, HPSG, XTAG), linguistic knowledge base or practical grammar implementation guide (LKB, (Copestake, 2002)), underspecified semantics (RMRS, (Copestake et al., 1999)), or integration of NLP components (Schaefer, 2003).

In our work, we have identified and discussed a certain numbers of concepts and topics introduced in the works cited above and we are proposing an XML-based way of representing the corresponding feature structures. As examples, given for this short paper, we show the actual XML implementation of *structure-sharing* (also called *reentrency*) and the XML treatment of *types*, two topics mentioned in 1.3.:

4.1. Structure Sharing

As shown in most of the works cited above, *structure sharing* (or *reentrancy*) requires the use of labelling for representation in graphic notation such as AVM. For example, to show that a given feature-value pair (or feature structure) occurs at multiple points in an analysis, it is customary to label the first such occurrence, and then to represent subsequent ones by means of the label.

In discussing how to represent this in an XML-based notation, we first proposed making use of a global attribute `label` or `n`, as in the following simple example:

```
<fs>
 <f name="specifier">
  <fs>
    <f name="agr" n="@1">
    <fs>
       <f name="number">
          <symbol value="singular"/>
       </f>
    </fs>
    </f>
    <f name="pos">
         <sym value="determiner"/>
    </f>
  </fs>
 </f>
 <f name="head">
   <fs>
    <f name="agr" n="@1"/>
    <f name="pos">
       <sym value="noun"/>
    </f>
   </fs>
 </f>
</fs>
```

The feature named “agr” is here labelled “@1”. Its first occurrence contains a feature-value pair (“singular number”); its second references this same feature-value pair.

An alternative way of representing this phenomenon is to use the XML ID/IDREf mechanism, as follows:

```
<fs>
 <f name="specifier">
  <fs>
    <f name="agr" id="N1">
    <fs>
       <f name="number">
          <symbol value="singular"/>
       </f>
    </fs>
    </f>
    <f name="pos">
         <sym value="determiner"/>
    </f>
  </fs>
 </f>
 <f name="head">
   <fs>
    <f name="agr" fVal="N1"/>
    <f name="pos">
       <sym value="noun"/>
    </f>
   </fs>
 </f>
</fs>
```

The working group has identified a need to distinguish the case where co-reference implies copying (or transclusion) of shared structures or values, from the case where co-reference simply implies multiple references to the same object, but has not yet reached a resolution as to which of the possible approaches best meets this need.

4.2. Typed Feature Structure

The *typed feature structure* has become a key tool in the linguistic description and implementation of many recent grammar formalisms,

4.2.1. Types

Elements of any domain can be sorted into classes called *types* in a structured way, based on commonalities of their properties. Such linguistic concepts as *phrase*, *word*, *pos* (parts of speech), *noun*, and *verb* may be represented as features in non-typed feature structures. But in typed feature structure particular feature-value pairs may be treated as types.

By *typing*, each feature structure is assigned a particular type. A feature specification with a particular value is then constrained by this typing. A feature structure of the type *noun*, for instance, would not allow a feature like TENSE in it or a specification of its feature CASE with a value of the type *feminine*.[3]

4.2.2. Definition

The extension of non-typed feature structure to typed feature structure is very simple in a set-theoretic framework. The main difference between them is the assignment of types to feature structures. A formal definition of typed feature structure can thus be given as follows:[4]:

[3]Note that atomic feature values are considered *types*, too.

[4]Slightly modified from (Carpenter, 1992).

Given a finite set of **Features** and a finite set of **Types**, a typed feature structure is a tuple $\mathcal{TFS} = < \mathbf{Nodes}, r, \theta, \delta >$ such that

i. **Nodes** is a finite set of nodes.

ii. r is a unique member of **Nodes** called *the root*.

iii. θ is a total function that maps **Nodes** to **Types**.

iv. δ is a partial function from **Features**×**Nodes** into **Nodes**.

First, each of the **Nodes** must be rooted at or connected back to the root r. Secondly, there must one and only one root for each feature structure. Thirdly, each of the **Nodes**, including the root r node and terminal nodes, must be assigned a type by the typing function θ. Finally, each of the **Features** labelling each of **Nodes** is assigned a unique value by the feature value function δ.[5]

This type type of information can be encoded in an XML notation, as an example (simplified, due to the length of the paper) shows below:

```
<fs type="word">
  <f name="orth">
    <string>love</string>
    </f>
  <f name="syntax">
    <fs type="verb">
      <f name="valence">
        <symbol value="transitive"/>
      </f>
    </fs>
  </f>
</fs>
```

Note here that the line <f name="pos"><sym value="verb"/></f> in the embedded feature structure <fs> has been replaced by typing that <fs> as in <fs type="verb">.

5. Related work within the ISO framework

A distinctive feature of the TEI Guidelines is its use of an integrated model of documentation and documentation outputs. The ODD system used to produce its recommendations, both as printed documentation and as formal syntax expressed in XML Schema or DTD languages, has recently been revised and re-expressed. This new modular system for documentation is likely to have wide take up in many different domains. In applying it to the expression of the feature structure analysis language, we have identified a number of potential areas of synergy with the ongoing ISO work on data category registry[6].

[5]The unique-value restriction on features does not exclude multi-values or alternative values because even in these cases each feature ultimately takes a single value which may be considered complex in structure.

[6]See for more details: http://jtc1sc36.org/doc/36N0581.pdf

6. Conclusions

The work reported has proved to be an excellent opportunity for experimenting with the new descriptive framework being developed for the TEI Guidelines themselves. The feature structure activity has been a useful opportunity to experiment with the creation of relevant tagging systems and tools in a relatively limited but formally complex domain.

In general, the activity reported in the paper shows that there is great scope for further convergence between the TEI consortium and ISO committee TC 37/SC 4, and many benefits to be gained from joint work on issues which require complementary expertise in textual representation methods and in the representation of linguistic concepts.

7. References

Ait-Kaci, H and R Nasr, 1986. Login: A logic programming language with built-in inheritance. *J. Log. Program.*, 3(3):185–215.

Bresnan, Joan (ed.), 1982. *The Mental Representation of Grammatical Relations*. Cambridge, MA: The MIT Press.

Carpenter, Bob, 1992. *The Logic of Typed Feature Structures*. Cambridge University Press.

Carpenter, Bob and Gerald Penn, 1996. Efficient parsing of compiled typed attribute value logic grammars. In H. Bunt and M. Tomita (eds.), *Recent Advances in Parsing Technology*. Kluwer, page Recent Advances in Parsing Technology.

Copestake, Ann, 2002. *Implementing typed feature structure grammars*. CSLI.

Copestake, Ann, Dan Flickinger, Ivan Sag, and Carl Pollard, 1999. Minimal recursion semantics: An introduction. Draft.

de la Clergerie, Éric Villemonte, 2002. Construire des analyseurs avec dyalog. In *Proceedings of TALN '02*.

Erbach, Gregor, 1995. Profit: Prolog with features, inheritance, and templates. In *Proceedings of EACL '95*.

Langendoen, Terence D. and Gary F. Simons, 1995. A rationale for the tei recommendations for feature-structure markup. *Computers and the Humanities*, 29:191–209.

Pollard, Carl J. and Ivan A. Sag, 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Schaefer, Ulrich, 2003. What: An xslt-based infrastructure for the integration of natural language processing components. In *Proceedings of HLT-NAACL 2003 Workshop: Software Engineering and Architecture of Language Technology Systems*.

Shieber, Stuart M., 1986. *An Introduction to Unification-Based Approaches to Grammar*, volume 4 of *CSLI Lecture Notes Series*. Stanford, CA: Center for the Study of Language and Information.

Simpkins, N. and M. Groenendijk, 1994. The alep project. technical report. Technical report, Cray Systems / CEC.

Wintner, Shuly and Nissim Francez, 1995. Parsing with typed feature structures. In *Proceedings of the Fourth International Workshop on Parsing Technologies*.