

# Talk 4: Named Entities – People, Places, and Organisations

James Cummings  
@jamescummings

10 October 2014

## Names, People, and Places

Names and other references to objects appear in most texts. Exactly how this appearance is made can vary significantly - 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

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 proposals for the *components* of such nominal expressions, e.g. `<surname>`,  
`<forename>`, `<geogName>`, `<geogFeat>` etc.

# 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>Bingley</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, often the proper names don't change, or are conventionally equivalent

# Entities

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

All other types of names are  
`<name type="otherTypeOfName">`



## 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

## How do we represent this association?

*@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 unnecessary, 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 using a private URI syntax: 'myproject:abc123' described in the header.

## 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, referred 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 ...

- Private URI syntaxes can also be used and the algorithm for turning these back into a fully-formed URI documented in the header.

## 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.

## <listPerson> example

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

## Other attributes

There are other attributes as well:

*@role* may be used to specify further information about the entity referenced by this name, for example the occupation of a person, or the status of a place.

*@nymRef* provides a means of locating the canonical form (*<nym>*) of the **names** associated with the object named by the element bearing it.

# Name Examples

```
<p>... <name ref="#jsbach" type="person">Johann Sebastian Bach </name> the German composer was born in 1685... </p>
```

```
<p>... <name ref="grove:jsbach" type="person">Johann Sebastian Bach </name>the German composer was born in 1685... </p>
```

```
<p>... <name role="composer">Engelbert Humperdinck</name> was born in 1854... </p>  
<p>... <name role="singer">Engelbert Humperdinck</name>was born in 1936... </p>
```

## 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>
```



## References are 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="NN123">
  <placeName notBefore="1400">Nancy</placeName>
  <placeName notAfter="0056">Nantium</placeName>
<!-- ... -->
</place>
```

## Components of <persName> elements

```
<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="composé">Jean-Sébastien</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') ...

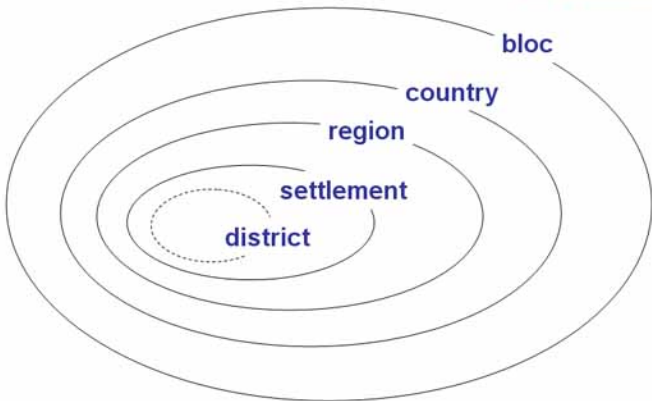
## Components of place names

- **<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**



# What can we say about named entities?

Potentially, quite a lot...

```
<person xml:id="VM1893">
  <persName xml:lang="ru">Владимир Владимирович Маяковский</persName>
  <persName xml:lang="fr">Wladimir Maïakowski</persName>
  <birth when="1893-07-19">7 July (OS) 1893,
<placeName ref="#BGDT" xml:lang="en">Baghdati, Georgia</placeName>
  </birth>
  <death when="1930-04-14"/>
  <occupation>Poet and playwright</occupation>
  <note>Among the foremost representatives of early-20th century Russian
Futurism.</note>
</person>
```

What elements should the TEI provide for such a purposes?

# Another Example

```

<person xml:id="ID1485">
  <persName>Ioannes Dantiscus</persName>
  <persName>Johannes von Höfen</persName>
  <persName>Jan Dantyszek</persName>
  <persName>Johannes Flachs binder</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 notBefore="1504-01-01"
    notAfter="1504-12-31">1504 royal scribe</occupation>
  <occupation notBefore="1504-01-01"
    notAfter="1504-12-31">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>

```

## Another basic &lt;person&gt;

```
<person xml:id="W0">
  <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>
```

## 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 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

## 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>
```

# A place encodes interpretation

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

# A place can be fictional, location imprecise

```
<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="craiglockhart">
        <placeName>Craiglockhart War Hospital</placeName>
        <location>
          <geo>55.91812, -3.24019</geo>
        </location>
      </place>
    </place>
  </place>
</place>
```

## Organizational names

An organization 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>
```

## <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>
```



## 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>Seperated from his wife in 1945</desc>
  </event>
</person>
```

## 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.

## 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="SS">
  <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>
```

# Nyms

The elements `<listNym>` and `<nym>` are used to document the canonical form of a name or name-component.

- `<nym>`
  - can contain `model.entryParts` (e.g. `<form>`, `<orth>`, `<etym>`) and may also include a number of other `<nym>`s
  - in addition to global attributes and `att.typed`, it includes the attribute `@parts` to point to constituent `<nym>`s
- `<listNym>` a list of canonical names
- `@nymRef` has been added to the attribute class `att.naming` to refer to the canonical name

# Example

```
<nym xml:id="J45">
  <form xml:lang="la">Iohannes</form>
  <nym xml:id="J450">
    <form xml:lang="en">John</form>
    <nym xml:id="J4501">
      <form>Johnny</form>
    </nym>
    <nym xml:id="J4502">
      <form>Jon</form>
    </nym>
  </nym>
  <nym xml:id="J455">
    <form xml:lang="ru">Ivan</form>
  </nym>
  <nym xml:id="J453">
    <form xml:lang="fr">Jean</form>
  </nym>
</nym>
```

## Next

That is a a far too quick overview of  
Names/People/Places.

Now let's have a short break before doing an exercise  
(if you don't need a break, just get stuck in).