What is the Text Encoding Initiative ?

1. The TEI and XML

1 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.

2 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

3 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.

4 Here is an example of a minimal XML document:

```
<?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>
```

5 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) or an end-tag (such as). 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 elements. The elements both have an `@n` attribute which supplies a number, and both contain plain text. The second element also contains an element called <placeName>

6 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 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.

7 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.

8 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 what 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 contain conform to a schema.

9 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.

10 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.

11 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 chapter #ODD below. The TEI's Guidelines, freely accessible from its web site at 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.

12 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.

2. The structural organization of a TEI document

13 All TEI documents are organized in a similar way, no matter what kind of original they represent. We will introduce some of the most commonly encountered variations on this structure by means of some simple examples.

2.1 Header, Text, and Divisions

14 To begin at the beginning, every TEI document (represented by means of a `<TEI>` element) has at least two parts: a *header* (represented by means of a `<teiHeader>` element) containing metadata describing the document, and the text itself (usually represented by a `<text>` element). For example :

```
<TEI xmlns='http://www.tei-c.org/ns/1.0'>
<teiHeader>
<!-- metadata describing a text  --> </teiHeader>
<text> <!-- a representation of the text itself --> </text>
</TEI>
```

Like every other TEI XML document, this one explicitly states that the elements it contain are by default to be understood as coming from the TEI namespace `"http://www.tei-c.org/ns/1.0"`.

15 We describe the header in more detail in section #HDR below; for the moment we note simply that a minimal header must contain information identifying the document itself (in the <titleStmt>), information about how it is distributed or published (in the <publicationStmt>) and some indication of its origins (in the <sourceDesc>). The <text> element is used to hold an encoded version of the text itself, in which its structure is represented by elements such as <front> (for prefaces etc.) <body> (for the body of the text proper) and <back> (for any appendixes etc.). Within these components, we may also represent further subdivisions such as volumes, parts, chapters etc. using the element.

16 For example, here is the start of a minimal TEI version of a famous novel, as it might be distributed by an imaginary digital publisher:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
<fileDesc>
<titleStmt>
<title>The life and opinions of Tristram Shandy, Gentleman: TEI
edition</title>
</titleStmt>
<publicationStmt>
<publisher>Web Head Press</publisher>
<date>2013</date>
</publicationStmt>
<sourceDesc>
<p>Transcribed from the first edition, 1708</p>
</sourceDesc>
</fileDesc>
</teiHeader>
<text>
<body>
<div type="volume" xml:id="TS01">
<div type="chapter" xml:id="TS0101">
<head>Chap. I</head>
<p>I wish either my father or my mother, or indeed both of them, as they
were in duty both equally bound to it, had minded what they were about
when they begot me; ...</p>
<!-- remainder of chapter 1 here -->
</div>
<div type="chapter" xml:id="TS0102">
<head>Chap. II</head>
<p>—— Then, positively, there is nothing in the question, that I can see,
either good or bad.—— Then let me tell you, Sir, it was a very
unseasonable question at least ...</p>
<!-- remainder of chapter 2 here -->
</div>
<!-- remaining chapters of volume 1 here -->
</div>
<!-- remaining volumes of work here -->
</body>
```

```
</text>
</TEI>
```

17 In this example the body of a text contains smaller divisions, themselves containing sub-divisions. Such textual subdivisions have different names in different cultures and in different kinds of document; many western cultures use names such as “section,” “part,” “book,” or “chapter” (or their equivalents) for these sub-parts of the body, but often in inconsistent or mutually incompatible ways. What is called a “part” within a “book” in one text may be called a “section” within a “chapter” in another, or a “book” within a “part” in a third. Hence, the TEI proposes a single element `<div>` for any such structural subdivision of the body of a text.

18 The `<div>` element can carry a number of attributes to indicate its function and its properties more exactly. Here, we have used the `@type` attribute to characterize or classify the content of the element, thus distinguishing the `<div>` elements containing “volumes” from those containing “chapters.” We have also used the `@xml:id` attribute to provide a unique identifier for each division of the novel.

19 This same kind of hierarchic structure may be used for any kind of text. For example, an epic poem divided into books might tag each book as a `<div type='book'>`; a play divided into acts and scenes might tag them `<div type='act'>` and `<div type='scene'>` respectively; and so on. The values available for use with the `@type` and `@xml:id` attributes are not defined by the TEI, but are chosen by the encoder. For `@xml:id` the values used are arbitrary codes unique to the element on which they appear: they provide a way of labelling the element concerned so that other parts of this or other documents can point to it directly. For `@type` the values are also arbitrary codes, chosen by the encoder to indicate the function of the `<div>` concerned. Some commonly encountered values such as `"chapter"` or `"volume"` are suggested in the Guidelines, but it is left to individual projects using the TEI to define their own taxonomy. It is however easy both to document and to enforce any chosen taxonomy by means of a TEI customisation, as discussed further in chapter #ODD.

20 The `<div>` element is so named because it is a *division*; it should not therefore be used for something which is complete in itself. If the body of a text is undivided there is no need to provide a single `<div>` element to contain it. Instead, the `<body>` element may contain directly one or more of the elements described in the next section.

21 A <text> is typically something like a book or an article, but may be anything which it is convenient to regard as a discrete but complete textual object, such as a single poem or archival document, or something as small as a postcard. As an alternative or in addition to the <text> element, the TEI also provides a <facsimile> element which could be used to provide a complementary visual representation, for example as a series of digitized page images. And the TEI also provides ways of representing collections of such things, as discussed in the final section of the next chapter.

3. Varieties of textual structure

22 In the TEI view, texts and their divisions may contain a fairly limited range of "structural" components. These include such things as the headings or preliminary matter at the start or end of a division, and a number of basic building blocks which are characteristic of prose, verse, or drama. Prose, for the most part, consists of paragraphs or lists, marked using the elements <p> or <list> respectively. Verse consists of verse lines, marked using the <l> element, or sequences of verse lines marked using the <lg> (line group) element. In drama, an additional building block is provided in the form of the <sp> (speech) element, which combines a <speaker> element with one or more <p> or <l> elements depending on whether the drama is in verse or prose.

3.1 A journal article

23 The body of an academic article, for example, might be encoded with a structure like the following :

```
<body xml:lang="en">
   <div type="section">
    <head>Introduction</head>
    <p>We recommend the use of TEI markup as a means of representing scientific
prose. It has a number of
    practical advantages: </p>
    <list>
     <item>TEI markup is easy to add;</item>
     <item>TEI markup is widely understood;</item>
     <item>TEI markup is easy to convert to other formats.</item>
    </list>
    <p>TEI markup also has some scientific advantages, which we discuss in section
<ptr target="#SEC3"/>.</p>

  <!-- further introductory paragraphs here -->
   </div>
   <div type="section">
    <head>Origins of the TEI</head>
    <p>The Text Encoding Initiative was born in 1987 .... </p>

  <!-- many more paragraphs here -->
   </div>
   <div xml:id="SEC3">
    <head>Scientific properties of TEI markup</head>
    <p>TEI markup expresses a view of the nature of text ... </p>

  <!-- many more paragraphs here -->
   </div>
  </body>
```

Note that this markup indicates only the organization of the document. It says nothing about how it should be visualised on screen or on paper. It indicates that there are sections which have titles, and which contain paragraphs and lists. It distinguishes the items in the list, but it does not specify whether they should be prefixed with a bullet or a dash or a number. The `<ptr>` element indicates the existence of a cross reference from one part of the document (the location of the `<ptr >` element) to another (the `<div>` element entitled “Scientific properties...”). The value of the `@xml:id` attribute introduced above specifies the target of the cross reference, but it does not

specify whether this should be realised as (say) an HTML link, or some added text (such as the section number) or both. Of course, it is not hard to imagine how we might display these document components using a formatting language such as Word, HTML, or LaTex but that is not the primary purpose of this markup. Because these aspects are all left unspecified in the encoding, the same document can be re-used in different processing contexts.

3.2 A poetic text

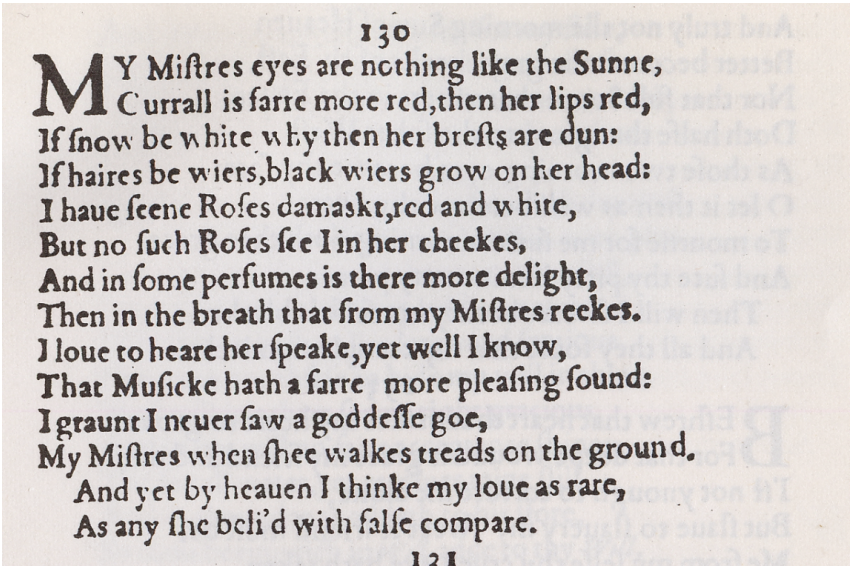
26 Our second example shows a famous Shakespearian sonnet, transcribed from a specific print copy:

```
<lg type="sonnet">
 <head>130</head>
 <l>My Mistres eyes are nothing like the Sunne,</l>
 <l>Currall is farre more red, then her lips red,</l>
 <l>If snow be white why then her brests are dun:</l>
 <l>If haires be wiers, black wiers grow on her head:</l>
 <l>I have seene Roses damaskt, red and white,</l>
 <l>But no such Roses see I in her cheekes,</l>
 <l>And in some perfumes is there more delight,</l>
 <l>Then in the breath that from my Mistres reekes.</l>
 <l>I loue to heare her speake, yet well I know.</l>
 <l>That Musicke hath a farre more pleasing sound:</l>
 <l>I graunt I never saw a goddesse goe,</l>
 <l>My Mistress when she walkes treads on the ground.</l>
 <lg type="couplet">
  <l>And yet by heaven I thinke my love as rare,</l>
  <l>As any she beli'd with false compare.</l>
 </lg>
</lg>
```

27 Again, this markup captures only the structure of the sonnet: indicating the individual verse lines of which it is composed, and marking explicitly the couplet at its end. Because individual lines are distinguished in the encoding, rather than being treated as if they were paragraphs, or accidents of formatting, a metrical analysis of the poem can automatically be generated.

28 Comparing it with the following digital image of the original source we can see that although the original spelling has been retained, this encoding has chosen to ignore such layout features as the use of dropped caps or the indentation associated with the couplet. It has retained the original spelling, but silently modernised some of the typographic variation, such as the long S, or the ligatured letters.

29 **Figure 1: Sonnet 130 from Shake-speares Sonnets. Never before Imprinted (1609)**



130

MY Miſtres eyes are nothing like the Sunne,
Currall is farre more red,then her lips red,
If ſnow be white why then her breſts are dun:
If haires be wiers,black wiers grow on her head:
I haue ſeene Roſes damaskt,red and white,
But no ſuch Roſes ſee I in her cheekes,
And in ſome perfumes is there more delight,
Then in the breath that from my Miſtres reekes.
I loue to heare her ſpeake,yet well I know,
That Muſicke hath a farre more pleaſing ſound:
I graunt I neuer ſaw a goddeſſe goe,
My Miſtres when ſhee walkes treads on the ground.
And yet by heauen I thinke my loue as rare,
As any ſhe beli'd with falſe compare.

131

We will see below how this encoding could be enhanced to produce an acceptable modern reading version as well as the quasi-diplomatic version shown here.

3.3 A play text

30 Our third example shows how we might encode the structure of a dramatic text : in this case, the end of Beckett's Waiting for Godot.

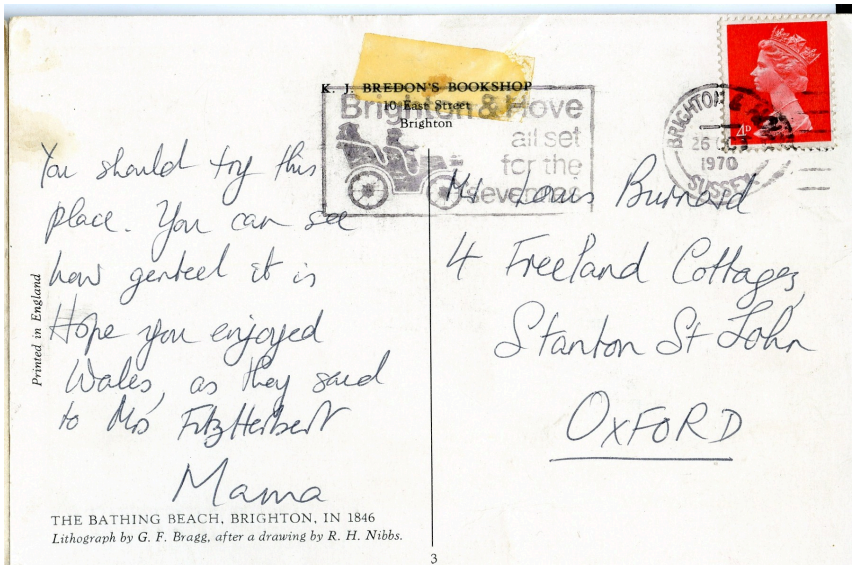
```
<div type="scene">

<!-- ... -->
 <sp>
  <speaker>Vladimir</speaker>
  <p>Pull on your trousers.</p>
 </sp>
 <sp>
  <speaker>Estragon</speaker>
  <p>You want me to pull off my trousers?</p>
 </sp>
 <sp>
  <speaker>Vladimir</speaker>
  <p>Pull <emph>on</emph> your trousers.</p>
 </sp>
 <sp>
  <speaker>Vladimir</speaker>
  <p><stage>(realizing his trousers are down)</stage>. True</p>
 </sp>
 <stage>He pulls up his trousers</stage>
 <sp>
  <speaker>Vladimir</speaker>
  <p>Well? Shall we go?</p>
 </sp>
 <sp>
  <speaker>Estragon</speaker>
  <p>Yes, let's go.</p>
 </sp>
 <stage>They do not move.</stage>
</div>
```

31 In this encoding, we have introduced a few more TEI elements, to enable us to distinguish the stage directions (`<stage>`) from the individual speeches of the characters (`<sp>`). Note that a stage direction can appear both within and between speeches and that a speech always contains both an identifying label to show who is speaking (`<speaker>`) and a paragraph (`<p>`) to contain what they say. In a verse drama, of course, the speeches would probably contain a sequence of `<l>` elements.

3.4 A postcard

32 The TEI is widely used for existing literary or formally published works. However it can also be used for entirely different kinds of document, such as authorial manuscripts, archival papers, or any other kind of informal writing. Our fourth example shows how we might use the TEI to encode the structure of this postcard :

Figure 2: A picture postcard (verso)



33 All postcards have two sides, a recto and a verso: we represent these as `<div>` elements. Within the side shown above, we can see that there is a clear distinction between the part carrying the message, to the left, and the part relating to the sending of the card, containing various stamps and an address. Here is one possible encoding, which uses the same `<div>` and `<p>` elements we have already seen, together with some additional more specialised elements and attributes:

```
  <div type="verso">
   <div type="message">
    <p>You should try this place. You can see how genteel it is</p>
    <p> Hope you enjoyed Wales, as they said to Mrs Fitzherbert</p>
    <signed>Mama</signed>
   </div>
   <div type="destination">
    <ab>
     <stamp type="publicity">Silhouette of vintage car and slogan
<mentioned>Brighton & Hove all set
      for the seventies.</mentioned></stamp>
     <stamp type="postmark">Circular mark specifying <mentioned>Brighton & Hove -
Sussex</mentioned>
      and <date when="1970-10-26">26 Oct 1970</date></stamp>
     <stamp type="postage"> Machin design. 4d, vermillion. </stamp>
    </ab>
    <address>
    <addrLine>Mr Louis Burnard</addrLine>
    <addrLine>4 Freeland Cottages,</addrLine>
    <addrLine>Stanton St John</addrLine>
    <addrLine>OXFORD</addrLine>
   </address>
   </div>
  </div>
```

34 The `<signed>` element used to enclose the signature block is one of several specialised elements provided by the TEI for material such as signatures or headings which can appear at the beginning or end of a `<div>`. It seems useful to distinguish it from the rest of the message, since other such "signing off" phrases or *formules de politesse* contain a different, more formal, kind of language from the rest.

35 In the other division, we have distinguished three different kinds of stamp: the postage stamp itself, the postmark indicating when and where the card was posted, and an additional publicity stamp promoting the Brighton and Hove Vintage Car Rally. The TEI `<stamp>` element is intended to contain descriptive text about any kind of stamp; where our description includes words actually appearing as part of the stamp we use the TEI `<mentioned>` element. In the case of the postmark,

we have also distinguished the date given. Much of our research on an archive of such encodings will want to search and group cards reliably by the date on which they were posted. Dates as given in postmarks often use different formats and may be incomplete: we therefore add to the encoded date a normalised value, supplied on the `@when` attribute. Finally, we have given the address to which the card was sent, simply dividing it up into lines.

36 In a more detailed encoding we would probably distinguish the names of people and places, perhaps adding geographic co-ordinate information for the place names mentioned. We might wish to add some explanation of the little joke about "Wales." We would probably want to distinguish the formulaic parts of the message, such as its opening and closing lines, from the content. And we would also consider how best to record the textual information printed on the back of the card — the title and the name of its publisher for example. The scope of such metadata, which provides useful information about how the card was produced and used is very large and it is hard sometimes to know where to stop. For example, the yellow stain visible on the image suggests that this card was once attached to something by means of cheap adhesive tape. If this were a very rare historical artefact such evidence of provenance might be of considerable importance and would need to be encoded using (for example) the `<damage>` element.

3.5 A minimally structured text

37 Finally, if this all seems too much, we should note that the TEI can also support a very simple structural view in which all these conventional, semantically-charged, components such as paragraphs, speakers, verse lines, etc. are elided or ignored in favour of a neutral segmentation based on orthographically-defined sentences. In such a view, we might decide to keep the main divisions of the text, but then simply split it up into linguistically convenient segments at every appropriate punctuation mark, using the generic `<s>` (segment) and `<ab>` (anonymous block) elements. To reprise our first example:

```
  <body xml:lang="en">
   <div type="section">
    <ab>
     <s n="1">Introduction</s>
     <s n="2">We recommend the use of TEI markup as a means of representing
scientific prose.</s>
     <s n="3">It has a number of practical advantages:</s>
     <s n="4">TEI markup is easy to add;</s>
     <s n="5">TEI markup is widely understood;</s>
     <s n="6">TEI markup is easy to convert to other formats.</s>
     <s n="7">TEI markup also has some scientific advantages, which we discuss in
section <ptr target="#SEC3"/>.</s>
    </ab>
   </div>
  </body>
```

38 Although such markup obviously lacks information present in the richer version we presented before, and is thus less generally useful, its simplicity and regularity means that it is much easier to process for such comparatively mechanical tasks as vocabulary analysis or linguistic enrichment. As ever, to get the best out of the TEI requires us to think carefully about our priorities before making our encoding decisions.

3.6 Composites

39 A TEI text may be *simple* (for example, a single book) or *composite* (for example a collection or anthology). In either case the main body of the text may be preceded or followed by additional distinct matter (title pages, prefaces, dedications, indexes etc.). The TEI proposes a mandatory element `<body>` for the main part of a text, and optional elements `<front>` and `<back>` to group additional matter preceding or following the body respectively. In a composite text, the "body" of the text is represented by a *group* element, which can contain multiple `<text>` elements, or groups of them, as in the following schematic:

```
<TEI>
  <teiHeader>

 <!--[ header of a composite text ]-->
  </teiHeader>
  <text>
   <front>

 <!--[ front matter for the composite text ]-->
   </front>
   <group>
    <text>
     <front>

 <!--[ front matter for the first text in the composite ]-->
     </front>
     <body>

 <!--[ body of the first text in the composite ]-->
     </body>
     <back>

 <!--[ back matter for the first text in the composite ]-->
     </back>
    </text>
    <text>
     <front>

 <!--[ front matter for the second text in the composite ]-->
     </front>
     <body>

 <!--[ body of the second text in the composite ]-->
     </body>
     <back>

 <!--[ back matter for the second text in the composite ]-->
     </back>
```

```
    </text>

<!--[ more texts, simple or composite ]-->
   </group>
   <back>

<!--[ back matter of the composite text ]-->
   </back>
  </text>
 </TEI>
```

41 A structure like this would be appropriate for encoding a pre-existing anthology of verse by many authors but a single editor.

42 In this model there is just one TEI Header for the whole of the composite text. The TEI also allows for a composite which represents a collection of documents originally distinct but put together by the encoder for some purpose: language corpora are typical examples. Each constituent document in a language corpus has its own metadata, represented by its own `<teiHeader>`, but there is an additional layer of metadata relating to the corpus as a whole. This is encoded in TEI using a structure like the following:

43
```
<teiCorpus><teiHeader>

  <!--[metadata relating to the whole corpus]-->
</teiHeader><TEI>
    <teiHeader>
  <!--[metadata relating to the first text in the corpus]-->
</teiHeader>
    <text>

  <!--[first text in the corpus]-->
    </text>
   </TEI><TEI>
    <teiHeader>
  <!--[metadata relating to the second text in the corpus]-->
</teiHeader>
    <text>

  <!--[second text in the corpus]-->
    </text>
   </TEI></teiCorpus>
```

44 A structure like this would be appropriate for encoding a newly-constructed collection of postcards or other independent textual objects.

4. The TEI cornucopia, part one

45 In this and the following chapter we discuss and exemplify some of the elements defined by the full TEI. The elements discussed in the present chapter are likely to be useful in almost every kind of encoding project; those discussed in the next are more specialized. It should be emphasized that all we aim to do here is to give some sense of the rich variety offered by the TEI; for complete information the Guidelines should always be consulted.

4.1 Milestones

46 We noted above that the TEI is most often used to represent the organization of a book into components such as chapters, paragraphs, etc. or of a poem into verses and stanzas. But books and printed poems also exist as physical objects, which are typically made up of pages with verso

and recto surfaces. Why does the TEI not provide an element <page> to represent each page of a book? Surely it is at least as important to know on which page a sentence begins as it is to know that it is in the fifth paragraph of the second chapter of the ninth subsection of a work? Surely we would like to be able to display the text paginated on screen in the same way as it was paginated in the original source?

47 The answer to these perfectly reasonable questions is perhaps surprisingly complex. In fact, the TEI does provide an element to mark up page boundaries (<pb>), but it is an empty element which should be placed at the beginning of the text transcribed from each page:

```
<body xml:lang="en">
 <pb n="42"/>
 <p>This paragraph begins on the page numbered 42...
<!-- lots more text here -->
  <pb n="43"/>

<!-- yet more text here -->
... and finishes half way down the page numbered 43.</p>
  <p>This much shorter paragraph begins and ends on page 43. </p>
</body>
```

48 A close study of the above example may help explain why a <page> element that actually encloses all the text on a page (rather than an empty <pb> which simply flags the beginning of a page) is not possible. In XML and similar markup languages, it is an absolute requirement that the elements defined by the markup tags should be properly nested one within another. If we start a <page> element, and then within it start a <p> element, the syntax of the language requires us to close that <p> before we can close the <page>, even though it is an observed fact of life that paragraphs frequently span across page boundaries.

49 Users of XML (and before that SGML) have been trying to find better ways of representing textual structure without falling foul of this constraint for decades, and we don't intend to debate the “overlap problem” in any more detail here. We note simply that one well-established TEI way of addressing this problem takes the form of the “milestone” tags, of which <pb> is probably the most frequently encountered example. Whereas normal tags in an XML document mark explicitly both the start and the end of an element and thus always enclose something, a milestone tag marks

simply a point at which something changes, but do not enclose anything. Milestone number 42 on the road from London to Bath marks the point at which the forty-second section of that road begins. We can guess that if we keep going we will eventually reach milestone number 43, which will mark the point at which the forty-third section starts, and hence (since a stretch of road cannot be in two sections at once) the point at which the forty-second section finishes, but it is not explicitly marked.

50 The elements <lb> (line break), <cb> (column break), and <gb> (gathering or folio break) are all milestones in this sense. In addition the TEI provides a generic <milestone> element which can be used to mark any kind of unit not otherwise provided for. As an example consider the practice (common in the 19th century) of publishing novels in serial form. We might wish to mark the boundary of the individual part issues, even though these do not necessarily fit well with the novel's internal organization as a series of chapters. (Dickens, for example, was not above closing a part issue with a mid-chapter "cliff-hanger"). A <milestone unit='serialPart' n='12'/> can be used to represent the point at which "serialPart" number 12 begins, independent of the structure of and <p> elements used for the text itself.

51 This generic element can be used for any kind of shift, of course, not simply structural units. A narratological analysis might be represented using <milestone> elements to mark points at which the narrative voice changes; a stylistic analysis might use them to mark points of stylistic variation. There are also other specific elements which behave in a milestone-like fashion for example to indicate changes of voice quality in transcribed speech, or changes of hand in transcribed manuscript.

4.2 Languages and writing systems

52 Unless it is an empty milestone every element in a TEI document can contain one of plain text, "element content," or (by far the most frequent case) "mixed content." Plain text consists of encoded characters represented using the Unicode standard; element content consists of other XML elements; and mixed content combines the two. TEI elements and attributes are used to associate semantic and other properties with the content so that a processor can treat it correctly in a variety of situations. One particularly important property is the human language in which the textual content is expressed; another is the way the textual content is displayed or formatted.

These properties are potentially important at almost every level: we might want to say that a whole document uses a specific language, or that just a few words here and there do; we might want to say that the whole of a document was typeset in a Roman font of a given size, and then indicate those parts of it which use some other font or size. Because such information is potentially applicable to every element, the TEI proposes *global attributes* to specify it.

4.2.1 Languages

53 The global `@xml:lang` attribute is the recommended means of specifying the human language in which the content of an element is expressed. (As the `xml:` prefix suggests, this attribute is common to all XML documents: it is not defined by the TEI but by the W3C, as part of the definition of XML.) Its value applies hierarchically to all the elements contained by that element, unless overridden:

```
<div xml:lang="la">
 <s>Pars haec Latine composita est.</s>
 <s xml:lang="en">Except that this sentence is in English.</s>
 <s>Vita brevis, ars longa.</s>
</div>
```

54 Here we specify that the whole `<div>` element uses the language with the coded identifier "la" (The codes used are taken from ISO standard 639 for language identifiers) i.e. Latin. Since it is contained by that `<div>` there is no need to supply this information again for the first `<s>` element. The second `<s>` element however overrides this value, and indicates that its content is in English (the language with identifier `en`). The third `<s>` element is again in Latin. To find the language for any element's content, a processor must first look to see whether that element, or any of its parents, supplies a value for `@xml:lang`. As you may suppose, the nearest parent is the one applicable, if there is more than one. Suppose, for example, that the second sentence above also contained a phrase in French. Such a phrase will need to be tagged with some element of course: the element `<foreign>` is provided for cases like this, where we simply want to show that some sequence of content uses a different language:

```
<div xml:lang="la">
 <s>Pars haec Latine composita est.</s>
 <s xml:lang="en">Except that <foreign xml:lang="fr">sacrebleu</foreign> this
sentence is in English.</s>
 <s>Vita brevis, ars longa.</s>
</div>
```

55 The codes used to identify language may (as here) be standard two-letter codes; more complex codes can be used to distinguish such things as geographically or socially-defined variants, or to distinguish the same language written using different scripts. These codes, and ways of using them, are defined by the ISO rather than the TEI, and we do not discuss them further.

4.2.2 Nonstandard characters

56 We noted above that a TEI document is an XML document, and therefore expressed using Unicode characters. The Unicode standard at the time of writing covers a very high proportion of existing and ancient writing systems, but quite properly does not attempt to standardize writing systems still in the course of being studied or defined, nor does it attempt to provide codes for every possible character or glyph variant likely to be encountered in the study of ancient documents. For these reasons, the TEI provides an element `<g>` which can be used to mark the presence in a document of some character or glyph for which there is no existing standard Unicode codepoint.

57 Suppose for example that we are studying a range of manuscripts characterized by a particularly striking variant form of some letter (say an R); our TEI transcriptions can distinguish this variant glyph from the usual form of the same letter by using a `<g>` element to mark the former. The content of the `<g>` might simply be a normalised form, or the element might be empty; in either case it will carry a `@ref` attribute which points to a `<glyph>` element somewhere, in which we define the nature, function, appearance, name etc. of this variation. Thus, supposing that we have identified and described three variant forms of the letter R, giving them codes R1, R2, R3, we might represent individual occurrences of these forms as `<g ref='#R1'/>`, `<g ref='#R2'/>` or `<g ref='#R3'/>`. The associated definitions may be provided within the`<charsDecl>` element of the TEI Header, as further discussed in #HDR below.

4.3 Rendition

58 The term *rendition* is used in a TEI context to describe the way in which an element is formatted or presented, on screen or on the page. The TEI is not a document formatting system, and (as we suggested above) generally aims to facilitate an encoding which is faithful to the perceived meaning of a text rather than one which reproduces its appearance, or intended appearance. This is one of many aspects which distinguish the TEI from systems such as those typically used for word processing, as we noted above. Nevertheless, recognizing that it may be important to retain information about the original rendition of a document for analytic purposes (not least because it may be that its appearance is all we can reliably describe in an ancient document), the TEI provides a number of mechanisms for encoding such data. We emphasize again that the function of such renditional markup is always to describe the appearance or rendition of one or more original source documents; of course, it is likely that when displaying a text encoded in such a manner we will wish to achieve a similar appearance using different technology, but this is by no means always or necessarily the case.

59 The simplest method is to use the global attribute `@rend`. This attribute behaves in a way similar to the `@xml:lang` attribute discussed above: it supplies a coded value for the rendition associated with an element and (unless overridden) its children. Unlike `@xml:lang` however, the values available for use with the `@rend` attribute are not formally defined, either by the TEI or any other agency, although a comparatively small number of codes are conventionally used. The encoder is free to make up their own set of labels and to apply them as consistently (or inconsistently) as they wish.

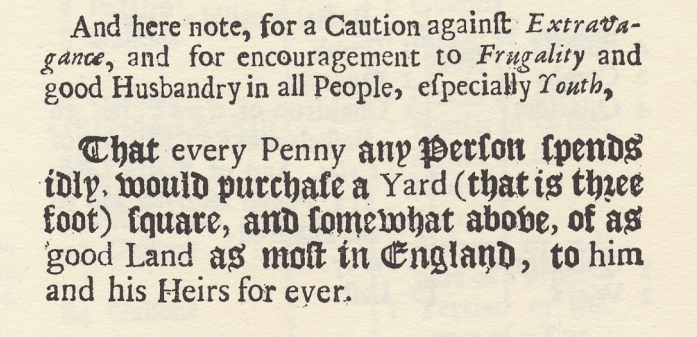
60 The `@rendition` attribute, by contrast, behaves more like the `@ref` attribute on `<g>` in that it references a definition (or set of definitions) provided elsewhere; the encoder is thus constrained both to use only a specified set of renditional labels and also to supply (or refer to) some kind of definition of the intended meanings for each such label.

61 Such "style definitions" are of course most easily expressed using an encoding language designed for formatting software, such as a web browser or document processing system. The TEI allows you to express your definitions in whatever such language you choose; currently a well established W3C recommendation called CSS (Cascading Stylesheets) is the most practical, since it is both

powerful and very widely implemented. The syntax of this language is very simple and it can even be used directly within a TEI document, either to provide a default rendition for an element, or to override that default using yet another global attribute: `@style`.

62 We show how each of these methods might be used to encode the following brief passage, taken from an early 17th century schoolbook.

Figure 3: Extract from Henry Care's The tutor to true English (1687), p 70


And here note, for a Caution againſt *Extravagance*, and for encouragement to *Frugality* and good Husbandry in all People, eſpecially *Youth*,

That every Penny any Perſon ſpends idly, would purchaſe a Yard (that is three foot) ſquare, and ſomewhat above, of as good Land as moſt in England, to him and his Heirs for ever.

63 In this passage, we observe sequences of words in a gothic or black letter font, in a normal roman font, and in italics. We can speculate about the reasons for this typographic variation, but supposing that at least initially all we want to do is record it, we might adopt a encoding like the following:

```
  <p rend="roman"><lb/>And here note, for a Caution against <hi
rend="italic">Extrava-<lb break="no"/>gance</hi>, and for encouragement to <hi
rend="italic">Frugality</hi> and <lb/>good Husbandry in all
   People, especially <hi rend="italic">Youth</hi>,</p>
  <p rend="gothic"><lb/>That <hi rend="roman">every Penny</hi> any Person spends
<lb/>idly, would purchase a
   <hi rend="roman">Yard</hi> (that is three <lb/>foot) square, and somewhat
above, of as <hi rend="roman">
    <lb/>good Land</hi> as most in England, to <hi rend="roman">him <lb/>and his
Heirs for ever.</hi></p>
```

64 This encoding uses the milestone `<lb>` element to mark explicitly the start of typographic lines. It also uses a new element `<hi>` (short for “highlighted”). This element has no particular semantics — it simply indicates that its content is visually distinct from what surrounds it, in rather the same

way that <foreign> simply shows that its contents are linguistically distinct from what surrounds it. The nature of this visual distinction is not fully specified since this encoding does not explain what the values "roman", "italic" or "gothic" mean; nor is there any way of providing such an explanation other than a documentary note. Of course, a human reader will know more or less what "italic" means, but a rendering program will probably not. Suppose instead that we have provided definitions (expressed in CSS) in the place in the <teiHeader> allocated for this purpose:

```
  <rendition xml:id="it" scheme="css">font-family: roman; font-style:
italic</rendition>
  <rendition xml:id="ro" scheme="css">font-family: roman; font-style:
normal</rendition>
  <rendition xml:id="go" scheme="css">font-family: unifraktur cursive</rendition>
```

65 With these definitions in place, we could now simplify our example slightly:

```
  <p rendition="#go">That <hi rendition="#ro">every Penny</hi> any Person spends
idly ...</p>
```

66 Any rendering program can now process the CSS directly and display our text in a visually distinctive manner.

67 Moreover, since the CSS expressions are actually quite simple, we might just use the @style attribute to provide them directly within our text:

```
  <p style="font-family: roman">And here note, for a Caution against <hi
style="font-style:italic">Extravagance</hi>, and for encouragement to <hi
style="font-style:italic">Frugality</hi> ...</p>
```

68 As we noted above, the <hi> element has no semantics other than to indicate that some part of the document being encoded looks significantly different in some respect. In large scale encoding projects identifying such variation may be all that is feasible; in projects working with very ancient, visually complex, or scarcely comprehensible resources, it may also be very hard to go beyond the point of saying "this part of the text is visually distinct in some way." But for the vast majority of TEI encoding projects it is usually considered both desirable and feasible to add value to an encoding by pausing to consider *why* something is visually distinct in the source. Where an answer

to that question can be reached with any degree of confidence, an encoding which shows us what motivated the original renditional decision is more likely to be generally useful than one that does not.

69 It is not so difficult to list the main reasons why a printer might choose to highlight some words in a text printed according to the conventions elaborated in Western Europe over the last 300 years or so. Words in a foreign language are almost always highlighted, as are words on which the author places some degree of linguistic emphasis. Titles and technical terms are conventionally identified by highlighting when they appear in running prose. Until recently, words indicating the names of persons, places, or abstractions would conventionally be highlighted as well.

70 Note that we use the word *highlighted* here to include any form of visual salience, whether it is a passage in italic embedded in one predominantly set in roman font, or the reverse, or (as above) a passage in roman embedded in one predominantly in gothic. If we extend the meaning of the term to include passages set off by quotation marks of various kinds, the list of motivations for highlighting expands to include passages of direct speech, material cited or quoted from elsewhere, or which the writer wishes to indicate as being in some sense non-authorial, words which are being talked about rather than used... and so on and so forth.

71 The TEI provides elements which enable the encoder to make all these and other semantic distinctions within the simple words of a text, guided by the visual clues implicit in the original rendition of a document, and the encoder's understanding of it. This is often a useful way of distinguishing an otherwise ambiguous passage of text. In a modern journal article, for example, a passage set in italic might be a foreign expression, the title of another article or of a film or song, a technical term, or a phrase the author wishes to emphasize. By using the appropriate tag (`<foreign>`, `<title>`, `<term>`, or `<emph>` respectively), the encoder enriches the text and facilitates more intelligent processing of it, for example to list the titles of works mentioned in a document, excluding emphasized or foreign words.

72 These tags can also be used even when no highlighting has been applied, of course. The `@rend` attribute is also available on all these elements, so that the encoder can say both what they believe a highlighted passage to indicate, and in what respects (if any) it is visually salient.

4.4 Names and dates

73 We mentioned above the convention of highlighting the names of persons, places, etc. in printed text. This practice, which was commonplace till the mid 19th century, testifies to a long standing desire on the part of readers to distinguish names clearly from other words, and it is one which remains true of modern day digital encoders. Identifying what are now often referred to as "named entities" from the rest of a sea of words is just as important a task for today's computational linguist building a language understanding system, or today's digital historian tracing social networks within a corpus of 18th century correspondence, as it is for today's intelligence operative trying to track down references to terrorist suspects in vast bodies of surveillance data.

74 The TEI provides several elements to encode such things once they have been identified. Probably the most general-purpose of these is `<rs>` (short for "referring string") which can be used to tag any sequence of words considered by the encoder to refer to some entity such as a person or place, even if the reference does not take the form of a name, but is rather a pronominal reference ("she," "the king") or deictic phrase ("that girl," "the chap we saw on Thursday"), or idiom ("the great wen," "the big apple"). When however a reference of this kind contains only one or more proper nouns, it should be encoded using the generic `<name>` element.

75 Both `<rs>` and `<name>` elements have a `@type` attribute which can be used to distinguish the type of entity being named, so that, for example, the place `<name type="place">London</name>` is distinguishable from the author `<name type="person" >London</name>`. However, the TEI also provides specialised tags which enable you to make the same distinction, and others, with more precision. These more specialised elements distinguish names which refer to different kinds of entity explicitly: thus it provides `<persName>`, `<placeName>`, `<orgName>`, to support the assertion that the name refers to a person, place or organization. It is also possible to identify significant subcomponents of such names: the TEI provides `<surname>`, `<foreName>`, `<altName>` (for an alternative name such as a nickname), `<roleName>` (for a socially-defined role or title such as "Lady," "Esquire," "Preacher," etc.), `<genName>` (for a generational name such as "Junior," "the elder," etc.) and others. For place names, the TEI distinguishes components placing the name socio-politically such as `<settlement>`, `<region>`, `<country>` and also components such as `<geogName>`

for any purely geographic name such as “Mount Sinai” or “River Thames,” which can also be further decomposed into a name (“Thames,” “Sinai”) and the name of some kind of geographic feature (“Mount,” “River”).

76 These elements are particularly useful when dealing with archival documents of any kind, in which indexing the names of people and places referred to in a document is a major concern, but they may be useful in any kind of digital edition or lexical study.

77 Suppose, for example, that we are preparing a digital edition of a primary document such as the Current Record of Events from 1792 to 1885, As recorded by The Shaker community of Canterbury, New Hampshire (a manuscript preserved in the Special Collections of Hamilton College Library). This 19th c. document begins with the following sentence: “On the first of February 1792 Father Job Bishop and Elder Edmond Lougee came from New Lebanon N.Y. to organize and establish a community of Believers at Canterbury N.H..”

78 We might be content simply to record the presence of the proper names (“Father Job Bishop,” “Canterbury” etc.) using the <name> element as follows:

```
  <p>On the first of February 1792 <name>Father Job Bishop</name> and <name>Elder
Edmond Lougee</name> came
   from <name>New Lebanon, N.Y.</name> to organize and establish a Community of
Believers at
   <name>Canterbury, N.H.</name></p>
```

79 Even this minimal tagging makes it easy to distinguish the names of the people and places mentioned, and hence to process them separately from the rest of the text, for example to construct an index of names, or to link these references to further information about the person or place concerned.

80 A more careful tagging would however distinguish the component parts of the names which (as is often the case) combine family names (“Bishop,” “Lougee”) with names indicating a social role (“Father,” “Elder”) and with forenames (“Job,” “Edmond”). Such distinctions can only be made explicit by proper tagging — “Bishop” might be a role or a surname, depending on context. Similarly, within a <placeName> it is convenient to distinguish the name of an administrative

or other region such as a US state from the name of a *settlement* (TEI uses this neutral term in preference to other terms such as "town," "village," "city" etc. which are hard to define outside a very precise context).

```
  <p>On the first of February 1792 <persName><roleName>Father</roleName>
    <forename>Job</forename>
    <surname>Bishop</surname></persName> and <persName><roleName>Elder</roleName>
    <forename>Edmond</forename>
    <surname>Lougee</surname></persName> came from <placeName><settlement>New
Lebanon</settlement>
    <region>N.Y.</region></placeName> to organize and establish a Community of
Believers at
   <placeName><settlement>Canterbury</settlement>,
<region>N.H.</region></placeName></p>
```

81 By definition, a name refers to a specific entity (a person or place) but historical or genealogical research would not be much of a challenge if it did so unambiguously or directly. Is this "Edmond Lougee" the same person as the "Edmund Lougee" recorded in other sources? A little further in the source, there is a mention of "Elder Lougee": does this refer to the same person? As ever, TEI cannot help you answer such questions, but it does provide ways of expressing the answer (if any) you have reached.

82 Proposing a standardised form for a name has, of course, long been the business of librarians and cataloguers. For literary works and persons, and for modern or classical geographies, there exist well-established reference works or "authority files" which provide a useful point of reference for many of the entities that may be named in a TEI document; for others, we may find some kind of canonical information in other online references such as wikipedia or ancestry.com. One way of making clear which of the many possible people which we believe this "Edmond Lougee" concerns would therefore be to provide a reference to an online definitive record for the person :

```
... <persName ref="http://records.ancestry.co.uk/Edmund_Lougee_records.ashx?
pid=24762763"> Elder Edmond Lougee
  </persName> ...
```

83 However, given that few of the people in an archival document of this kind are likely to have such records, it will often be necessary for a project to define their own authority file, comprising a series of records, one for each identified individual. The TEI provides additional elements which can be used to record such information in some considerable detail: biographical or prosopographical information for example can be grouped within a `<person>` element, while geographical information can be grouped within a `<place>` element; if this is done, then the `@ref` attribute can be used to point to such a description:

```
<persName ref="#P1234"> Elder Edmond Lougee </persName>

<!-- .... -->
<person xml:id="P1234">
 <p>Edmund or Edmond Lougee was born in Exeter Newmarket, Rockingham, New
Hampshire, USA on 1731 to John
  Lougee and Anne Gilman. He married Hannah Lord and had 7 children. He passed
away on 3 Jun 1807 in
  Loudon, New Hampshire, USA.</p>
</person>
```

84 Note that this example simply provides a prose summary of the available information for Elder Lougee within the referenced `<person>` element; it might alternatively have been constructed using specialised elements such as `<birth>`, `<death>`, `<marriage>`, `<relation>` etc.

85 It may however be sufficient for the needs of a project to provide with each reference an arbitrary code, or a normalised form of the name, perhaps including dates or a generational marker to make that form unambiguous. The `@key` attribute is provided for this purpose:

```
<persName key="Lougee, Edmond (1731-1807)"> Elder Edmond Lougee </persName>
```

86 Normalisation by means of attribute values is a mechanism found throughout the TEI. It is particularly useful wherever something that a search engine or other processor would prefer to find in one particular format actually appears in a document in many different forms, some of which may have some linguistic interest or significance. If we want to recover from this document

all references to events occurring before or after a particular date, for example, the task will be much easier if the dates all use a standard representation. In the above example, we might therefore choose to encode the first date as follows:

```
<p>On <date when="1792-02-01">the first of February 1792</date> ... </p>
```

87 using the W3C recommendation for the format of dates. The `@when` attribute used here allows us to associate a date with a specific point in time. The TEI also provides a set of other attributes which can be used in combination to express a variety of temporal concepts:

```
<p><date notAfter="1792-02-28" notBefore="1792-02-01">In February 1792</date>
... </p>
```

```
<p><date from="1792-02-01" to="1792-02-07">During the first week of February
1792</date> ... </p>
```

4.5 Tables, figures, and bibliographies

88 Documents in the real world contain many non-textual or only semi-textual components such as figures or tables, as well as textual components that have their own internal structure such as indexes and bibliographies. Particularly where the TEI is used to encode a new document, but also where there is a need to indicate the presence of such features in an old one, the encoder cannot do without elements such as `<table>`, `<figure>`, or `<bibl>`, each of which we introduce very briefly in the following sections.

4.5.1 Tables

89 A table is a way of organizing several related snippets of textual information into rows (or columns) composed of cells. Most document production systems have developed quite complex ways of indicating exactly how the cells of a table should be displayed, but these are largely missing from the TEI table model. In the TEI, a `<table>` element is composed of `<row>`s, which are composed of `<cell>`s; a TEI table cannot therefore easily be represented as a series of columns. There is an attribute `@role` to indicate whether a particular row or cell contains labelling information or data, and there are attributes `@rows` and `@cols` which can be used to indicate when a row or a cell spans more than one row or column respectively.

90 Suppose we find the following (imaginary) table in a document we are encoding:

Figure 4: An imaginary table

Fruit	apple	banana	cherry		date
Nuts	almond	brazil	coconut	doughnut	pistachio

In our TEI version, we can record that the first column of cells supplies a label for the entries in the remainder of each row, and also that the cell in the third column of the first row spans two columns.

```
<table>
 <row>
  <cell role="label">Fruit</cell>
  <cell>apple</cell>
  <cell>banana</cell>
  <cell cols="2">cherry</cell>
  <cell>date</cell>
 </row>
 <row>
  <cell role="label">Nuts</cell>
  <cell>almond</cell>
  <cell>brazil</cell>
  <cell>coconut</cell>
  <cell>doughnut</cell>
  <cell>pistachio</cell>
 </row>
</table>
```

4.5.2 Figures

91 Any kind of graphic component such as an illustration, chart, or diagram may be embedded within a document, sometimes as a separate component such as a frontispiece or separately paginated illustration, sometimes as a part of some division of the text. Such figures often contain headings

or titles, possibly associated with some running text, as well as an image. To encode them, the TEI provides a `<figure>` element which typically contains at least `<graphic>` and one or more `<head>` elements, though other textual elements may be used as well if the illustration contains text.

92 The `<graphic>` element is a specialised kind of pointer. Its `@url` attribute points to a location where a digital representation of the image concerned may be found. Attributes `@scale`, `@width` or `@height` are available to specify the desired size of the image when it is displayed.

93 Consider the following example taken from Punch, a well known 19th century British humorous publication:

Figure 5: John Leech: Domestic Bliss,cartoon in Punch vol 13 (July 1847), page 14.

94 We may encode this as follows:

```
<figure type="cartoon" place="topLeft">
 <head rend="caps">Domestic Bliss</head>
 <graphic url="vol13p14.png"/>
 <sp>
  <speaker rend="italic">Wife of your bussum</speaker>
  <p rend="smallcaps"><q>Oh! I don't want to interrupt you dear. I only want
some money for Baby's
   socks – and to know whether you will have the mutton cold or hashed.</q></p>
 </sp>
 <figDesc>A domestic interior drawn by Leech, showing a wild-haired man in a
dressing gown sitting at a
 desk covered in papers, with his wife, two small children, one of them banging
a drum, and a cat.
 </figDesc>
</figure>
```

95 In our encoding we have transcribed the text below the graphic using the <sp> element discussed earlier, to show that the text is presented as a piece of dramatic dialogue. We have also used the <figDesc> element to enclose additional text, not present in the figure, but supplying useful descriptive metadata which might be displayed as an alternative to the graphic itself, or used to index its content.

4.5.3 Bibliographic descriptions

96 Bibliographic descriptions, such as would usually be included as a bibliography or list of references at the end of an academic article, or in a footnote, are a common feature of scientific writing. It is useful to distinguish explicitly such items and in particular their components (author, title, publisher, publication date, etc.) both to make them more accurately searchable and to make it easier to render or format them in different ways. Where large numbers of such references are to be handled, specialist tools such as Zotero which can import or export in a TEI format provide an efficient way of maintaining them, but they can be treated in just the same way as any other component of a TEI document whether they appear in the body of the text, as a separate division of the back matter, or in the header.

97 It is conventional practice in scientific writing to provide references in a standard format, in which the components appear in a specified order, and use consistent layout. Whether in a footnote, or in a standalone list of references, a book will typically be described something like this:"Cameron, D. (1995) Verbal Hygiene London and New York: Routledge. "TEI markup can be used to show which part of this is the title, which the place of publication, etc. The TEI provides two distinct elements for this purpose: `<bibl>` and `<bibl>`.

98 The `<bibl>` element provides a "structured" or *data-centred* view of a bibliographic item, in which each part of the description (author, title, etc.) is tagged distinctly, treating it as if it were part of a database. Note that punctuation is not allowed between components: this is because different visualisations might be appropriate for different bibliographic styles. A detailed tagging like the following allows this to be done simply: the formatting program simply chooses the elements needed in the order required, inserting punctuation according to the style required.

```
<bibl xml:id="Cameron1995">
 <monogr>
  <author><surname>Cameron</surname><forename>Deborah</forename></author>
  <title>Verbal Hygiene</title>
  <imprint>
   <publisher>Routledge</publisher>
   <pubPlace>London</pubPlace>
   <pubPlace>New York</pubPlace>
   <date>1995</date>
  </imprint>
 </monogr>
</bibl>
```

99 The `<monogr>` element here indicates that this is what bibliographers call a *monographic* item, that is an individual work, rather than an *analytic* item such as an article in a journal.

100 The `<bibl>` element provides a less structured, more *text-centred*, view of a bibliographic item, in which punctuation is permitted between tagged items, along with any running prose found, and the order of the components is not fixed. It is useful where the encoder wishes to follow the conventions of the (usually printed) catalogue from which such a description is taken or for which it is intended.

101 Using this, we might tag this item in a way that more closely resembles the original version, at the price of making its subsequent processing a little more difficult:

```
  <bibl><author>Deborah Cameron</author>: <title level="m">Verbal
Hygiene.</title>. London, New York:
   Routledge.<date>1995</date></bibl>
```

102 Most of the elements available within `<bibl>` are also available (but optional) within `<bibl>`, so it is possible to use either element to provide detailed tagging of any kind of bibliographic citation. We give some further examples below.

103 When a book is mentioned within running text, particularly if it is mentioned more than once, it is usual to do so by means of a very short reference such as *Cameron 1995*. To link such a reference to a fuller bibliographic description of the work in question we can use the `@target` attribute on a `<ref>` or `<ptr>` element:

```
  <p>Humans have <q>a healthy obsession with language</q> (<ref
target="#Cameron1995">Cameron 1995</ref>).
  It would be surprising if we did not...</p>
```

5. The TEI cornucopia, part two

5.1 Editorial matters

104 When digitization of primary source materials is undertaken, the first priority is usually to produce good quality digital images of the original source documents. Transcribing and encoding those images is an expensive and difficult operation which cannot readily be automated, if at all, but it is an important next step in the creation of a true "digital edition." For the skilled human reader, the digital image of its pages may be all that is needed to make the source useful. For a wider audience however, and certainly for any kind of automated analysis or searching, an encoded transcription will be an essential complement to the image. The TEI provides a number of mechanisms to facilitate the production of digital editions of every kind.

5.1.1 Image and transcription

105 We mentioned above the <facsimile> element which can be used to group and organize digital images in terms of the physical surfaces they represent (using the <surface> element) and the logical zones of interest within those surfaces (using the <zone> element). A <facsimile> may be all that a TEI document contains, or this may be complemented by a transcription.

106 Where both facsimile and transcription are available, a global attribute @facs may be used to link them, or parts of them, as appropriate. Suppose for example that we have a single leaf, with images of its two surfaces in two files called page1r.png and page1v.png respectively. We transcribe the two surfaces within a <text> element as usual, using <pb> to mark the points in the transcription that correspond with the start of each new page in the source. The resulting document might look like this:

```
<facsimile>
 <surface xml:id="s1">
  <graphic url="page1r.png"/>
 </surface>
 <surface xml:id="s2">
  <graphic url="page1v.png"/>
 </surface>
</facsimile>
<text>
 <body>
  <pb facs="#s1"/>

<!-- text transcribed from page 1 recto here -->
  <pb facs="#s2"/>

<!-- text transcribed from page 1 verso here -->
 </body>
</text>
```

107 Note that associating the transcription with the surface rather than with the image file directly offers us the flexibility of adding further image files (of higher or lower resolution, for example) without perturbing the structure.

108 The @facs attribute used above is global, and can therefore be used to associate any part of a transcription with any part of a <facsimile>. We could for example link a few words in the transcription with that part of the image where they appear.

5.1.2 Editing a transcription

109 When transcribing a source, particularly an ancient one, there is always a tension between the desire to represent faithfully the actual state of the document, and the desire to make that document comprehensible to a modern or untrained reader. The process of transcription necessarily involves a kind of normalization, in which the different written symbols (the glyphs) of a manuscript or an early print document are mapped on to the unambiguous codes of a modern character set. An encoding which tries to represent the words of a document produced according to norms different from our own, or to correct "self evident" errors in the copy text is thus almost inevitable. The TEI provides ways of marking stretches of text as having been normalized or corrected, by the encoder, whether or not we consider this to be "editing" in the traditional sense.

110 We mentioned above that the TEI consciously places itself within the long western philological tradition concerned with the identification and recovery of text from scattered documentary witnesses. It is scarcely surprising therefore to find that it also provides facilities for the representation of such phenomena as scribal correction or alteration where these are manifest in a particular document, or for the representation of a collation of variant readings. Where pre-digital textual editors were necessarily concerned with the production of a single carefully considered and substantiated print version of a text, the economics of digital production have greatly facilitated the production of "diplomatic" digital or documentary editions, which claim to present objectively the facts of a given collection of documents in such a way that the reader may determine the plausibility of any accompanying "edited" version, or even derive such an edition for themselves.

111 When transcribing a source text, the <corr> element can be used to enclose a part that has been corrected and the <reg> element a part that has been normalised. For the opposite case, the <sic> element may be used where something should have been corrected but has not; and the <orig> element for cases where the original spelling has been retained even though it looks unfamiliar. For example, a modernized version of the Shakespeare sonnet shown above would probably begin

```
<l>My Mistress' eyes are nothing like the sun,</l>
<l>Coral is far more red than her lips red,</l>
<l>If snow be white, why then her breasts are dun:</l>
<l>If hairs be wires, black wires grow on her head:</l>
```

112 Even in such a modernized version, a conscientious encoder wishing to show where they have departed from the original source may simply wrap the words “Mistress',” “sun” etc. with the <reg> element. More usually however an encoder will wish to retain the original orthography along with the modernized version, so that either is available for display. The <choice> element addresses this requirement:

```
<l>My <choice><reg>Mistress'</reg><orig>Mistres</orig></choice> eyes are nothing
like the
 <choice><reg>sun</reg><orig>Sunne</orig></choice>,</l>
<l><choice><reg>Coral</reg><orig>Curral</orig></choice> is far more red
 <choice><reg>than</reg><orig>then</orig></choice> her lips red,</l>
```

113 The <choice> element may be used wherever several different possible encodings are possible but only one of them may be chosen for a particular application. It is less appropriate in the case where there are many different possible witnesses, providing multiple readings all of which need to be taken into consideration. The TEI provides a more powerful and complex set of tags to mark up a traditional critical apparatus of this kind. The element <app> (apparatus) is used to mark the point of variation being considered; within this element, the encoder can supply a series of <rdg> (reading) elements, one for each variant reading identified. If editorial policy permits, one of these readings may be considered as the “primary” or “correct” one, in which case it will be tagged using the <lem> element rather than as just another <rdg>. The <rdg> element has a @wit attribute which can be used to identify the witnesses in which the reading is question is attested. For example, the first line of Chaucer's Wife of Bath's Tale begins “Experience though noon Auctoritee...,” but its first word is spelled quite differently in three of the surviving manuscripts. Taken as a whole, such spelling variations provide useful evidence for the reconstruction of the history of the manuscript tradition in question, and our encoding therefore wishes to preserve them. We might do so as follows:

```
<l n="1">
 <app>
  <rdg wit="#El #Hg">Experience</rdg>
  <rdg wit="#La">Experiment</rdg>
  <rdg wit="#Ra2">Eryment</rdg>
 </app> though noon Auctoritee... </l>
```

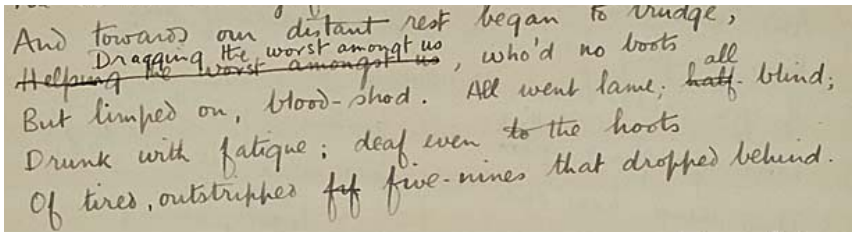
114 Here the values of the `@wit` attribute identify the manuscripts in question: as the hash-sign indicates, they are actually links to items in a list of the manuscripts which were collated to produce this apparatus.

5.1.3 Encoding the writing process

115 Our last example of the facilities offered by the TEI for working on the detailed transcription of primary source materials introduces a few of the elements needed to record the actual writing process, for example to record passages which have been deleted, added, corrected etc. whether by the author of a literary text or by a scribe copying out a manuscript. An analysis of such documentary modifications may be essential before a reading text can be presented, and is clearly of importance in the editorial process.

116 The example is taken from the surviving authorial manuscript of a poem by the English writer Wilfred Owen, a part of which is shown here:

Figure 6: Detail from Dulce et decorum est autograph manuscript in the English Faculty Library, Oxford University.



117 Owen first wrote “Helping the worst amongst us,” but then deleted it, adding “Dragging the worst amongt us” over the top. In the same way, he revised the phrase “half-blind” by deleting the “half-” and adding “all” above it. In the last line, he started a word beginning “fif” before deleting it and writing the word “five-nines.” We can encode all of this as follows:

```
<l>And towards our distant rest began to trudge,</l>
<l><subst><del>Helping the worst amongst us</del>
  <add>Dragging the worst amongt us</add>
 </subst>, who'd no boots </l>
<l>But limped on, blood-shod. All went lame; <subst>
  <del status="shortEnd">half-</del>
  <add>all</add></subst> blind;</l>
<l>Drunk with fatigue ; deaf even to the hoots</l>
<l>Of tired, outstripped <del>fif</del> five-nines that dropped behind.</l>
```

118 The tags <add> and <del> elements are used to enclose passages added or deleted respectively. Additional attributes are available such as @resp to indicate responsibility for the modification, or @place to indicate where in the text (for example, above or below the line) the modification has been made. Where the encoder wishes to assert that the addition and deletion make up a single editorial act of substitution, these elements can be combined within a <subst> element as shown above.

119 A very careful examination of Owen's second modification shows that he really did write “amongt” rather than “amongst,” presumably in error. An equally careful editor wishing to restore the missing “s” might use the <supplied> element to indicate that they have done so:

```
<add>Dragging the worst among<supplied resp="#ED">s</supplied>t us</add>
```

120 Here the @resp attribute has been used to indicate that the "S" was not supplied by Owen but by someone else, specifically the person documented elsewhere by an element with the identifier "ED".

5.2 Transcribed speech

121 The digital medium stores and represents sound or video just as easily and effectively as it does images. Transcriptions of sound or video recordings may be used to complement them in the same way as they are used to complement page images. Whether automatically or manually produced, transcriptions of speech are of particular interest in many branches of linguistics; for example,

in language acquisition studies, in studies of language variation, or of the relationship between speech and writing in both literate and pre-literate societies. The study of transcribed speech is also an essential component of sociological studies such as oral history.

122 However, the units in which such analysis is carried out — in TEI terms, the component elements of a <text> which contains transcribed speech — are not universally agreed on, and may even be controversial. Another complicating factor is that the process of transcribing speech is technically difficult and often carried out with the aid of special-purpose software; this is a complication because most such software systems have a preference for using their own internal formats rather than an externally defined standard.

123 Nevertheless, the TEI does propose a set of elements which may be useful for the detailed encoding of speech transcriptions, and which could be used as a kind of interlingua for existing alternative encoding methods.

124 The TEI model is concerned with speech as a communicative system: it therefore proposes the explicit labelling of any part of a recorded sound which has a communicative function or effect. Events such as a bell ringing or an airplane flying overhead are therefore explicitly noted using the <incident> element. Spoken communication is effected using the voice, but not all voice effects are regarded as lexical, and not all communication is vocal: the TEI therefore provides an element <vocal> for non-lexical phenomena such as coughs or laughter as well as an element <kinesic> for non-lexical and non-vocalic phenomena such as gestures or facial expression. These elements are complemented with a small number of others concerned with what might be called the "rendition" of speech : notably <pause> and <shift>; the latter being used to indicate changes in voice quality (for example in loudness, pitch, or speed). These function in a way very similar to the milestone elements introduced earlier.

125 One of the more controversial notions in speech research is that of "turn" or "utterance": the TEI proposes an element <u> which can be used to delimit a stretch of speech from a single speaker, without making any claim as to the status or discourse function of such a stretch. The element carries a @who attribute which can be used to link it to another element providing information about the person speaking, in the same way as we have already seen for the @ref attribute.

126 For many kinds of linguistic analysis, the <s> element mentioned above may be a more useful way of organizing a transcription since it permits the encoder to categorize each segment independently of the utterance containing it, particularly since utterances may be much longer (or shorter) than the usual units of analysis.

127 Here is a simplified example taken from the British National Corpus (BNC) in which both <s> and <u> have been used to segment the transcribed speech signal:

```
  <u who="#PS6P0">
   <s n="3">There ain't</s>
  </u>
  <u who="#PS6NY">
   <s n="4"><shift new="laughing"/>There is<shift/></s>
  </u>
  <u who="#PS6P0">
   <s n="5">Oh shut up</s>
  </u>
  <u who="#PS6NY">
   <s n="6">Ach you do get corners on boats</s>
  </u>
  <u who="#PS6P0">
   <s n="7">No</s>
   <s n="8">Boats are shaped like a bloody rugby ball shape type</s>
  </u>
  <u who="#PS6NY">
   <s n="9">No they ain't</s>
   <s n="10">One end is and the other one ain't <pause/> and it was a yacht
<pause/> and a yacht they got
    little rooms in or something cos <unclear/> innit</s>
  </u>
  <u who="#PS6P0">
   <s n="11">Oh that<pause/> th the rooms are shaped ni <pause/> like to the size
of the boat<pause/> you
    nonce <vocal desc="tut"/></s>
  </u>
  <u who="#PS6NY">
   <s n="12">Let's ask your Mum if there's <pause/> if there's any corners on a
boat</s>
  </u>
  <u who="#PS6P0">
   <s n="13">Of course there ain't</s>
  </u>
  <u who="#PS6NY">
   <s n="14">Just ask your Mum that</s>
  </u>
  <u who="#PS6P0">
```

```
 <s n="15">Yeah</s>
</u>
<u who="#PS6NY">
 <unclear/>
</u>
<u who="#PS6P0">
 <s n="16">I bet she'll probably side with you</s>
</u>
```

128 When transcribing speech, the transcriber must also make decisions about how to represent the non-standard or only semi-lexicalised sounds that speakers make, such as “hmmm,” or the “ach” or “tsk tsk” in the above example (in this case the decision has been to treat the former as a word and the latter as a <vocal>). Even when it is clear that the sound is lexical, it is not always clear how it should be spelled: for example, the sound represented above as “yeah” might equally well be spelled “yeh” or “yuh” or even “yes.”

129 In language corpora of this kind it is also common practice to markup each word explicitly, often using the <w> element. Deciding how this tokenisation should be done is not always self-evident: in the BNC, for example, elided forms like “ain't” and “there's” are tagged as two <w> elements and additional tags are used to distinguish false starts or truncated forms like “th” and “ni” in the above example. The main use of such tagging is however to support additional information such as “part of speech” or “morphosyntactic” analyses, as described in the section on #ANA below.

130 There is one further important feature of the TEI model for transcribing speech: alignment. Whereas in an image the transcription has to be aligned within a two-dimensional co-ordinate space, within a time-based medium such as audio, the various parts of a transcription must be located within time. Speakers typically interrupt one another, or speak at the same time; understanding a dialogue is impossible without information about the sequence of events transcribed. The TEI proposes a way of explicitly recording a kind of temporal scale, using the element <timeline>, which contains definitions of points in time, represented as <when> elements. Events in a transcription taking place at a given point in time can then be aligned with the appropriate <when> elements using attributes such as @sync (synchronous). The advantage of this method is that it permits a transcription to represent the minimum of information needed to align

its components, without necessitating an accurate timestamp for each of them; such timestamping is however generally provided by an automatic transcription system and it may therefore be simpler to use that.

5.3 Dictionaries

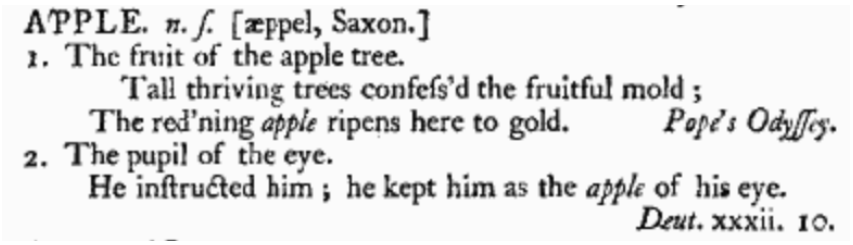
131 The TEI provides elements for the detailed encoding of only a few distinct kinds of document, preferring generally to define elements specific to particular kinds of analysis or approach. However, it does provide a set of tags for the markup of dictionaries and lexica, perhaps because historically these were amongst the first types of major reference work to be demonstrably better managed in digital form than in print.

132 Dictionaries, particularly in print form, are unlike many kinds of document in that they are not composed of running text, but rather of discrete entries, within which individual and semantically significant subcomponents can readily be recognised, even though they may not always be presented as consistently as a database designer might wish. A dictionary entry always has a headword, followed typically by some linguistic information such as its part of speech, and by a series of sense definitions, often arranged hierarchically. Etymological information, examples of usage, references to homonyms, etc. may also be present.

133 In a TEI-style encoding, all of these and other parts of an entry will be distinguished as clearly as possible, whether the purpose is to produce a new dictionary (in which these things must all be rendered in a consistent way), or to represent an existing one (in which distinguishing these things will improve the range of intelligent searching).

134 As a first simple example, consider the following :

Figure 7: "Apple" from Johnson's Dictionary of the English Language(1755)


APPLE. *n. ſ.* [æppel, Saxon.]
1. The fruit of the apple tree.
Tall thriving trees confeſs'd the fruitful mold;
The red'ning *apple* ripens here to gold. *Pope's Odyſſey.*
2. The pupil of the eye.
He inſtructed him; he kept him as the *apple* of his eye.
Deut. xxxii. 10.

135 A minimal TEI encoding for this entry would distinguish the word form used as lemma, its spelling and grammatical coding, the etymology provided, and the two senses :

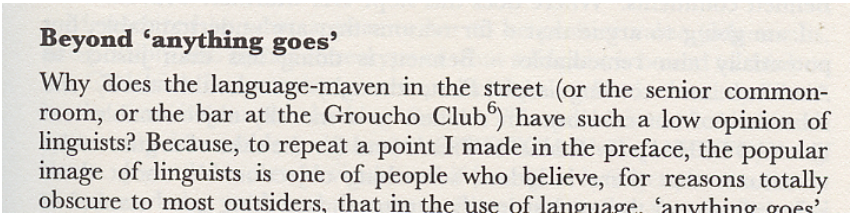
```
  <entry>
   <form type="lemma">
    <orth rend="ALLCAPS">A'PPLE</orth>. <gramGrp><pos
norm="noun">n.s.</pos></gramGrp>
   </form>
   <etym>[<foreign xml:lang="ang">æppel</foreign>, <lang>Saxon</lang>.] </etym>
   <sense n="1">
    <def>The fruit of the apple tree.</def>
    <cit>
     <quote type="verse">
      <l>Tall thriving trees confess'd the fruitful mold;</l>
      <l>The red'ning <hi>apple</hi> ripens here to gold.</l>
     </quote>
     <bibl>Pope's <title>Odyssey</title>.</bibl>
    </cit>
   </sense>
   <sense n="2">
    <def>The pupil of the eye.</def>
    <cit>
     <quote> He instructed him; he kept him as the apple of his eye.</quote>
     <bibl>Deut. xxxii. 10.</bibl>
    </cit>
   </sense>
  </entry>
```

5.4 Notes

136 Texts, particularly old ones, are often accompanied by extensive annotation and commentary, either supplied by the original writer, or more usually by an editor or later commentator. All kinds of academic writing are also typified by footnotes, sidenotes, glosses, bibliographic references, etc. For all of these features, the most natural kind of tagging is to use a `<note>` element to contain the annotation itself, and to embed it within the text being annotated at the "point of attachment," that is, the place where a footnote reference, or other mark typically appears in a printed text. Attributes on the `<note>` element can be used to categorize it in some way, to indicate its approximate placement, and to indicate who is responsible for it.

137 In modern printing practice, explanatory annotations are usually moved off to the back of the book. For example, in a discussion of contemporary attitudes to linguists published in 1995, we find the following comment:

Figure 8: Extract from Deborah Cameron's Verbal Hygiene (p 229)

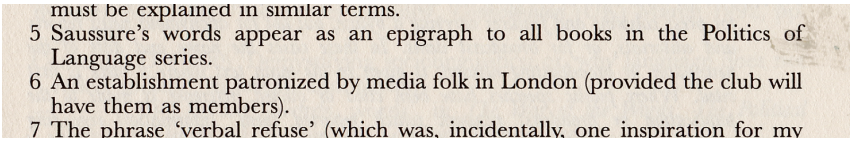


Beyond 'anything goes'

Why does the language-maven in the street (or the senior common-room, or the bar at the Groucho Club[6]) have such a low opinion of linguists? Because, to repeat a point I made in the preface, the popular image of linguists is one of people who believe, for reasons totally obscure to most outsiders, that in the use of language, 'anything goes'

The raised figure 6 here is an indication that the author has provided a further comment, which appears duly collected with others some 18 pages later on:

Figure 9: Extract from Deborah Cameron's Verbal Hygiene (p 247)



must be explained in similar terms.
5 Saussure's words appear as an epigraph to all books in the Politics of Language series.
6 An establishment patronized by media folk in London (provided the club will have them as members).
7 The phrase 'verbal refuse' (which was, incidentally, one inspiration for my

.

138 One simple way of encoding this would be as follows:

```
<div>
 <head>Beyond "anything goes"</head>
 <p> Why does the language-maven in the street (or the senior common-room, or
the bar at the Groucho Club
  <note>An establishment patronized by media folk in London (provided the club
will have them as
  members).</note>) have such a low opinion of linguists? Because...</p>
</div>
```

139 Here the note is encoded inline even though the printed source text has separated the notes out from the text. This makes it easier to present the note in a variety of ways (for example, as a pop up in an online version) and also simplifies its encoding. However, it is also perfectly feasible to treat the notes as a separate section of the document, and to place explicit links in the text at the point of attachment, using the <ptr> element:

```
  <div>
   <head>Beyond "anything goes"</head>
   <p> Why does the language-maven in the street (or the senior common-room, or
the bar at the Groucho Club
    <ptr target="#note6"/>) have such a low opinion of linguists? Because...</p>
  </div>
  <back>
   <head>Notes</head>

  <!-- other notes here -->
   <note xml:id="note6">An establishment patronized by media folk in London
(provided the club will have
   them as members).</note>

  <!-- and here -->
  </back>
```

To represent the link between the text and the note, we first give the <note> element a unique identifier (note6) and we then place a <ptr> element at the point in the text from which we wish to refer to it (the "point of attachment"). This pointer element both represents the supralinear 6 in our original source indicates that its function is to point to the note whose identifier is supplied by its @target attribute. Alternatively, if we care to record exactly the way the point of attachment appears in the text, we could use the <ref> element:

```
...the Groucho Club <ref target="#note6">6</ref>) have
...
```

5.5 Linguistic annotation

140 In the course of preparing of an encoded text, there is a natural tendency for the encoder to wish to add their own annotations. In a sense, of course, every piece of markup added to a text represents the result of an analysis, whether human or automatic, and so it is natural to think of representing such annotations incrementally by means of markup to a digital text. The `<note>` element may be used for this purpose. However, for many people, linguistically motivated analyses (such as "this is a verb," or "this sequence of words has this syntactic function," or even "this is a metaphor") are of a rather different kind from the kinds of annotation ("this is a title," "this is a personal name," etc.) discussed elsewhere in this book. In particular, there is less consensus about the categories involved: most people will readily agree that "place name" is a useful and definable concept, but the concepts behind linguistic analysis are much more various. The TEI does not, therefore, propose tags such as `<noun>` or `<adjunct>` or `<metaphor>`, but instead it proposes general purpose elements for segmenting texts into smaller units, any of which can carry labels defined by the project.

141 This method is particularly useful when a digital text is automatically analysed by a computer program, (for example a morphosyntactic analyser, or part-of-speech tagger) since the format used for the outputs from such automated systems differ considerably. The TEI offers a simple way of explicitly segmenting running text into word-like units, and associating linguistic codes with each such segment. For example, like many other language corpora, the BNC provides codes identifying the part of speech (noun, adjective, verb etc.) of each word as well as its base or uninflected form. We might therefore represent the sentence quoted above in TEI as follows :

```
<u who="PS6NY">
 <s n="12">
  <w ana="#VM0" lemma="lets">Lets</w>
  <w ana="#VVI" lemma="ask">ask </w>
  <w ana="#DPS" lemma="you">your </w>
  <w ana="#NN1" lemma="mum">Mum </w>
  <w ana="#CJS" lemma="if">if </w>
  <w ana="#EX0" lemma="there">there</w>
  <w ana="#VBZ" lemma="be">'s</w>
  <w ana="#DT0" lemma="any">any </w>
  <w ana="#NN2" lemma="corner">corners </w>
  <w ana="#PRP" lemma="on">on </w>
  <w ana="#AT0" lemma="a">a </w>
  <w ana="#NN1" lemma="boat">boat</w>
  <pc>. </pc>
 </s>
</u>
```

142 In this encoding (production of which is largely automated, of course) the `<w>` element has been used to delimit each lexical item. For each such unit, the attribute `@ana` is used to point to a part of speech definition, and the attribute `@lemma` to supply a base form.

143 Linguistic analyses of this kind have been in use for many decades and are essential to most current systems used in Natural Language Processing (NLP). Consequently, both the techniques used and the categories represented are in the process of being standardised, at least for major Western European languages. The TEI approach allows the encoder to choose how much or how little they wish to make explicit the meaning of the analysis pointed to by the `@ana` attribute: that is, how to define what `VBZ` or `NN1` actually means.

144 One method is to provide a further definition for the category, typically within a TEI Header, using the `<taxonomy>` element discussed in section #encDesc. Such a definition will typically be meaningful only to a human being, but it can still be very useful to provide one.

145 At the other extreme, categories may be formally defined using the joint TEI/ISO recommendations for representing linguistic analyses in terms of *feature structures* (see further Chapter 18 of the TEI Guidelines). Such analyses are now used quite widely in the NLP community, for example as a means of mapping information between different computational lexica.

146 Or, intermediate between the two, linguistic categorizations may be made with reference to an existing standard, such as that provided by the ISO Data Category Registry (see further Ide and Romary 2009.). This approach has much in common with many other data harmonization efforts on the Web: it makes it simpler for systems to interoperate, in the same way as TEI documents in general can be interoperable.

147 It is also worth noting that the same mechanisms are available for any kind of analysis, not necessarily a linguistic one; stylistic analyses (which typically characterize stretches of text larger than a single word) can also be encoded in much the same way, by attaching the `@ana` attribute to whatever element is chosen to enclose the stretch of text concerned, for example a `<p>`, `<div>`, `<s>`, etc.

148 Of course the units of text being analysed do not necessarily coincide with the units of text directly represented within the text. We mentioned one way of dealing with this common "overlap problem" in #milestones above. Another is to use "stand off" markup, in which the units analysed are represented not by XML elements but by pointers. Chapter 20 of the TEI Guidelines discusses these and other techniques in more detail.

6. The TEI Header

149 Every TEI document must have a TEI Header, represented by a `<teiHeader>` element. This is a container for all the metadata associated with the digital document itself, analogous to the title page of a printed book. In digital libraries and other online repositories, it is customary for the metadata associated with each digital document to be stored or managed separately, for example in a database structure for reasons of efficiency; it is also common practice to expose a subset of such metadata to web search engines according to one or more common standards such as Dublin Core. The TEI Header element is provided as a way of storing all such information in one place, independent of how it may be used. Of course, the scope and quantity of data collected in the Header may vary considerably in documents prepared for different purposes or by different projects. Furthermore, at least as originally conceived, the content of a TEI Header may not necessarily conform to the highest or most precise of cataloguing standards. Nevertheless it serves a useful purpose as a place where (for example) the data creator and the data curator can share information.

150 Originally, the TEI header was designed to suit two rather different requirements. On the one hand, it was intended to assist bibliographers and librarians faced with the (then) new problem of documenting 'electronic books'; on the other it was intended to meet the needs of researchers working with digital text collections, and needing to document the 'coding practices' applied to them. For the researcher, the important thing about the Header is that it provides a place for everything, supporting as far as possible the full range of divergent practice in different research communities, without imposing such barriers as a detailed knowledge of cataloguing standards or practices. For the librarian, the important thing about the TEI Header is that it conforms to standard bibliographic models, using similar terminology, and that it provides a single source of information for bibliographic description of a digital resource, with some established mappings to other such records (such as MARC or EAD). There is (naturally) some tension between these two perspectives, and a need for the elaboration of detailed profiles or "Guides to Best Practice" for different communities.

151 The TEI header has four main components, corresponding to the parts defined in one of the first attempts at a universal bibliographic description the International Bibliographic Standard Description (ISBD) :

- `<fileDesc>` (file description) contains a full bibliographic description of an electronic file.
- `<encodingDesc>` (encoding description) documents the relationship between an electronic text and the source or sources from which it was derived.
- `<profileDesc>` (profile description) provides a detailed description of non-bibliographic aspects of a text, specifically the languages and sublanguages used, the situation in which it was produced, the participants and their setting.
- `<revisionDesc>` (revision description) summarizes the revision history for a file.

6.1 The file description

152 Since it is not usually a pre-existing document, the structure of a TEI Header can be fairly tightly prescribed by the TEI. The first and only mandatory part, known as the file description, is represented by a `<fileDesc>` element, which contains three mandatory parts the title statement, publication statement and source description. A minimal TEI Header thus looks something like this:

```
<teiHeader>
 <fileDesc>
  <titleStmt>
   <title>Title of the work</title>
  </titleStmt>
  <publicationStmt>
   <p>Information about the publication of the work</p>
  </publicationStmt>
  <sourceDesc>
   <p>Information about the source from which the work was derived</p>
  </sourceDesc>
 </fileDesc>
</teiHeader>
```

153 Within the four main sections of the TEI header, many elements are possible; we can give only a superficial overview here. For example, the `<fileDesc>` can also contain an `<editionStmt>` documenting the particular edition of the resource being described, a `<seriesStmt>` if it is published as part of a series, an `<extent>` element to indicate its size , and even a `<notesStmt>` to capture notes or comments of various types.

154 The mandatory title statement, as its name suggests, provides a title for the resource, together with information about the people or agencies responsible for its intellectual content, in much the same way as the title of printed book, or a conventional catalogue record. For example:

```
<titleStmt>
 <title xml:lang="sk">Yogadarśanam (arthāt yogasūtrapūphah).</title>
 <title>The Yoga sūtras of Patañjali: a digital edition.</title>
 <author>Patañjali</author>
 <funder>Wellcome Institute for the History of Medicine</funder>
 <principal>Dominik Wujastyk</principal>
 <respStmt>
  <name>Wieslaw Mical</name>
  <resp>data entry and proof correction</resp>
 </respStmt>
 <respStmt>
  <name>Jan Hajic</name>
  <resp>conversion to TEI-conformant markup</resp>
 </respStmt>
</titleStmt>
```

155 In this example, the resource has two titles, one of which is given in Sanskrit. Details of its original author, the funder of the digital edition, the principal investigator, and other people with more specialized responsibilities are also provided. Many digital library projects will have their own rules about, for example, the format of names, or the use of authority lists to provide titles, which may vary considerably: the TEI therefore permits a more exact tagging of components such as proper names (for example by distinguishing forenames and surnames) but does not require it.

156 The file description describes the “file” (i.e. the whole of a digital resource — it may of course consist of several operating system files). The *publication statement* describes how this resource may be obtained, in much the same way as the imprint of a printed book suggests who is responsible for distributing or publishing it. Even if the TEI document is a private document which is not actually available anywhere beyond the owner's private hard disk, that fact must be mentioned here; more usually, of course, a header will be created only for a TEI resource which is being shared. The publication statement allows you to specify who is making the resource available, any identifier used for it, such as a catalogue number or Resource Identifier, and information about the terms under which it is distributed such as a licence, as shown here:

```
  <publicationStmt>
   <publisher>Humanities Media and Computing center</publisher>
   <pubPlace>University of Victoria</pubPlace>
   <date when="2011-08-04">19 August 2011</date>
   <idno type="filename">maladies_des_femmes.xml</idno>
   <availability status="free">
    <p>Copyright 2011. This text is freely available provided the text is
distributed with the header
    information provided.</p>
    <p xml:lang="fr">Les droits de reproduction des gravures ont été achetés de la
Bibliothèque Nationale de
     France grâce à une subvention accordée par le <ref target="http://www.sshrc-
crsh.gc.ca/">Conseil de
     recherches en sciences humaines du Canada</ref>. Les autres éléments du
projet (les contributions
     des éditeurs, les transcriptions des textes, l'encodage et le code) sont
distribués sous les termes de
     cette licence: <ref target="http://creativecommons.org/licenses/by-nc-nd/2.5/
ca/deed.fr_CA">Creative
     Commons Paternité - Pas d'Utilisation Commerciale - Pas de Modification 2.5
Canada</ref>.</p>
   </availability>
  </publicationStmt>
```

157 This example shows a typical mixture of structured elements and informal prose. The name and address of the agency responsible for distribution of the file have been distinguished as <publisher> and <pubPlace> respectively (a more detailed address, using the <address> element we saw above, could also be used, of course); in addition a date of publication and a filename to identify the resource have been provided. Since the terms of availability here are a little complex, they have been presented informally as a sequence of paragraphs of prose, containing references to other online documents as appropriate. In a simpler situation, the TEI <licence> element might have been used.

6.2 Varieties of source description

158 The next and only other mandatory component of the <fileDesc> is the source description, represented by the <sourceDesc> element. Even for documents which are "born digital" and thus have no pre-existing source, this element must be provided. Its purpose is to document formally the object or objects from which the TEI document has been derived, using traditional bibliographic terms. The <sourceDesc> may contain a simple paragraph of description, or it may contain one or more of the specialised elements for bibliographic description provided by the TEI.

159 As a first example, consider a document which has no form of existence beyond its digital version. We may represent this using a pared-down <bibl> :

```
<sourceDesc>
 <bibl>
  <title>Manifeste des Digital humanities</title>. <author>Marin Dacos et
al.</author> Available from <ref target="http://tcp.hypotheses.org/318">http://
tcp.hypotheses.org/318</ref>
   <date when="2010-05-21"/>
 </bibl>
</sourceDesc>
```

160 or simply give a short paragraph of explanation

```
<sourceDesc>
 <p>No source: this is a born digital document.</p>
</sourceDesc>
```

161 It is often the case that a digital edition derives from a specific printed source. This may also be described using the <bibl> element:

```
<sourceDesc>
   <bibl xml:id="Sue1846">
    <author><surname>Sue</surname>, <forename>Eugène</forename></author>
    <title level="m">Martin, l'enfant trouvé : Mémoires d'un valet de
chambre</title>
    <imprint>
     <publisher>C. Muquardt</publisher>
     <pubPlace>Bruxelles</pubPlace>
     <pubPlace>Leipzig</pubPlace>
     <date when="1846">MDCCCXLVI</date>
    </imprint>
   </bibl>
  </sourceDesc>
```

162 For digital transcriptions of audio or video recordings, a specialised `<recordingStmt>` element should be used. This can be used to store any technical details of the recording itself, and also to store detailed biographical or sociological data about the participants in a transcribed dialogue.

```
  <sourceDesc>
   <recordingStmt>
    <recording type="audio" dur="P30M">
     <respStmt>
      <resp>Location recording by</resp>
      <name>Sound Services Ltd.</name>
     </respStmt>
     <equipment>
      <p>Multiple close microphones mixed down to stereo Digital Audio Tape,
standard play, 44.1 KHz
       sampling frequency</p>
     </equipment>
     <date>12 Jan 1987</date>
    </recording>
   </recordingStmt>
  </sourceDesc>
```

163 For other non-book sources, such as the postcard mentioned above, the `<bibl>` element can also be used:

```
<sourceDesc>
 <bibl>
  <title level="m">The Bathing Beach, Brighton, in 1845 [postcard]</title>
  <respStmt>
   <resp>Lithograph by</resp>
   <name>G. F. Bragg</name>
  </respStmt>
  <respStmt>
   <resp>after a drawing by</resp>
   <name>R. H. Nibbs</name>
  </respStmt>
  <publisher>K. J. Bredon's Bookshop</publisher>
  <pubPlace>10 East Street, Brighton</pubPlace>
 </bibl>
</sourceDesc>
```

164 For a digital edition of a manuscript or early print source, where it is important to record the characteristics of the specific copy being transcribed, the detailed elements proposed by the TEI for the descriptive cataloguing of manuscripts may be more appropriate. Such descriptions use a distinct element, the `<msDesc>` which has quite an elaborate internal structure, as the following example shows:

```
  <sourceDesc>
   <msDesc>
    <msIdentifier>
     <country>France</country>
     <settlement>Paris</settlement>
     <repository>Archives nationales</repository>
     <collection>Commerce et Industrie</collection>
     <idno>F/12/5080</idno>
    </msIdentifier>
    <msContents>
     <p>Minute d’un rapport de proposition à la Légion d’honneur fait, en 1850,
par le ministre du Commerce
     et de l’Agriculture et président de la Société de géographie, Jean-Baptiste
Dumas, au Président de
     la République, en faveur des frères d’Abbadie, Antoine (1810-1897) et Arnaud
(1815-1893), auteurs
     d’un voyage en Abyssinie.</p>
    </msContents>
    <physDesc>
     <p>Deux feuilles de papier 24 x 12 cm ; écriture à l’encre noire.</p>
     <handDesc>
      <handNote xml:id="AA" scope="major">Antoine d’Abbadie</handNote>
      <handNote xml:id="DJB" scope="minor">Jean-Baptiste Dumas</handNote>
      <handNote xml:id="EPR" scope="minor">membre inconnu du cabinet du
ministre</handNote>
     </handDesc>
    </physDesc>
   </msDesc>
  </sourceDesc>
```

6.3 The encoding description

165 The second major division of the TEI Header is the *encoding description*, which is represented by an `<encodingDesc>` element. This optional element may be used to supply information about almost any aspect of the encoding process itself, either simply summarized as running text, or

provided within more specific elements. Taken as a whole, the content of a full <encodingDesc> approximates to the kind of information typically found in a technical manual associated with a project.

166 Some of its components are entirely documentary, for example:

- <projectDesc>: notes on the overall goals of the project;
- <samplingDecl>: notes on the sampling principles applied;
- <editorialDecl>: notes on the editorial principals applied, for which further specialised elements such as <correction>, <normalization>, <quotation>, <hyphenation>, <segmentation>, <interpretation> are provided.

167 Here is an example showing some of these elements in use:

```
  <encodingDesc>
   <projectDesc>
    <p>Texts collected for use in the Claremont Shakespeare Clinic, June 1990.</p>
   </projectDesc>
   <samplingDecl>
    <p>Each text contains a sample of up to 2000 words, running from the start of
the document to the end of
    the sentence after the 2000 word mark. For the purposes of word counting,
hyphens and apostrophes were
    treated as spaces. </p>
   </samplingDecl>
   <editorialDecl>
    <normalization>
     <p>Word forms broken by end of line hyphenation have been reconstructed
without comment. The hyphen
     has been removed except for hyphenated forms attested elsewhere in the text.
</p>
    </normalization>
    <quotation marks="all" form="std">
     <p>All quotation marks have been removed. Direct speech is represented by the
use of the <gi>said</gi>
      tag; other quoted material is represented by means of the <gi>q</gi> tag.
</p>
    </quotation>
   </editorialDecl>
  </encodingDesc>
```

168 Such documentation is of course useful only to a human reader. However, other components of the encoding description are intended for use by automated processes. Typically they provide a set of declarations for particular codes which are subsequently referenced or used in the body of the text. Examples include:

- `<charDecl>` : XML requires the use of Unicode throughout a document, Non-standard glyphs or characters may however be indicated in a TEI document using the `<g>` element to reference definitions for such things provided by this element;

- `<classDecl>`: any kind of classification system or taxonomy can be declared using this element; parts of a text can then indicate the classification code or codes associated with them by using (for example) the `@ana` attribute. In a TEI Corpus, this element is typically provided by the corpus header so that individual text headers can point to it using the `<catRef>` element.
- `<refsDecl>`, `<geoDecl>`, `<metDecl>`, `<fsdDecl>`, `<variantEncoding>` : provide a home for information concerning the encoding of reference systems, geographical information, systems of metrical analysis, feature systems for linguistic analyses, or the encoding of textual variation. Other similarly technical declarations can be added.

169 We give more detailed examples for each of these three below.

170 First, a typical `<charDecl>` element might define a variant form of the character Z which the encoder wishes to distinguish in a transcription. This variant has two strokes, but can be replaced by a normal Z.

```
<charDecl>
 <glyph xml:id="z103">
  <glyphName>LATIN LETTER Z WITH TWO STROKES</glyphName>
  <mapping type="standardized">z</mapping>
  <mapping type="PUA">U+E304</mapping>
 </glyph>
</charDecl>
```

171 Occurrences of this variant form in the transcription can now be distinguished using a `<g>` element to reference the above definition by means of its `@xml:id` value as usual. A processor can choose to render it using either the standardized mapping or the nonstandard code given by the PUA [1] mapping.

```
<p> ... mulct<g ref="#z103"/> ... </p>
```

172 Next we consider the `<classDecl>` element. This can be used to define any kind of private classification system or taxonomy. For a collection of newspaper articles we might use a taxonomy like the following:

```
<classDecl>
 <taxonomy xml:id="size">
  <category xml:id="large">
   <catDesc>story occupies more than half a page</catDesc>
  </category>
  <category xml:id="medium">
   <catDesc>story occupies between quarter and a half page</catDesc>
  </category>
  <category xml:id="small">
   <catDesc>story occupies less than a quarter page</catDesc>
  </category>

<!-- etc -->
 </taxonomy>
 <taxonomy xml:id="topic">
  <category xml:id="politics-domestic">
   <catDesc>Refers to domestic political events</catDesc>
  </category>
  <category xml:id="politics-foreign">
   <catDesc>Refers to foreign political events</catDesc>
  </category>
  <category xml:id="social-women">
   <catDesc>refers to role of women in society</catDesc>
  </category>
  <category xml:id="social-servants">
   <catDesc>refers to role of servants in society</catDesc>
  </category>

<!-- etc -->
 </taxonomy>
</classDecl>
```

173 An individual text containing (say) a story of less than a quarter page concerning the role of women in society would then reference this classification using the `<catRef>` element as follows:

```
<catRef target="#small #social-women"/>
```

174 The same mechanism can be used to document the codes used in simple linguistic analyses such as those discussed in chapter #ANA above.

175 Finally, we consider the <tagsDecl> element. This can be used just to list the elements actually used within a document, and also to define default formatting styles for them (typically using the W3C Cascading Stylesheet Language CSS). In the following example, we first define an two font styles, Italic and Roman, using CSS. We then state that within the TEI namespace anything tagged as <emph> or <hi> is by default in italic. We also state that the <text> element, and by default everything contained by it, uses Roman Font.

```
<tagsDecl>
 <rendition xml:id="IT" scheme="css">font-style: italic</rendition>
 <rendition xml:id="FontRoman" scheme="css">font-family: serif</rendition>
 <namespace name="http://www.tei-c.org/ns/1.0">
  <tagUsage gi="emph" render="#IT"/>
  <tagUsage gi="hi" render="#IT"/>
  <tagUsage gi="text" render="#FontRoman"/>
 </namespace>
</tagsDecl>
```

176 Within the body of a document, the @rendition attribute can over-ride the default styling for an element; see further section #rend above.

6.4 The profile and revision descriptions

177 The third major division of the TEI header is the rather oddly-named *Profile Description*, represented by a <profileDesc> element. Like the others, this is a group of optional notes or more specialised elements; they have in common simply that they are 'non-bibliographic'. Default members of the model.profileDescPart class include:

- <creation>: information about the origination of the intellectual content of the text, e.g. time and place. In a genetic edition, this may include a structured <listChange> element documenting each significant stage identified in the evolution of a text;
- <langUsage>: information about languages, registers, writing systems etc used in the text; each language used being identified by means of a <language> element and an identifying code taken from an ISO standard.

- `<textDesc>` and `<textClass>`: classifications applied to the text by means of a list of specified criteria or by means of a collection of pointers, respectively
- `<particDesc>` and `<settingDesc>`: information about the 'participants', either real or depicted, in the text

178 We noted above the availability of the `<catRef>` element as a means of classifying a text with respect to a predefined taxonomy. The `<profileDesc>` provides several complementary ways of doing this:

using `<catRef>`

by referring to a locally defined (e.g. in the corpus header) category

using `<classCode>`

by referring to some commonly agreed and externally defined category, such as the Universal Decimal Classification system,

using `<keywords>`

by assigning descriptive terms taken from a bibliographic controlled vocabulary or a tag cloud

179 In the following example, taken from the British National Corpus, a text is classified using all three possible methods:

```
<profileDesc>
 <creation>
  <date when="1962"/>
 </creation>
 <textClass>
  <catRef target="#WRI #ALLTIM1 #ALLAVA2 #ALLTYP3 #WRIDOM5 #WRILEV2 #WRIMED1
#WRIPP5 #WRISAM3 #WRISTA2 #WRITAS0"/>
  <classCode scheme="DLEE">W nonAc: humanities arts</classCode>
  <keywords scheme="COPAC">
   <term>History, Modern - 19th century</term>
   <term>Capitalism - History - 19th century</term>
   <term>World, 1848-1875</term>
  </keywords>
 </textClass>
</profileDesc>
```

180 Note that this categorization applies to the whole text. For more fine grained classification, the `@decls` attribute may be used to select the classification applicable to any declarable elemnt, for example to an individual `<div>`.

181 The fourth and last part of the TEI Header is the optional *revision description*, represented by a `<revisionDesc>` element which contains a list of `<change>` elements, each with `@date` and `@who` attributes, and each indicating significant stages in the evolution of a document; by convention the most recent such element is given first. The `<listChange>` element mentioned above may also be used here to to refer to identified stages in the evolution of the electronic file, as distinct from the text encoded. In a production environment an auomated version control system such as SVN will be used to keep detailed track of the evolution of a dcument; the TEI encoding of significant stages in the development of a document may be performed semi-automatically by such tools, or manually.

```
<revisionDesc>
 <listChange>
  <change when="2013-05-11">First complete draft</change>
  <change when="2013-04-07">Created header and document structure</change>
 </listChange>
</revisionDesc>
```

7. Customizing the TEI

182 As we have seen, the TEI is designed to support a very wide range of encoding choices. It can be used for a simple reading-oriented transcription of a primary source, whether that be an authorial manuscript, a printed literary work, an audio broadcast, or a dictionary. It can be used for enriched encodings in which many aspects of such texts are made explicit, so that software of all kinds can operate upon them, from visualisation tools and digital publishing systems to specialised statistical analysis packages. It can be used to provide additional annotations and metadata of all kinds. Almost no-one needs everything defined by the TEI, yet every one of its elements is of use or interest to someone. How should you go about choosing just the parts of the TEI you need? One major motivation for creating TEI documents in the first place is the possibility of sharing them with others, and integrating them with other TEI documents. How should you communicate the particular TEI encoding choices you have made to others so that such integration remains possible?

183 The TEI provides a way of adressing these concerns, as well as satisfying the important need for detailed project-specific documentation, by providing a set of elements which can be used both to specify a *schema* in terms of the names and formal properties of the elements and attributes it contains and also to document the way those elements and attributes are used in a given application.

184 The notion of a schema is fundamental to XML: it provides a kind of document grammar, naming the possible components and constraining the organization of an XML document. A schema makes it possible to express constraints such as "`<p>` elements may appear within `<div>` elements" or "every `<list>` element must contain at least one `<item>`." Rules of this kind are easily checked by an automatic processor (a *validator*). Constraints such as "Use the `<p>` element to contain paragraphs, not pages" or "use `<placeName>` for names of places, `<persName>` for names of people, and `<name>` for any other name" are much less easy to check automatically, since they relate to the semantics of the content rather than its organization. Such rules are defined by documents such as the TEI Guidelines, or by documents referrring to them. Understanding of these semantic constraints is extremely important for those developing software intended to take full advantage

of the marked up document, as well as for those wishing to create new documents encoded in the same way. This is particularly true if the schema in use permits a wide variety of elements with very similar meanings, as will be the case for an unmodified and uncustomised TEI schema.

185 For this reason, serious use of the TEI requires careful consideration of exactly which of its elements is appropriate to the needs of a project, and also perhaps of things which the project needs to specify more exactly than the TEI does. For example, the TEI makes no binding requirements for the possible values of the attribute `@type` used on `<div>` elements, since these are likely to vary greatly across different projects. In a given project, however, it is likely that standardising on an agreed set of values will be very helpful, and there will consequently be a need both to ensure that all documents use only the agreed set of values, and to ensure that documentation about what those values are and what they signify is maintained along with the rest of the schema, and readily available to XML-aware document preparation and editing systems (such as the widely used oXygen program) to help and guide the human document editor.

186 The TEI provides a special set of elements which can be used to create such a schema specification. The elements concerned (`<schemaSpec>`, `<moduleRef>`, `<elementSpec>`, `<classSpec>` and others) combine formal XML declarations for inclusion in a DTD or Schema with detailed documentation and examples, for inclusion in a technical manual about the encoding scheme being specified. For this reason, a document using these elements is informally known as an "ODD," for One Document Does it all: it provides information for a computer to process along with documentation of that information for a human being to read in a single integrated XML document. Unsurprisingly, the TEI system itself is expressed using this exact same set of elements, but here we focus on its use in the creation of a TEI customization.

7.1 Building a customization

187 The simplest TEI customization is the null customization, which simply says "permit every element defined by the TEI." The resulting schema, called `tei_all`, provides (at the time of writing) some 450 different elements, and many, many, different ways of solving often over-lapping encoding problems. The size and permissiveness of this schema alike make it more or less unusable for practical purposes — except for one very important one related to TEI *conformance*, to which we return below.

188 Probably the most widely used and frequently referenced TEI customization is tei_lite, a subset of some fifty elements claiming to satisfy the needs of 90% of TEI users, as evidenced by their actual practice in creating digital texts. This was originally produced for a TEI training workshop back in 1997, but with some updating has continued to be useful in a very wide range of contexts, to the extent that some people think that it *is* the TEI schema. `tei_lite` is included (as an Exemplar) with the standard release of TEI, in source and derived forms.

189 A quick glance at the XML source code for the tei_lite ODD shows that it appears to be a typical TEI document, with `<div>` elements containing `<head>`s, `<p>`s and `<list>`s, containing much discursive prose, as well as `<ptr>` elements for cross references and a few other specialised elements such as `<egXML>` for XML examples. The actual schema specified by this document is represented, towards the end of the document, by a `<schemaSpec>` element which contains declarations like the following :

```
  <schemaSpec ident="tei_lite" start="TEI teiCorpus">
   <moduleRef key="analysis" include="interp interpGrp pc s w"/>
   <moduleRef key="linking" include="anchor seg"/>
   <moduleRef key="tagdocs" include="att code eg gi ident val"/>
   <moduleRef key="tei"/>
   <moduleRef key="textstructure" include="TEI argument back body byline closer
dateline div docAuthor docDate docEdition docImprint docTitle epigraph front group
imprimatur opener postscript salute signed text titlePage titlePart trailer"/>
  </schemaSpec>
```

The `<moduleRef>` element is used to reference a TEI *module*. A module is simply a named container for a number of declarations for TEI elements and classes. The TEI currently defines 22 modules, each one corresponding to a chapter in the TEI Guidelines where its contents are described in detail. By default, a module reference implies that all the declarations it contains are to be transferred to the schema being specified, but the attributes `@include` and `@except` can be used to modify this default behaviour. Thus, in the above example the schema being created will include all the declarations found in the module called `tei`, but element declarations for only elements `<anchor>` and `<seg>` from the module called `linking`. Alternatively, the attribute `@except` might have been used to select everything from this module *except* for the elements named as its value.

190 A module makes available to the <schemaSpec> a declaration for each element selected, in the form of an <elementSpec> element. This has several components, most of which are visualised in the TEI reference documentation. It includes, for example:

- one or more <desc> elements to provide a short description of the function of the element, possibly in different languages;
- a <classes> element containing a <memberOf> element for each class of which the element is a member;
- an <attList> element, containing a structured list of all the attributes locally defined for the element, together with information about their datatype and any predefined set of values, expressed by means of the elements <attDef>, <valList> and <valItem> respectively;
- a <content> element providing a formal declaration of the legal content of the element, and optionally <constraint> elements used to express any additional usage constraints;
- one or more <exemplum> elements, each containing an examples of usage, as far as possible taken from real documents, plus commentary.

Appendix C of the TEI Guidelines provides nicely-formatted documentation for every defined TEI element on the basis of such declarations.

191 As well as selecting or excluding declarations, a <schemaSpec> can modify some parts of a declaration. A little later in the ODD for tei_lite, we find the following:

```
<elementSpec ident="TEI" mode="change">
 <attList><attDef ident="version" mode="delete"/></attList>
</elementSpec>
```

the effect of which is to provide a second declaration for the element <TEI>, along with that selected from the textstructure module. An ODD processor must reconcile or unify such duplications, under control of the attribute @mode . In this particular case, the effect is to change the <attList> declared for the <TEI> element by deleting the attribute @ident from it.

192 Exactly the same procedure is used in the remainder of the TEI Lite customization to remove from it some other unwanted attributes. For example, the attributes `@notBefore`, `@notAfter`, and others are provided by the attribute class `att.datable.w3c`. The easiest way of removing them is therefore to provide an additional declaration for the class as follows:

```
<classSpec type="atts" ident="att.datable.w3c" module="tei" mode="change">
 <attList>
  <attDef ident="notBefore" mode="delete"/>
  <attDef ident="notAfter" mode="delete"/>
  <attDef ident="from" mode="delete"/>
  <attDef ident="to" mode="delete"/>
 </attList>
</classSpec>
```

193 An ODD specification can include or exclude elements and attributes in these and other ways. It can also modify an existing declaration by adding additional constraints to it. For an example, consider the TEI customization known as Epidoc, which is widely-used by epigraphers and others working with inscriptions from the ancient world. As we noted above, the `@type` attribute on the `<div>` element is not constrained in any way by the TEI. The Epidoc community however has decided that it wishes to enforce the presence of this attribute and to permit only six predefined values for it. Here is the fragment of the Epidoc ODD which achieves that effect:

```
  <elementSpec ident="div" mode="change" module="textstructure">
   <attList>
    <attDef ident="type" mode="replace" usage="req">
     <valList type="closed">
      <valItem ident="apparatus">
       <desc>to contain apparatus criticus or textual notes</desc>
      </valItem>
      <valItem ident="bibliography">
       <desc>to contain bibliographical information, previous publications,
etc.</desc>
      </valItem>
      <valItem ident="commentary">
       <desc>to contain all editorial commentary, historical/prosopographical
discussion, etc.</desc>
      </valItem>
      <valItem ident="edition">
       <desc>to contain the text of the edition itself; may include multiple text-
parts</desc>
      </valItem>
      <valItem ident="textpart">
       <desc>used to divide a div[type=edition] into multiple parts (fragments,
columns, faces,
       etc.)</desc>
      </valItem>
      <valItem ident="translation">
       <desc>to contain a translation of the text into one or more modern
languages</desc>
      </valItem>
     </valList>
    </attDef>
   </attList></elementSpec>
```

Note that further constraints could be added to check that these values are correctly used: for example to check that a `<div type="textpart">` always has a `<div type="edition">` as its parent.

194 The Epidoc project shows how a particular research community can adapt the TEI to their own needs, and its customization is a good demonstration of the kinds of things which the ODD language makes possible.

7.2 Adding a new element

195 The ODD system also makes it possible to add entirely new elements to a schema. At its simplest this involves no more than the addition of a `<elementSpec>` for the new element to the schema specification, but deciding on the proper content for that new element does require some knowledge of the way the TEI system is designed. Choosing which class memberships should be specified for a new element, for example, is difficult without knowing which classes exist and how they interact. Choosing an appropriate content model for the new element similarly requires some thought about how other elements are defined, assuming that we wish our new element to behave consistently with the rest of the TEI. And finally, because this is a non-TEI element, we must take care to define it in a non-TEI namespace.

196 For example, suppose we wish to add a new element `<speciesName>` to mark up the names of botanical or other species appearing in a text. Our new element is semantically similar to the existing `<persName>` or `<name>` elements, so a good starting point will be to look at the ODD specification for other name-like elements. For example, in the `<elementSpec>` for the element `<persName>`, we see that it is a member of a model class `model.nameLike.agent`, which is a subclass of the more general `model.nameLike` class. Adding our new element to this class will ensure that it will appear in the content models of other TEI elements in the same places as other naming elements, without any further work on our part. We also note that `<persName>` is a member of several attribute classes, notably `@att.global` (which has several subclasses of its own) and `att.canonical`, both of which look useful for our purposes. Adding our new element to these classes will ensure that it uses these attributes in the same way as other name-like elements.

197 Next, we consider the content model of our new element. The easiest course of action would be to do the same as the other naming elements, and say that it can contain just text, mixed in with other phrase-level elements, elements such as `<g>`, and global elements. This is so common a requirement in the TEI scheme that there is a short cut (a "macro") defined for it called `macro.phraseSeq`. If however we want to represent the internal content of a `<speciesName>` (for

example to distinguish the "genus name" from the "specific name" component), we might want to define a more specific content model, possibly involving other new elements or other constraints. Whichever course of action we take will be explicitly documented in our ODD, so that other users of our data can see how we have customized the basic TEI framework.

7.3 Customization and Conformance

198 The TEI must reflect the diversity of practice within its user community if it is to continue to be useful. It is sometimes said that asking two academics the same question will always provide you with at least three different answers to it, and so it should be unsurprising to find considerable diversity in the TEI's recommendations.

199 Nevertheless, one of the purposes of the TEI Guidelines is to *guide* encoding practice. It is not a standard which tells you what to do (in the way that engineering standards, for example, specify exactly the shape and dimensions of electrical fittings); instead it tells you how to communicate what you have done to others. This is why, in the section of the Guidelines defining TEI conformance, one of the essential characteristics specified is the existence of an ODD, defining a schema which may be used to validate the documents in question, along with other possible constraints. Other essential criteria for TEI conformance include formal validity with respect to the schema tei_all, and respect for the defined semantics of the TEI elements used: a conformant TEI document may contain elements from many namespaces, but any elements from the TEI namespace must have the semantics defined for them by the TEI.

200 Consider for example the redoubtable English nineteenth century novelist Bulwer-Lytton. In one corpus of TEI documents we might hope to find his full name tagged, and associated with a definition of the person himself, as follows:

```
  <persName ref="http://en.wikipedia.org/wiki/Edward_Bulwer-
Lytton,_1st_Baron_Lytton"><forename>Edward</forename><forename>George</forename><surname
type="linked">Bulwer-Lytton</surname>,
   <roleName>Baron Lytton of <placeName>Knebworth</placeName>
   </roleName></persName>
```

201 But in others we will find versions with less informative markup or using different TEI elements, such as

```
<p>
 <rs type="name">Baron Lytton of Knebworth</rs>
</p>
```

or

```
<p>
 <name type="person" key="BWLY">The Right Honourable the Lord Lytton PC</name>
</p>
```

or

```
<p>
 <persName ref="#BWLY">First Baron Lytton of Knebworth</persName>
</p>
```

Such variations are only to be expected: projects vary considerably in their priorities. For some, it is enough just to show that a string of words is a name; for others, it is important to distinguish names of persons from names of places; for yet others it is essential to disambiguate named entity references, as they are known. The outputs from such projects clearly cannot simply be combined in a straightforwardly interoperable way, without some additional effort. Can they all be considered TEI-conformant?

202 An explicit customization, using the features of ODD we have sketched in this section, is the easiest way for a project to communicate its own decisions about how the TEI should be applied, and hence the easiest way to ensure its documents are interchangeable with others. A project can simply suppress (say) the element `<rs>` from its schema, or specify a closed list of values for its `@type` attribute. It can choose to use one or other (or both) of the attributes `@key` and `@ref` to associate different versions of a person's name with data about the person intended, and hence disambiguate them. Such choices can be made explicit in a customized schema, and hence tell us which of the many very different approaches to tagging an individual's name has been adopted in a given set of documents. And because it is also possible to generate an ODD automatically from a

set of TEI XML documents, we can also distinguish an intended customization — expressing what the creators of a document intended to be their encoding practice — from an actual customization, expressing the encoding practices actually found.

203 If the TEI is designed to be customized, how can it also claim to be an interchange format? For "blind interchange" (also known as "interoperability"), the recipient of a document must be confident that it will contain only TEI elements, and that those elements have been used according to the rules defined by the TEI. Unmodified, those rules permit so much variation that it is all but impossible for a developer to anticipate every possibility that their software must be able to handle. The availability of an ODD greatly simplifies this task, both by limiting the elements available, and by optionally adding to the constraints they represent on data content.

8. Conclusion: what *is* the TEI?

204 When we talk about “the TEI” what do we mean? The administrative body which is supported financially by the members of the Text Encoding Initiative Consortium? When we say “the TEI sent me this newsletter” or “we support the TEI” we are probably talking about the Consortium as an organization. But when we say “The TEI now supports markup of musical notation” or “The TEI now provides methods for the encoding of genetic editions” we are talking about something rather different — the technical content of the TEI Guidelines. And when we say “that's a really TEI point of view” (for example, with reference to best practice in the long-term archiving of digital resources) or “the TEI exemplifies our approach to open source” we're clearly talking about something more than either a set of technical recommendations or the people who maintain or who choose to use them. Some people talk about “the TEI” as it were a kind of club or religion, with members and non-believers, to which some aspire and from which others wish to distinguish themselves. Within the noisy market place of the “Digital Humanities,” the TEI is a kind of senior member, an annoying parental figure for some, a benevolent one for others, something just too old-fashioned even to be considered for others. Yet, over the last decade, it has become increasingly clear that the TEI is part of what makes the digital humanities happen: it has become a part of the infrastructure everyone has to engage with, both technically and socially, once they start thinking about text or other forms of cultural resource in digital form. The TEI provides a toolkit with which to do that thinking, and, most importantly, it also reflects the thinking that is

done, both in its preoccupations and its occasional oddities. It is this sense of TEI, as an information architecture, which we have used throughout most of this text. However, it seems appropriate to conclude with some consideration of the TEI as an organization.

205 One of the odder things about the TEI, from some perspectives at least, is that it is not a government agency like ANSI, or a well-funded international organization like ISO, or an industrial consortium like the W3C, though organizationally the TEI Consortium has points in common with all of these. The TEI is a not-for-profit membership organization, powered entirely by a small amount of funding from institutions and individuals who care enough about its activities to put money into their support and maintenance and also by a large amount of volunteer effort from institutions and individuals who want to contribute to its continued development. Every year, the TEI hosts a conference where the membership meets to transact the necessary official business of electing officers, and of course to discuss the ways in which the TEI is intellectually strong or weak. The community of those using the TEI and contributing to its continued development is however much larger than the paid-up membership. Any interested person is free to propose modifications or extensions to the system, or to report errors or inconsistencies — and many people do. The TEI maintains an elected Technical Council which is responsible for assessing and acting upon all proposed modifications and managing production of regular new releases. This process is carried out in public, in the spirit of Open Source development. We thus find it natural to refer to the TEI Community, even though the boundaries and constituents of that user community remain largely uncharted. It is that community which owns the TEI, which speaks for it in the larger Humanities community, and which informs and determines its evolution.

206 The word “evolution” is not chosen lightly. The TEI of 2013 is not the same as that of 1998, nor even as that of 2009, although most of the element names are still the same. The organization and flexibility of the TEI as a system architecture enables it to adapt to the changing needs and priorities of its community, in much the same way as other life forms have evolved in response to a changing environment. In Burnard 2013 I have argued that this is the secret of the TEI's longevity: it is organized in such a way as to ensure that it can be modified simply and effectively, depending on the needs of its users. It seems likely therefore that the TEI will continue to evolve, both in taking on new areas of encoding, and in modifying what has already been proposed to keep up with the changing digital landscape. At the end of 2010, for example, release 2.0 of TEI P5 introduced a

whole new way of representing genetic or documentary editions defined over the preceding year by an externally-funded workgroup. A method of making TEI documents interoperable with a new standard for the representation of notated music was introduced at about the same time. More recently, ongoing work in the TEI's Special Interest Group on Ontologies ensures that TEI encoded texts will continue to be hospitable to emerging best practice in representing Linked Data.

207 The TEI as an organization is committed to improving the accessibility and comprehensibility of the TEI as a technical system. In addition to the TEI website, which acts as a complete online reference for the scheme, there is a continual need to provide up to date training materials, and to promote active engagement with those interested in using, developing, and maintaining the TEI by wikis, discussion groups, and whatever other appropriate means emerge in the future. The development of such resources is an ongoing task for all members of the TEI community and beyond. This little book is intended as a modest contribution to that agenda.

NOTES

1 PUA is short for Private Use Area: a Unicode concept permitting the definition of private codes for non-interchangeable characters

2 Detailed encoding manual for early print materials; in French.

3 Chinese translation of the TEI tutorial (P5 edition) and accompanying articles.

4 Detailed encoding manual for epigraphical and other ancient primary source materials.

5 The original one stop generic tutorial introduction to using the TEI. Also available in a French translation by Sophie David and Lou Burnard.

6 Italian translation of the TEI tutorial (P4 edition); accompanied by introductory articles on XML, on the transition from P4 to P5, and on TEI tools and techniques.

7 Detailed specification of a subset of TEI appropriate for use in capture of primary sources for digital library work.

8 Website providing documentation and guidance on using the TEI for the encoding of early printed books.

9 Short introduction to encoding with XML for absolute beginners.

10 Detailed overview of TEI P4 and related XML standards; in French.

11 Online tutorial site providing practical exercises and examples for the full range of TEI (P4) modules.

12 Technical article summarizing how SGML was used in TEI P3 to implement modularity, customization, and hyperlinking. Of interest in the history of ODD.

13 Technical description of the new ODD system developed for TEI P5.

14 Press release describing the relaunch of the TEI as a membership Consortium.

15 Brief article stressing the interpretive nature of markup and placing it within an ancient scholarly tradition.

16 Historical review of the development of the TEI to date.

17 Classic and influential polemic arguing for descriptive rather than procedural markup as a scholarly activity.

18 Classic statement of the “OHCO” hypothesis according to which text is fundamentally an "Ordered Hierarchy of Content".

19 Discusses use of TEI for the encoding and management of academic publications.

20 Essential collection of 17 articles originally published as a double issue of Computers and the Humanities in 1994. Major TEI contributors of the first generation provide overviews of the TEI approach to encoding specific text types, complemented by classic articles on TEI objectives and design, and on SGML.

21 Describes the development of a Data Category Registry (DCR) component for the Linguistic Annotation Framework; an ISO standard of major importance in the encoding of linguistic analysis.

22 Interesting perspective on TEI-XML as seen from outside the scholarly community.

23 Introduction to a retrospective collection of 12 selected papers produced for the TEI 10th Anniversary Conference, held at Brown University in November 1997.

24 Surveys existing transcription schemes for spoken language and the feasibility of using TEI as an interlingua amongst them.

25 Classic statement of markup theory as applied to the editing of primary sources.

26 Collection of articles by 24 leading practioners covering all aspects of electronic textual editing within the TEI paradigm. An early online version is available at http://www.tei-c.org/Activities/ETE/.

27 Detailed report on the editorial work leading up to the first edition of TEI P5.

AUTHORS