The TEI and XML

DOI : 10.4000/books.oep.680
Éditeur : OpenEdition Press
Lieu d'édition : Marseille
Année d'édition : 2014
Date de mise en ligne : 11 avril 2014
Collection : Encyclopédie numérique
ISBN électronique : 9782821834606

http://books.openedition.org

Référence électronique
The TEI emphasizes what is common to every kind of document, whether physically represented in digital form on disk or memory card, in printed form as book or newspaper, in written form as manuscript or codex, or in inscribed form on stone or wax tablet. This continuity facilitates the migration of text from older manifestations such as print or manuscript to newer ones such as disk or display. Hence, the TEI view of what text actually is is largely conditioned by what text has been in the past, without however compromising too greatly what text may become in the future. It attempts to treat all kinds of digital document in the same way, whether they were ‘born digital’ or not.

As a consequence, the TEI framework provides a useful way of thinking about the nature of text: it constitutes a kind of encyclopaedia of generally-agreed textual notions. In this brief guide we will try to exemplify some of these notions, using the vocabulary defined by the TEI in its Guidelines.

At present TEI documents in digital form are expressed using a very widely-used formal encoding language called XML, the extensible markup language, first published by the World Wide Web Consortium (W3C) in 1998, but with origins in the document preparation systems of the 1980s. XML provides a simple way of representing structured data as a linear stream of character data, and of labelling particular parts of that stream with named tags to indicate structural function or semantics. Because it has become such a pervasive technology, many excellent introductory guides are available elsewhere and we will therefore assume in the reader a basic understanding of such key concepts as element, attribute, schema, name space, briefly glossed below for the sake of intelligibility. The TEI Guidelines include a ‘Gentle Introduction’ to XML, which may be useful to the novice, but many other tutorials on the subject are to be found.
Here is an example of a minimal XML document:

```xml
<?xml version="1.0"?>
<doc xmlns="http://example.org/namespace">
  <p n="1">This is a paragraph.</p>
  <p n="2">This paragraph mentions <placeName>Bristol</placeName>.</p>
</doc>
```

The first line of an XML document always takes the form shown above: a special kind of instruction indicating that what follows is an XML document conforming to the version of the XML standard indicated (in this case, version 1.0). An XML document consists of a sequence of human-readable characters, with no special additional codes or binary data. The characters `<` and `>` are used to mark the start and end of *tags* within this sequence. A tag may be a start-tag (such as `<p>`) or an end-tag (such as `</p>`). A tag always begins with a name (doc, p, placeName in the above example) and may also contain attribute specifications (such as `n="1"`).

The purpose of a start-tag is to mark the point in the sequence of characters at which some *element*, of a type indicated by the tag name, starts, and the purpose of an end-tag is to mark where that element ends. The purpose of an attribute specification is to add some extra information about an element occurrence beyond its name. In the above example we have an element named `<doc>` which contains two `<p>` elements. The `<p>` elements both have an `@n` attribute which supplies a number, and both contain plain text. The second `<p>` element also contains an element called `<placeName>`.

An XML document like this is said to be *well-formed* if it respects the syntax exemplified here, with start- and end-tags both present and correctly nested. But the XML standard says nothing at all about how elements or attributes should be named (unlike, for example, HTML which defines a specific set of tags that must be used in a particular way in all documents), much less what their names mean. We may guess that the `<p>` elements above are marking up numbered paragraphs, but there is nothing in the XML representation to warrant that
assumption - they could just as well mark up pages, or entries in a glossary, or lines of verse. If therefore I find another document containing <p> elements, how will I know whether they have the same function? The function of the @xmlns attribute above is to help solve this problem by supplying a default for what is called the namespace of all the elements contained by the <doc> element.

It is not unusual to find elements from many namespaces in a single document: for example, a document containing music notation, vector graphics, and text all represented in XML might use tags from three different namespaces, one for the musical elements, one for the graphical ones, and one for the textual ones. A namespace is a way of labelling a group of elements: in our example, its use makes clear that the <p> elements here are different from any <p> elements defined by some other namespace.

The reason for introducing tagging into a document is to label and organize it for machine processing. If the paragraphs are clearly marked, then a formatter can lay them out properly. If the place-names are clearly marked, a program can automatically pick them out to make a geographical index. But this can really only be done reliably if we have some control over how tags are introduced into the document and where they appear. XML technology provides this additional level of control by means of what is called a schema, a kind of combined lexicon and grammar for valid XML documents. We noted above that an XML document is said to be well-formed if it respects the syntactic rules of the XML standard. It may, optionally, also be said to be valid if the tags it contains conform to a schema.

A schema specifies a set of element names, the names and datatypes of any attributes associated with them, and rules about the contexts in which they may legally appear. A schema for our simple example above will say that elements named <doc>, <p>, <placeName> etc exist. It may also specify that <p> elements may appear within <doc> elements, that <placeName> s may appear within <p> s, that the attribute @n must have a numeric value etc. Note however that an XML schema still has no way of specifying that the tag <placeName> indicates the name of a place, or what we mean by "a
place": such additional semantic constraints must be specified elsewhere, for example by documentation such as that provided by the TEI Guidelines.

The TEI provides names and definitions for many hundred tags, together with rules about how they may be combined. More exactly, the TEI Guidelines define some five or six hundred different concepts, along with detailed specifications for the XML elements and element classes which may be used to represent them. Most, if not all, TEI documents need to use only a small amount of what is provided. It is therefore somewhat misleading to think of the TEI as a single monolithic schema. To facilitate interoperability, every TEI document uses components taken from the same mammoth schema, but most TEI projects use quite small subsets of it, and a well-organised project will generally have its own customized documentation identifying that subset.

Software such as the web application Roma can be used to select amongst the TEI specifications and generate from them a schema appropriate to the needs of your own project, or you can simply assemble a schema by hand. We discuss this topic in more detail in a subsequent chapter. The TEI’s Guidelines, freely accessible from the web site at http://www.tei-c.org/Guidelines, constitute a complete reference manual for these concepts, combining technical specification with detailed discussion of how they are meant to be used.

There is a very wide range of software tools available to create, transform, and process XML documents in general, or TEI XML in particular. This is a very large and fast-moving topic which is not treated here, though you will find some useful pointers on the TEI website and the TEI Wiki.