TEI BY EXAMPLE

MODULE 4: POETRY

Edward Vanhoutte

Ron Van den Branden

Melissa Terras

TABLE OF CONTENTS

5. Metaphorical Language

Just like any other aspect of the transcription and encoding process, analysing metaphorical language involves interpretation. Unlike the expression of this interpretation in structural markup, the analysis of metaphorical language addresses logical structures that probably will cross structural boundaries. Due to the primary focus of the TEI encoding scheme on representing structural characteristics of texts, and the syntactic requirement that all XML structures should nest properly, this kind of logical analysis will need a kind of workaround. One such workaround defines different analytic categories separately and links them to the relevant passages in the poems. A first requirement is that the smallest addressable structures be identified with an @xml:id attribute:[1]

```xml
<lg xmlns="http://www.tei-c.org/ns/1.0" xml:id="p001" type="poem">
  <lg xml:id="s001" type="stanza">
    <l xml:id="l001">Poppadom</l>
    <l xml:id="l002">Oatmeal</l>
    <l xml:id="l003">Bubble gum</l>
    <l xml:id="l004">Cut of veal</l>
  </lg>
  <lg xml:id="s002" type="stanza">
    <l xml:id="l005">Mince for pie</l>
    <l xml:id="l006">Frozen peas</l>
    <l xml:id="l007">Video for Guy</l>
    <l xml:id="l008">Selection of teas</l>
  </lg>
  <lg xml:id="s003" type="stanza">
    <l xml:id="l009">Paper towels/garbage bags</l>
    <l xml:id="l010">Pasta sauce and Parmesan</l>
    <l xml:id="l011">Pumpkin seed and olive oil</l>
  </lg>
  <lg xml:id="s004" type="stanza">
    <l xml:id="l012">Cheesy crisps and favourite mags</l>
    <l xml:id="l013">Kidney beans (1 large can)</l>
    <l xml:id="l014">Cling film and kitchen foil</l>
```

......

1 @xml:id provides a unique identifier for the element bearing the attribute. Its value must start with a letter, or _ and may be followed by one or more Named Characters (letter, digit, ., -, _, Unicode Combining Character, or Extender). See the XML Specification for more guidance on the use of @xml:id.

```
    </lg>
  </lg>
```

Example 26. Making structural units addressable via `@xml:id`.

These `@xml:id` values can later be used to identify spans of interpretive categories. Such interpretive spans are encoded as `<span>`, identifying the structural scope of this interpretation with the `@from` attribute marking its beginning and the `@to` attribute marking its end. Their values are values for an `@xml:id` attribute elsewhere in the document, prefixed with a hash character (#), in order to indicate it as the identifier part of a formal *URI* reference. Preferably, the `@resp` attribute is used to identify who is responsible for the interpretation.[2] Related `<span>` elements can be grouped in a `<spanGrp>` element. A `@type` attribute can characterise the kind of analysis. In our poem, one way of identifying all images concerned with food and non-food could be:

```xml
<lg xmlns="http://www.tei-c.org/ns/1.0" xml:id="p001" type="poem">
  <lg xml:id="s001" type="stanza">
    <l xml:id="l001">Poppadom</l>
    <l xml:id="l002">Oatmeal</l>
    <l xml:id="l003">Bubble gum</l>
    <l xml:id="l004">Cut of veal</l>
  </lg>
  <lg xml:id="s002" type="stanza">
    <l xml:id="l005">Mince for pie</l>
    <l xml:id="l006">Frozen peas</l>
    <l xml:id="l007">Video for Guy</l>
    <l xml:id="l008">Selection of teas</l>
  </lg>
  <lg xml:id="s003" type="stanza">
    <l xml:id="l009">Paper towels/garbage bags</l>
    <l xml:id="l010">Pasta sauce and Parmesan</l>
    <l xml:id="l011">Pumpkin seed and olive oil</l>
  </lg>
  <lg xml:id="s004" type="stanza">
```

......

---

2   The value of the `@resp` attribute must be a pointer to an element in the document header that is associated with a person asserted as responsible for some aspect of the text's creation, transcription, editing, or encoding. See chapter 14 Certainty and Responsibility of the TEI Guidelines.

```
    <l xml:id="l012">Cheesy crisps and favourite mags</l>
    <l xml:id="l013">Kidney beans (1 large can)</l>
    <l xml:id="l014">Cling film and kitchen foil</l>
  </lg>
</lg>
<spanGrp xmlns="http://www.tei-c.org/ns/1.0" resp="#RvdB" type="imagery">
  <span from="#l001" to="#l006">food</span>
  <span from="#l007">non-food</span>
  <span from="#l008">food</span>
  <span from="#l009">non-food</span>
  <span from="#l010" to="#l013">food</span>
  <span from="#l014">non-food</span>
</spanGrp>
```

Example 27. Identifying interpretive spans in a text with `<span>`.

Notice that the @`from` attribute is mandatory on `<span>`. The @`to` attribute is optional; when it is missing, the entire structure identified by the @`from` attribute will be taken as the context for the interpretation.

As will be clear from this example, this system of linking analyses to textual structures is not very economic when discontinuous structures share the same analysis. Another approach for logical analysis works from the opposite angle, by hooking text structures to specific interpretations. In order to do so, interpretive categories should be formally defined somewhere else, either in the same document or externally. This is done inside the `<interp>` element, bearing a unique @`xml:id` attribute. Related interpretive categories can be grouped inside an `<interpGrp>` element. Following example could identify the same semantic categories from the previous example:

```
<interpGrp xmlns="http://www.tei-c.org/ns/1.0" resp="#RvdB" type="imagery">
  <interp xml:id="food">food</interp>
  <interp xml:id="non-food">non-food</interp>
</interpGrp>
```

Example 28. Defining interpretive categories in `<interp>` **elements.**

Inside the transcription of the poem, reference can be made to these interpretations with an @ana attribute. Its value should always point to an identifier, either locally or externally. Suppose these interpretive categories are stored in a separate document named **analysis.xml**. Then the poem could be analysed as follows: [3]

```xml
<lg xmlns="http://www.tei-c.org/ns/1.0" type="poem">
  <lg ana="analysis.xml#food" type="stanza">
    <l>Poppadom</l>
    <l>Oatmeal</l>
    <l>Bubble gum</l>
    <l>Cut of veal</l>
  </lg>
  <lg xml:id="s002" type="stanza">
    <l ana="analysis.xml#food">Mince for pie</l>
    <l ana="analysis.xml#food">Frozen peas</l>
    <l ana="analysis.xml#non-food">Video for Guy</l>
    <l ana="analysis.xml#food">Selection of teas</l>
  </lg>
  <lg xml:id="s003" type="stanza">
    <l ana="analysis.xml#non-food">Paper towels/garbage bags</l>
    <l ana="analysis.xml#food">Pasta sauce and Parmesan</l>
    <l ana="analysis.xml#food">Pumpkin seed and olive oil</l>
  </lg>
  <lg xml:id="s004" type="stanza">
    <l ana="analysis.xml#food">Cheesy crisps and favourite mags</l>
    <l ana="analysis.xml#food">Kidney beans (1 large can)</l>
    <l ana="analysis.xml#non-food">Cling film and kitchen foil</l>
  </lg>
</lg>
```

......

3   Of course, analyses can be much more complex. The TEI provides a generic mechanism for expressing complex hierarchies of interpretive relationships, called "Feature Structures." See chapter 12 Feature Structures of the TEI Guidelines for a detailed treatment.

Example 29. Linking text structures with interpretations with the `@ana` **attribute.**

SUMMARY

For the study of metaphorical language in poetry, logical structures should be identified and grouped into interpretive categories which can be related to other such categories. These interpretive categories with their logical relationships can be stored inside or outside the XML document containing the analysed text. In the poem, reference can be made to these internally or externally documented analyses.