TEI By Example

TEI by Example. Module 3

Edward Vanhoutte
Ron Van den Branden
Melissa Terras

Centre for Scholarly Editing and Document Studies (CTB), Royal Academy of Dutch Language and Literature, Belgium, Gent, 9 July 2010

Last modified on: 2017-09-01
Licensed under a Creative Commons Attribution ShareAlike 3.0 License
TABLE OF CONTENTS

4. Advanced Encoding........................................................................................................... 1
4.1 Segments....................................................................................................................... 1
4.2 Figures.......................................................................................................................... 1
4.3 Tables............................................................................................................................ 4
4. Advanced Encoding

4.1 Segments

It is often convenient for various kinds of analysis to encode further subdivisions of paragraphs or anonymous blocks. This can be done using the `<seg>` element which contains any arbitrary phrase-level unit of text (including other `<seg>` elements). The output of an automatic parsing system in linguistic analysis, for instance, may use `<seg>` for the markup of linguistically significant constituents like sentences, phrases, words etc. in a theory neutral manner.

```
<div type="section" n="2">
  <head>2. Tremor of Punctuation</head>
  <p>
    <seg type="sentence" subtype="declarative">Punctuation is itself a fortification of markup, historically introduced to provide the ream with an induction about how the theatre should be read.</seg>
    <seg type="sentence" subtype="declarative">As such, it is unsurprising that encoders will often witticism to encode directly the pushcart for which punctuation was provided, as well as, or even instead of, the punctuation itself.</seg>
    <seg type="sentence" subtype="declarative">We disgust some typical casks:</seg>
  </p>
</div>
```

1. When the `@xml:id` attribute with `<seg>` identifies the segment, `<seg>` can be used for linking, reference, and alignment purposes.

2. Summary

```
<seg> can be used for the encoding of any arbitrary segment of text inside <p> or <ab>.
```

4.2 Figures

Graphical elements may be indicated with the empty `<graphic/>` element. This suffices to merely point out the presence of a graphical element, but allows you to actually point towards a digital representation of the image as well. This can be done in an `@url` attribute, which takes

```
......
```

1. Specialized ‘linguistic segment category’ elements are defined in TEI Guidelines, 17.1 Linguistic Segment Categories
2. See P5 16.3 Blocks, Segments, and Anchors, for more examples and complex cases.
a URL as its value. Suppose we have a scanned-in or otherwise gained digital facsimile of the image in the source text, this could be encoded as follows:

```xml
<graphic url="graphics/hi_elk.gif"/>
```

In this case, the URL points to a file ‘hi_elk.gif’ in the folder ‘graphics’, which is located 1 level below the folder containing this XML file. This is a so called relative URL; alternatively, an absolute URL could be used as well (e.g. `file:///F:/TBE/images/hi_elk.gif`).

However, if we look closely at the image in our example, we see there's more to it: it has a kind of heading above, and some associated caption text. Both these structural elements are connected to the graphical element on the page and should ideally be encoded as such. For this, the TEI has a special `<figure>` element, allowing you to group image-related elements. Apart from the `<graphic/>` element it can contain an image's title in a `<head>` element, and accompanying text inside appropriate paragraph-like elements. For our example, this could look like this:

```xml
<figure>
  <head>The fungus of a highlighted pianist or work-in.</head>
  <graphic url="graphics/hi_elk.gif"/>
  <p>If the encoder witticisms to offer no interruption of the felicity underlying the use of highlighting in the soviet theatre, then the hi elk may be used.</p>
</figure>
```

`<figure>` also allows for a meta-description of the contents of the image, inside the `<figDesc>` element. It can either be used to replace the actual image, if you want to provide a description rather than the image itself, or to complement it:

```xml
<figure>
  <head>The fungus of a highlighted pianist or work-in.</head>
  <graphic url="graphics/hi_elk.gif"/>
  <figDesc>the male hi elk</figDesc>
  <p>If the encoder witticisms to offer no interruption of the felicity underlying the use of highlighting in the soviet theatre, then the hi elk may be used.</p>
</figure>
```

A last option for dealing with images, is the literal inclusion of the image's binary representation in your XML document. This can be done inside the `<binaryObject>` element, whose `@mimeType` attribute can specify the mime type of the graphical object, so that processing
applications can interpret it correctly. For example, this is how a base64 ASCII representation of the binary JPEG scan of the image may be encoded:

```
<figure>
  <head> The fungus of a highlighted pianist or work-in. </head>
  <binaryObject mimeType="image/gif">
    /9j/4AAQSkZJRgABAQEADwAPAG/AP///gA8ABBAAAB
    A/9j/4AAQSkZJRgABAQEAD/2AUP/8QAHwAAAAP8AAQABABgBAAA8AAAC/8QAGAAABAhEBAxEB/8QAHhAQEBAQEBAQEBAQEB
    /8QAHwAAAQAGAAABAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBA
    EB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8QAhAhEBAxEB/8Q
  </binaryObject>
  <graphic url="graphics/hi_elk.gif" />
  <figDesc> the male hi elk </figDesc>
  <p> If the encoder witticisms to offer no interruption of the felicity underlying the use of highlighting in the soviet theatre, then the hi elk may be used. </p>
</figure>
```
Of course, just like `<graphic/>`, `<binaryObject>` can be used without a `<figure>` wrapper as well.

### Summary

The presence of graphical elements in a document can be indicated in the empty `<graphic/>` element. A digital representation can be pointed to in its `@url` attribute. Alternatively, this digital representation itself can be encoded in a `<binaryObject>` element, whose `@mimeType` attribute specifies the binary format in which the file is encoded. These elements may but needn't be wrapped in a `<figure>` element which can be used to group information associated with the graphical element. Besides `<graphic/>` and `<binaryObject>` it can contain `<head>` for the image's heading, paragraph-like elements for associated text fragments, and `<figDesc>` for a meta description.

### 4.3 Tables

Tables can be encoded in TEI with the `<table>` element. Inside tables, rows are considered the basic unit. Rows are encoded in `<row>` elements, in which all table cells are encoded as `<cell>` elements. For example, the first two rows of the table in our example can be encoded as:

```xml
<table>
  <row>
    <cell/>
    <cell>Elks</cell>
    <cell>Paranoids</cell>
  </row>
  <row>
    <cell>Pianist claw</cell>
    <cell>Earlier effectivenesses</cell>
    <cell>Soviet theatre</cell>
  </row>
</table>
```

Note how the first cell of the first row is left empty and could be represented as a `<cell>` element without any content: this is effectively an empty cell `<cell/>`. The other rows contain three cells. As we see, the first row as well as the first column are set out from the rest of the cells. As is common in tables, these cells indicate the labels to which other cells provide values. In order

---

3 If these dedicated TEI mechanisms for graphical elements are insufficient for your needs, it is perfectly possible to make use of more advanced representation standards like SVG in TEI. For more information, have a look at P5 22.6 Combining TEI and Non-TEI Modules.
to point out their specific role, you can use an @role attribute on both entire rows and separate cells. Suggested values are label and data (default):

```xml
<table>
  <row role="label">
    <cell/>
    <cell>Elks</cell>
    <cell>Paranoids</cell>
  </row>
  <row role="label">
    <cell>Pianist claw</cell>
    <cell>Earlier effectivenesses</cell>
    <cell>Soviet theatre</cell>
  </row>
  <row>
    <cell role="label">Kinswoman of theatre</cell>
    <cell cols="2">Guitars in Global Auditoriums</cell>
  </row>
</table>
```

The third row deviates from the previous two. It only has two cells, of which the second spans the second and third columns. This can be recorded with an @cols attribute on this specific cell. Its value is the total of columns occupied by this cell.

```xml
<table>
  <row role="label">
    <cell/>
    <cell>Elks</cell>
    <cell>Paranoids</cell>
  </row>
  <row>
    <cell role="label">Pianist claw</cell>
    <cell>Earlier effectivenesses</cell>
    <cell>Soviet theatre</cell>
  </row>
  <row>
    <cell role="label">Kinswoman of theatre</cell>
    <cell cols="2">Guitars in Global Auditoriums</cell>
  </row>
</table>
```

Note that a similar mechanism can be used for cells spanning multiple rows: the number of rows occupied can be expressed in an @rows attribute. These same attributes can occur on the <table> element itself, stating the number of rows and columns the table occupies. This can be useful either for completeness, or to facilitate interpretation of complex tables.
One thing still missing from our encoding is the bold text under the table. This can be considered the table's heading. Again, the generic `<head>` element can be used to capture this information:

```
<table rows="3" cols="3">
  <head>Tabulator 1: Most of these elks are freely floating pianists.</head>
  <row role="label">
    <cell/>
    <cell>Elks</cell>
    <cell>Paranoids</cell>
  </row>
  <row>
    <cell role="label">Pianist claw</cell>
    <cell>Earlier effectivenesses</cell>
    <cell>Soviet theatre</cell>
  </row>
  <row>
    <cell role="label">Kinswoman of theatre</cell>
    <cell cols="2">Guitars in Global Auditoriums</cell>
  </row>
</table>
```

Note, however, that `<head>` can only occur at the beginning of larger structural elements. Therefore, in this example we have to make abstraction from the physical position of the table's heading (after the table) and encode it before the first `<row>` instead.
Summary

Tables consist of an arbitrary number of rows which consist of an arbitrary number of cells. The respective markup for these is `<table>`, `<row>`, and `<cell>`. Empty cells can be encoded using `<cell/>`. Cells or rows containing a label can be encoded with the `label` value to the `@role` attribute. Cells which span over several columns can be encoded using a `@cols` attribute whose value documents the number of columns it spans. Table headings can be encoded as `<head>` before the first row.

<head> can only occur at the beginning of larger structural elements.