Page Up | Cambridge Academic Links: Content Page | About ?

The TEI Dictionary Schema for XML: its worth and weakness for the lexicographer

Introduction

Since 1998, a Greek Lexicon project has been underway at Cambridge, and is due for completion in 2016. From the start, we decided to make extensive use of digital resources, both for the linguistic research which we undertook, and for the writing environment itself. Our major digital research tool is a pre-searched text database, which displays all the inflected forms for each Greek headword (this is described in Fraser 2008a).

> As a publishing medium and an authoring platform we chose an XML environment. After researching existing XML structures, including the Text Encoding Initiative (TEI) DTD, and the SGML used for the Oxford English Dictionary and the Oxford Latin Dictionary, we chose to develop our own XML structure (described in Fraser 2008b). Over the following years, I've several times been asked by colleagues why we didn't choose the TEI system. In case the reasons are of interest to other lexicographers, I've attempted to summarise them here. And first, I'll describe why we selected an XML system, not just as a publishing medium, but also as the authorial environment.

Our project and its XML requirements

We needed a system for both print and online publication, which would also function as an **authorial tool**. As we had limited funds, all our resources had to be focussed on the writing team, and so we needed an authorial tool which would **avoid double-handling** (writing plus encoding), and which accurately **reflected our entry structure** (so would already be familiar to our writers), and whose elements had **familiar names**, which would take up minimal physical space in the editing window. Our solution was a "hands-on" approach, which I call **tag and type**, as described below.

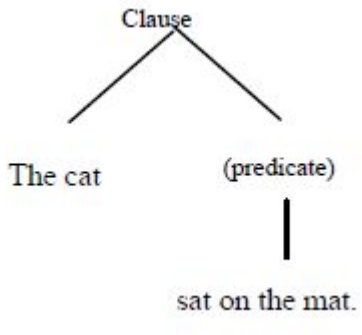
> The importance of an efficient and user-friendly authorial environment is heightened by a two additional factors. Firstly, our dictionary has a **historical and literary** approach. This requires a slightly different structure from that of a synchronic speaker's dictionary: not only describing a semantic network, but also mapping how meanings change over time and across genres. Secondly, we give **contextual information**, not only describing the meanings of the headwords, but also noting the grammatical contexts: what kinds of noun were used as subjects to each verb, or as qualified nouns to each adjective. Our entries therefore contain a wide range of text styles, from **highly-granular** morphological information to semantic and contextual information given in **sequences of connected prose**.[1]

Reasons for rejecting the TEI schema

Our reasons for rejecting the TEI approach can be explained in general or in specific terms. The **general** reason is that an off-the-peg system needs to be adapted for each project, while a tailor-made one can be designed to fit the lexicographic requirements. The **specific** reason may be stated in mathematical terms: that a beautiful equation is preferable to an ugly one. Rather than an unnecessary addition, elegance appears to be a sign of efficiency, and messiness is a signal of a weak structure and redundancy. I did in fact initially decide to use the TEI system, and my mind was changed only as it became evident that the coding was redundant, cluttered, and over-complex. These weaknesses stem from its dependence on **mixed-content elements**, all of which have a large number of child elements.

So what's wrong with mixed content?

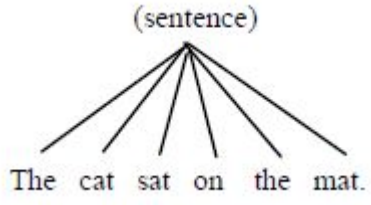
The problems with TEI stem from a basic **structural feature of XML** itself: that a precise hierarchical structure can be maintained only by using standard-content elements (those with only other elements as children), but not with mixed-content elements (those with other elements plus text as children), because the latter have almost no internal constraints. Consequently, a precise structure cannot be maintained using only such elements, especially if they have a large number of children.



The problem can be illustrated in linguistic terms, where structures of great complexity can be specified by hierarchies consisting entirely of **binary-branching nodes**: in XML terms, of elements all having just two children. For example, the top-level structure of 'the cat sat on the mat' would show the division between subject and predicate:

These two child nodes would then be sub-divided: on the one hand, into the article with noun, and, on the other, the verb with its prepositional phrase. The division can be repeated until every word is given a unique position in the hierarchy.

The precision of the structure derives from the limitation of choice to two children. If, instead, we allowed all the words to be children of the top-level clause, we would have a 'flat' structure, which is less informative, because it does not specify how each of the child nodes should be ordered. The loss of a structural heirarchy means that the syntactic information will have to be marked by extra annotations (in XML, by **adding 'attributes'** to each element).



In XML terms, the second structure corresponds to mixed-content elements, where the children can be repeated any number of times, and their order is not specified. A schema composed purely of mixed-content elements therefore has an inherent weakness.

Unfortunately, this was the structure chosen by the TEI. In 2003, every element is mixed-content, and every one of them has over 60 children. The result may be seen in the element "tr", designed to contain a translation.

```
<!ELEMENT tr (#PCDATA | abbr | address | date | dateRange | expan | lang | measure | name | num | s | time |
timeRange | add | corr | del | orig | reg | sic | unclear | oRef | oVar | pRef | pVar | distinct | emph | foreign | gloss |
hi | mentioned | soCalled | term | title | ptr | ref | xptr | xref | seg | bibl | biblFull | biblStruct | cit | q | quote | label |
list | listBibl | note | stage | text | anchor | gap | alt | altGrp | index | join | joinGrp | link | linkGrp | timeline | cb | lb
| milestone | pb)* >
```

This coding signifies that the element "tr" may contain any amount of text inside it (PCDATA) and also 63 child elements, which may appear any number of times, in any order, interspersed in any way with the text.

In TEI Release 4, the situation had become worse: "tr" now contained 106 children. In the latest P5 Release, considerable reorganisation has taken place, and the "tr" element appears to have been eliminated, but the similar "def" (definition) element has 55 children, still showing the same dependence on unmanageably-large mixed-content elements. This is particularly disappointing because it discards at one stroke what is generally thought to be the greatest strength of XML: that it can **add structure** to a text.

We can, of course, recover the lost structure by adding constraints to the XML, either through variable attributes applying to each element, or through the styling (by CSS Cascading Style Sheets or XSL Extensible Stylesheet Language). However, all such fix-its **create even more complexity**, and so require more work than a structure which was clearly defined at the start. As noted above, it seemed more efficient to us to **compose** our lexicon in XML, rather than having a two-stage operation, with lexicographers writing in word-processing documents, and IT specialists encoding their work into XML. And this seemed to call for a structure which mirrored our lexicographic methodology as closely as

possible, and so could be learned very easily by the lexicographers, so that they would be able to type directly into the editing software.

An alternative to TEI: self-design

How easy is it to design one's own XML structure? In our experience, the hard work was less in designing the digital structure than in finalising the lexicographic principles. Once that was done, the structure was relatively simple to create, because, the more structure was coded into the elements themselves, the less heavy-lifting remained to be done by XML attributes or XSL styling. It proved straightforward to configure a suitable structure, because our lexicographic requirements turned out to be expressible in less than 100 elements.

> We found it possible to circumvent the structural weakness of mixed-content elements by means of a **hierarchical structure of elements**, giving standard-content elements at the top levels, so as to give maximum constraints there, and **progressively introducing mixed-content** and text-only elements at the lower levels, where we needed most freedom to compose continuous text. And by using XSLT to add inter-element punctuation and whitespace, we found that the XML authoring environment actually reduced the writers' work-load. Self-design enabled us to configure a structure which made the work of the authors as simple as possible: it is now much easier for us to compose in XML than it had been in Word. The method, which I call **tag and type** (previously "author-tagging"), is described in Fraser (2005).[2]

In this system, the writers select and enter the elements as they compose. Each writer has an XML editor set in a 'tags-on' view, which shows only those elements which are available at any given point in the entry. The process of composition is therefore **a series of choices**, presented in turn. The first choice is to select an entry **structure for the part of speech**: a noun, name, adjective, verb, adverb or preposition. After selecting the part of speech, the next choice available is the **headword**, then the **inflection, part of speech**, and so on, and the writer selects each in turn and enters text into the chosen element. The entry is therefore built up in the editing window in its natural order, with the text visible and automatically formatted, and the tag positions are also visible, so the writer can move about easily in the entry, making additions and corrections. At any time, a PDF view can also be generated.

> The advantage of such an XML environment, compared with a database interface, is that it's more **natural to tag and type continuous text**, than to type into a modular layout of boxes, and so we can concentrate on meaning and writing style as we compose. This is possible only with a **highly-constrained XML element structure**. Though the writers have to select the correct 'tag' (from a maximum of 8 available at any given point), they can then concentrate totally on writing: there are **no variable attributes** to be set, and **all intra-element text formatting and inter-element punctuation and white-space is set automatically** by the XSLT stylesheet. The XML element structure alone leads the writer through the writing process for each entry. This method therefore serves not just as a writing tool, but also as a **learning tool** for new members of the team: we can use the XML to teach our lexicographic methodology.

Another advantage of the tailor-made approach is that the **development** of the DTD and the associated XSL stylesheet depends on the the development of the lexicographic approach. The two processes can be undertaken in parallel, with the **writing team fully involved in the digital development**, by writing new entries throughout the period, and suggesting improvements to the digital output. Our writers could be fully involved in the XML design process, because that involved configuring a **small number of XML elements**, each of which was named according to our own **preferred terminology** (another reason not to use an 'off-the-peg' system).

Using TEI and other existing schemas

Due to the loss of structure and the extra complexities which the TEI schema introduces into the authorial environment, as described above, it does not seem to provide a suitable structure for a new dictionary. However, the inclusivity of its structure does mean that it will be capable of capturing the structure of **dictionaries which have already been written**, and just need to be encoded into XML. Clearly, many dictionaries do use TEI, and

in a number of publications have described how it was possible to adapt the TEI structure to their lexicographic requirements.

> However, none has demonstrated the advantages of doing so, rather than designing their own schema. I'd be happy to learn that the TEI system does have advantages, but so far these are not evident, when compared with **alternative schemas** which have been designed for existing print dictionaries, including the SGML structure developed for the *Oxford English Dictionary*. Its development is described by Tompa (1996) and Elliot (2000), and its economical and uncluttered structure mirrors the remarkably clear lexicographic format which it was designed to encode.[3]

One benefit of an off-the-peg system is likely to be **cross-compatibility** in linking to other material. This was a major consideration for our own project: we were initially concerned that we might have problems with the digital edition of our lexicon, because that will be posted on Perseus, who use TEI-conformant coding. But bilingual lexicography always relies on the possibility of translation, and Perseus tell us that linking will in fact be possible.[4] Our choice of a self-designed system appears to be compatible with the digital as well as the print platform.

> In case others may be interested in the practical details of this approach, the DTD and XSLT Stylesheet will be archived on the University of Cambridge D-Space Digital Repository, available on an open-source basis.

The future and the advantage of an author-centred approach

I'm keen to see the end of a perceived division between humanities academics as scholars – in literature, linguistics, epigraphy, archaeology and so on – or as IT specialists. Because **we are all involved in publishing**, we all need to be comfortable with digital text-formatting, and the XML environment is a very accessible one. I'm deeply grateful to programmers who encouraged me to engage with the technical details of text composition, because the experience has shown me that the best results can be achieved when the technical details are designed to serve the scholarly task in hand: by **configuring our IT systems to our academic methodology**, rather than the other way round. And the principles of XML are not difficult to grasp, if you know how web pages are tagged, and XSL, though rather more challenging, can be regarded as an exercise in organising conditional clauses.

> For our publishing environments to encompass **digital as well as print output**, we need to give **structure as well as format** to our texts, and we should eliminate features which weaken it. However, TEI has many users who are committed to it, and I would not expect current projects to abandon it. I do however suggest that the TEI consortium might consider making a further and more radical revision of the Dictionary schema, introducing top-level standard-content elements, and, for mixed-content elements, reducing the number of variable attributes and the maximum number of children to around 10, so as to **create a structured hierarchy** and reduce redundancies. I also suggest that lexicographers who have not yet chosen an XML system might wish to design a structure which is tailored to their scholarly methodology, and so can provide an efficient vehicle for it.

---

[1] This may help explain why, of the pre-existing structures we examined, the OED's schema came closest to meeting our requirements, because it is also a historical and literary dictionary, albeit a monolingual one.

[2] The original name was intended to emphasise that the lexicon writers choose the XML tags themselves, rather than the tags being added to the composed text. The name has been changed to highlight the order in which the writers work: first selecting the appropriate tag, and then composing their text.

[3] In the Preface of the *Oxford English Dictionary*, Second Edition, 1989, the editors Simpson and Weiner described the development of the SGML system in these terms: "The structure designed by Sir James Murray and used by him and all his successors for writing

dictionary entries was so regular that it was possible to analyze them as if they were sentences of a language with a definite syntax and grammar. They could therefore be parsed."

[4] I'm most grateful to Bridget Almas of Perseus for explaining the technical details: our source XML will be transformed into the Perseus TEI-Analytics schema (a subset of TEI P5) using the Abbot toolset, and scripts will be written to identify cross-reference points between the lexicon and the Perseus material, and to insert the tags for linking.

| Top of Page | Bruce Fraser, April 2011 | Page Up |
