Encoding names and named entities

Magdalena Turska

28 June 2014

Names, People, and Places

Names and other references to objects appear in most texts. Exactly how this appearance is made can very significantly differ - from text to text, but between references within the same text as well..

> *"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"*
> *Mr. Bennet replied that he had not.*
> *"But it is," returned she; "for Mrs. Long has just been here, and she told me all about it."*
> *Mr. Bennet made no answer.*

Now know ye that We have consented and do by these Presents signify Our Consent to the contracting of Matrimony between Our Most Dearly Beloved Grandson Prince William Arthur Philip Louis of Wales, K.G. and Our Trusty and Well-beloved Catherine Elizabeth Middleton

References are not the entities which they refer to

One entity(person, place, organisation) might be known by many names or might be referred to by some other description entirely.

> *"Why, my dear, you must know, Mrs. Long says that Netherfield is taken by a young man of large fortune from the north of England; that he came down on Monday in a chaise and four to see the place, and was so much delighted with it, that he agreed with Mr. Morris immediately; that he is to take possession before Michaelmas, and some of his servants are to be in the house by the end of next week."*
>
> *"What is his name?"*
>
> *"Bingley."*

Names in the TEI

TEI provides several ways of marking up names and nominal expressions:

- <rs> ("referring string") -- any phrase which refers to a person or place, e.g. ‘the girl you mentioned’, ‘my husband’...
- <name> - any lexical item recognized as a proper name e.g. ‘Siegfried Sassoon’ , ‘Calais’, ‘John Doe’ ...
- <persName>, <placeName>, <orgName>: ‘syntactic sugar’ for `<name type="person">` etc.
- A rich set of elements for the *components* of such nominal expressions, e.g. <surname>, <forename>, <geogName>, <geogFeat> etc.

References may be also ambiguous

```
<s>Jean likes <name ref="#NN123">Nancy</name>
</s>
```

Using a more precise element (<persName> or <placeName>) is one way of resolving the ambiguity; another is to follow the pointer:

```
<person xml:id="NN123">
 <persName>
  <forename>Nancy</forename>
  <surname>Ide</surname>
 </persName>
<!-- ... -->
</person>
```

or...

```
<place xml:id="N123">
 <placeName notBefore="1400">Nancy</placeName>
 <placeName notAfter="0056">Nantium</placeName>
<!-- ... -->
</place>
```

Names, People, and Places in TEI

<rs>, <name>, <persName>, <placeName>, <surname>, <forename> ...

```
 "Why, <rs>my dear</rs>, you must know, <persName>Mrs.
<surname>Long</surname>
</persName> says that
<placeName>Netherfield</placeName> is taken by a <rs>young man of large
fortune from the north of England</rs>; that he came down on Monday in a
chaise and four to see <rs>the place</rs>, and was so much delighted with
it,
that he agreed with <persName>Mr. <surname>Morris</surname>
</persName>
immediately; that he is to take possession before Michaelmas, and some of
his
servants are to be in the house by the end of next week."
"What is his name?"
"<persName>
 <surname>Bingley</surname>
</persName>."
```

Reference theory

Reference is a fundamental semiotic concept

- We can talk about the real world using natural languages because we know that some types of word are closely associated with real, specific, objects
- Proper names and technical terms are canonical examples of this kind of word
- ‘Wilfred Owen’ refers to a single real world entity; ‘Lyon’ and ‘River Thames’ to others: a specific place, a specific river respectively
- When we translate between natural languages, usually the proper names don't change, or are conventionally equivalent

Entities

Recognising the need to distinguish clearly the encoding of references from the encoding of referenced entities (occurrences in the real world) themselves, the TEI provides:

- <person> corresponding with <persName>
- <place> corresponding with <placeName>
- <org> corresponding with <orgName>
- and in addition <relation>, <event> and others

Why?

- To facilitate a more detailed and explicit encoding source documents (historical materials for example) which are primarily of interest because they concern objects in the real world
- To support the encoding of "data-centric" documents, such as authority files, biographical or geographical dictionaries and gazeteers etc.
- To represent and model in a uniform way data which is only implicit in readings of many different documents

Where to store information about named entities?

Information about a person is stored within a <person> element. Information about a group of people regarded as a single entity (for example ‘the audience’ of a performance) may be encoded using the personGrp element. These elements may appear only within a <listPerson> element, eg within <particDesc> (participant description) element in the <profileDesc> element of a TEI header

```xml
<profileDesc>
 <particDesc>
  <listPerson type="historical">
   <person xml:id="ART1">
    <persName>Arthur</persName>
   </person>
   <person xml:id="BERT1">
    <persName>Bertrand</persName>
   </person>
<!-- ... -->
  </listPerson>
 </particDesc>
</profileDesc>
```

Basic <person>

```
<person  xml:id="WO">
 <persName>
   <forename>Wilfred</forename>
   <forename>Edward</forename>
   <forename>Salter</forename>
   <surname>Owen</surname>
 </persName>
 <birth  when="1893-03-18">
   <placeName>Oswestry</placeName>, 18th March
    1893</birth>
 <death  when="1918-11-04">
   <placeName>Ors</placeName>, 4th November
    1918</death>
 <bibl  type="wikipedia">
   <ptr  target="http://en.wikipedia.org/wiki/Wilfred_Owen"/>
 </bibl>
</person>
```

What can we say about named entities?

Potentially, quite a lot...

```
<person  xml:id="ID1485">
  <persName>Ioannes Dantiscus</persName>
  <persName>Johannes von Höfen</persName>
  <persName>Jan Dantyszek</persName>
  <persName>Johannes Flachsbinder</persName>
  <persName>Ioannes de Curiis</persName>
  <birth  notBefore="1485-01-01"
    notAfter="1485-12-31">1485</birth>
  <death  when="1548-10-27">†1548-10-27</death>
  <occupation>diplomat, neo-Latin poet and traveller</occupation>
  <occupation  from="1504-01-01"
    to="1504-12-31">1504 royal scribe</occupation>
  <occupation  from="1507-01-01"
    to="1507-12-31">1507 referendary for Prussian affairs at the court of Sigismund Jagiellon;
</occupation>
  <occupation  from="1508"  to="1513">1508-1513 royal envoy to Prussian towns and to the Prussian
assemblies;</occupation>
  <occupation  from="1515">1515 secretary of the Polish legation at the imperial court;
</occupation>
  <occupation  from="1516"  to="1532">in 1516-1532 envoy in the service of the king of Poland
Sigismund Jagiellon and emperors Maximilian and Charles V of Habsburg; </occupation>
  <event  when="1529">Kulm canon; </event>
  <occupation  from="1530"  to="1537">1530-1537 bishop of Kulm; </occupation>
  <occupation  from="1537"  to="1548">1537-1548 bishop of Ermland</occupation>
</person>
```

Traits, States, and Events

Inside entities there are generally three *classes* of information:

- <state>: more general-purpose, but usually a time-related property (e.g. occupation for a person, population for a place)
- <trait>: if you want to a distinguish between time-bound and static, use this for properties that (usually) don't change over time (e.g. eye colour for a person, location for a place)
- <event>: an independent event in the real world which may lead to a change in state or trait (e.g. birth for a person, a war for a place)

Additionally, all these elements are members of the 'datable' class so can have time/dating attributes.

Traits

Some typical traits of a person

- <faith>: faith, belief system, religion etc. of a person
- <langKnowledge>: linguistic knowledge of a person
- <nationality>: nationality (socio-politico status)
- <sex>: sex
- <socecStatus>: socio-economic status

Some typical traits of a place:

- <climate>: describes the climate
- <location>: describes where a place is (see later)
- <population>: describes its population
- <terrain>: describes its terrain

States

Some typical states for a person

- <occupation> an informal description of a person's trade, profession or occupation
- <residence> (residence) a person's present or past places of residence
- <affiliation> an informal description of a person's present or past affiliation with some organization
- <education> a description of the educational experience of a person
- <floruit> contains information about a person's period of activity

Events

For persons, only two specific event elements are defined: <birth> and <death>. Anything else must be defined using the generic <event> element and its *@type* attribute.

```
<person  xml:id="SS">
  <persName>Siegfried Loraine Sassoon</persName>
  <birth  when="1886-09-08">
    <placeName>
      <placeName>Weirleigh Mansion</placeName>
      <settlement>Matfield</settlement>
      <region>Kent</region>
    </placeName>
  </birth>
  <death  when="1967-09-01"/>
  <event  when="1914-08-04"  type="military">
    <desc>In service with Sussex Yeomanry on the day the United Kingdom
         declared war</desc>
  </event>
  <event  when="1933-12"  type="marriage">
    <desc>Married Hester Gatty in December 1933</desc>
  </event>
  <event  when="1945"  type="separation">
    <desc>Separated from his wife in 1945</desc>
  </event>
</person>
```

UNIVERSITY OF OXFORD

How do we identify the entity being named?

Every element which is a member of the att.naming class inherits two attributes from the att.canonical class:

@key provides an externally-defined means of identifying the entity (or entities) being named, using a coded value of some kind.

@ref provides an explicit means of locating a full definition for the entity being named by means of one or more URIs.

Arguably, *@key* is redundant, since *@ref* is defined as `anyURI`, this can point from the name instance to the *@xml:id* of metadata about the entity, prefixing it with a '#' if in the same file, or use a private URI syntax.

References take many forms

Even within a single language, in a single document, there may be many ways of referencing the same person:

```
 ...  <persName>Leslie Gunston</persName>....  <persName>Leslie</persName>
....
<rs>Wilfred's cousin</rs>
```

The *@ref* can be used simply to combine all references to a specified person:

```
 ....
<persName  ref="#LG">Leslie Gunston</persName>....
<persName  ref="#LG">Leslie</persName> ....
<rs  ref="#LG">Wilfred's cousin</rs>
<!-- ... elsewhere -->
<person  xml:id="LG">
  <persName>Leslie
     Gunston</persName>
<!-- everything we want to say about Leslie -->
</person>
```

Pointing Mechanisms

The ref attribute can take any kind of pointer.
Entity defined within the same XML document

```
That silly man<name ref="#DPB1" type="person">David Paul Brown</name> has
suffered ...
```

or in some other place, refered to by means of a URI

```
That silly
man <name ref="http://www.example.com/personography.xml#DPB1"
 type="person">David Paul Brown</name> has suffered ...
```

Multiple pointers: reference to ‘the Browns’ might be encoded

```
That wretched pair <name ref="#DPB1 #EBB1" type="person">the Browns</name>
came to dine ...
```

Organizational names

An organizations is any named collection of people regarded as a single unit. An <orgName> can point back to an <org> in the header.

```
<p>On <date when="1915-10-21">21 October 1915</date> Owen enlisted in the
<orgName ref="#AROTC">Artists' Rifles Officers' Training
    Corps</orgName>.</p>
```

```
<org xml:id="AROTC">
<!-- Information about the organization -->
</org>
```

Components of <persName> elements

if it's a person we can use specialized elements divided further into subparts

```
<p>
 <persName>
  <forename>Wilfred</forename>
  <forename>Edward</forename>
  <forename>Salter</forename>
  <surname>Owen</surname>
 </persName>
did not know <persName ref="#jsbach" xml:lang="fr">
  <forename type="composer">Jean-Sebastien</forename>
  <surname>Bach</surname>
 </persName>
</p>
```

Not to mention... <roleName> (e.g. ‘Emperor’), <genName> (eg ‘the Elder’) <addName> (e.g. ‘Hammer of the Scots’), <nameLink> a link between components (e.g. ‘van’) ...

plus handy attributes to categorize or sort them

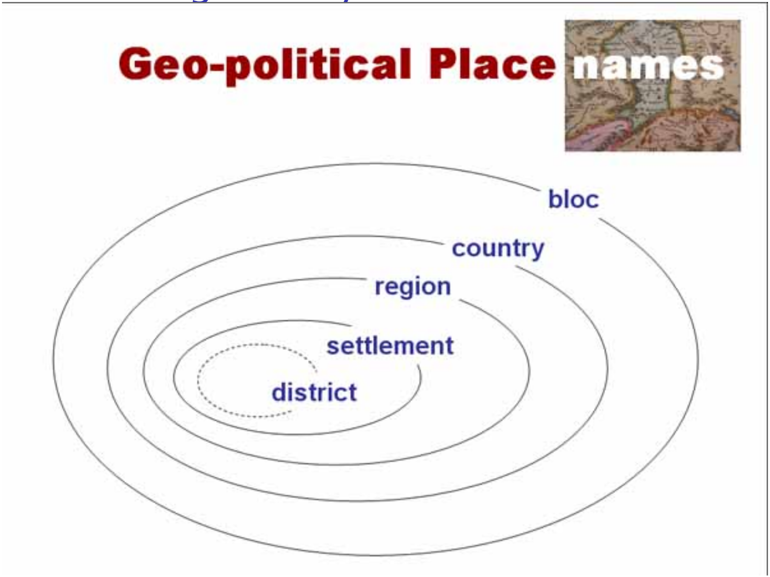
```
ersName ref="tag:projectname.org,2012:pn9">
```

Components of place names

- <placeName> (names can be made up of other names)
- <geogName> a name associated with some geographical feature such as a mountain or river
- <geogFeat> a term for some particular kind of geographical feature e.g. ‘Mount’, ‘Lake’

```
<placeName>
 <geogFeat>Mont</geogFeat>
 <geogName>Blanc</geogName>
</placeName>
```

Place names generally fall into a kind of hierarchy



Geo-political Place names

A place is defined by its <location>

The <location> element can contain

- a more or less well-structured description using the hierarchy of place name components mentioned earlier (a politico-geographical location)
- a set of geographical co-ordinates

```
<place  xml:id="craiglockhart">
 <placeName>Craiglockhart War Hospital</placeName>
 <settlement>Edinburgh</settlement>
 <region>Scotland</region>
 <country  key="UK">United Kingdom</country>
 <location>
  <geo>55.91812 -3.24019</geo>
 </location>
</place>
```

Another <location>

```xml
<place type="building">
 <placeName>Brasserie Georges</placeName>
 <location>
  <country key="FR"/>
  <settlement type="city">Lyon</settlement>
  <district type="arrondissement">Perrache</district>
  <placeName type="street">cours de Verdun</placeName>
 </location>
 <location>
  <geo>45.748 4.828</geo>
 </location>
</place>
```

A place can be fictional

```
<place  type="imaginary">
 <placeName>Atlantis</placeName>
 <location>
  <offset>fifty leagues beyond</offset>
  <placeName>Pillars of <persName>Hercules</persName>
  </placeName>
 </location>
</place>
```

Places can self-nest

```
<place  type="soverignState">
  <placeName>United Kingdom</placeName>
  <placeName  type="full">United Kingdom of Great Britain and Northern
     Ireland</placeName>
  <place  type="country">
    <placeName>Scotland</placeName>
    <place  xml:id="edinburgh"  type="city">
      <placeName>Edinburgh</placeName>
      <place  xml:id="craiglockhart2">
        <placeName>Craiglockhart War Hospital</placeName>
        <location>
          <geo>55.91812 -3.24019</geo>
        </location>
      </place>
    </place>
  </place>
</place>
```

<listPlace> in context of <settingDesc>

```xml
<settingDesc>
  <listPlace>
    <place xml:id="west01">
      <placeName>West Copice</placeName>
      <region>Shropshire</region>
      <note>'Westcopice' was approximately three-quarters of a mile
             east of Sheinton, on the south bank of the Severn opposite
             Buildwas, near the abbey ruins. Probably Henry Wood's manor
             or estate is named in this reference.</note>
    </place>
    <place xml:id="shei01">
      <placeName>Sheinton</placeName>
      <region>Shropshire</region>
    </place>
    <place xml:id="shro01">
      <placeName>Shropshire</placeName>
    </place>
  </listPlace>
</settingDesc>
```

<listOrg> example

```
<listOrg>
 <org xml:id="star01">
  <orgName>Star Chamber</orgName>
  <note>The Star Chamber (Latin: Camera stellata) was an English court
      of law that sat at the royal Palace of Westminster from the late
      15th century until 1641. </note>
 </org>
</listOrg>
```

W3C Date Formats

All these events are 'datable' and so can be associated with a more or less exact date or date range using any combination of the following attributes:

@when supplies the value of a date or time in a standard form

@notBefore specifies the earliest possible date for the event in standard form

@notAfter specifies the latest possible date for the event in standard form

@from indicates the starting point of the period in standard form

@to indicates the ending point of the period in standard form

The ‘standard form’ is that defined by W3C. All dates are normalised to the Gregorian calendar.

The most commonly-encountered format f[illegible]e date part of the when attribute is yyyy-mm-dd, but y[illegible]- -[illegible]n

Personal Relationships

The <relation> (relationship) element describes any kind of relationship or linkage amongst other entities

We distinguish ‘mutual’ relationships (e.g. sibling) from non-mutual or directed relationships (e.g. parent-of).

The following attributes are available:

@name supplies a name for the kind of relationship of which this is an instance

@active identifies the 'active' participants in a non-mutual relationship, or all the participants in a mutual one

@mutual supplies a list of participants amongst all of whom the relationship holds equally

@passive identifies the `passive` participants in a non-mutual relationship

Example

```
<person xml:id="SLS">
 <persName>Siegfried Loraine Sassoon</persName>
</person>
<person xml:id="HG">
 <persName>Hester Gatty</persName>
</person>
<person xml:id="GS">
 <persName>George Sassoon</persName>
</person>
<!--...-->
<relationGrp type="children">
 <relation name="parent" active="#SS"
  passive="#GS"/>
<!--...-->
</relationGrp>
```

Thank You!

Any Questions?

MARIE CURIE ACTIONS SEVENTH FRAMEWORK PROGRAMME DiXiT