Names / Dates / People / Places

James Cummings

December 2010

Names, People, and Places

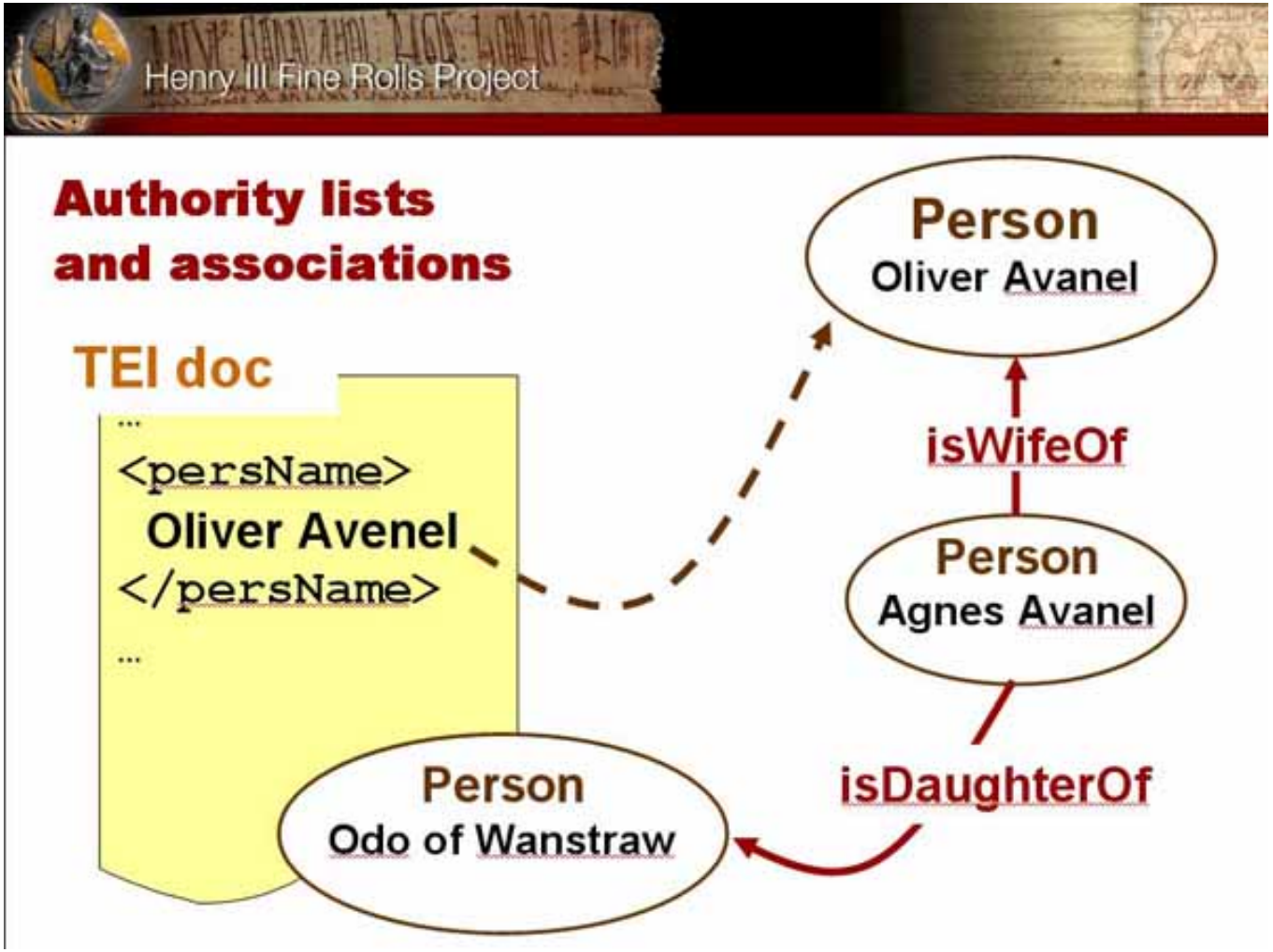
The TEI P5 chapter entitled *Names, Dates, People, and Places* gives recommendations on the encoding of names and of data about names.

- Aimed at the application of text encoding to a various range of fields that can span from history to geography, from onomastic and toponomatiscs to biography and prosopography, this chapter also deals with the definition and standard expression of temporal dimensions

Linking Names and their Referents

- In TEI P5 there are ways to link names of people/places/organizations with information about the entities themselves
- This is done following the general principle adopted in P5 of using URIs to make explicit connections between textual resources
- This application of so called 'stand-off markup' is particularly useful when
  - the objective of a specific encoding strategy is to study individuals and places mentioned in a certain text
  - the corpus of texts in question is large enough to justify the creation of separate structures to contain data about entities only mentioned -eventually with different referential strings- in the core texts

Example of Reference in Henry III Fine Rolls project



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 afre canonical examples of this kind of word
- 'Martin Luther King' 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

How do we represent this association?

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.

as well as the attributes

@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.

Arguably, *@key* is redundant, since *@ref* is defined as anyURI

Example

```
<p>... <name ref="#jsbach" type="person"> Johann Sebastian Bach </name> was a prolific
German composer ... </p>
```

Names as referents (1)

In a text we might find the same person referred to on different occasions in any number of different ways:

```
...<persName>Clara Schumann</persName>.... <persName>Clara</persName> ....
<persName>Frau Schumann</persName>
```

All of these names refer to the same *entity*

We can use an attribute on any naming element to specify which entity is being referenced:

- *@key* if we are supplying an externally-defined code for the entity
- *@ref* if we are pointing to a definition of the entity

Names as referents (2)

For example:-

```
....
<persName   ref="#CS">Clara Schumann</persName>....
<persName   ref="#CS">Clara</persName> ....

<persName   ref="#CS">Frau Schumann</persName>
<!-- ... elsewhere -->
<person   xml:id="CS">
  <persName   xml:lang="de">
    <forename   type="first">Clara</forename>
    <forename   type="middle">Josephine</forename>
    <surname   type="maiden">Wieck</surname>
    <surname   type="married">Schumann</surname>
  </persName>
</person>
```

The thing itself (1)

TEI provides special-purpose elements for maintaining structured information about named entities (as well as their names):

- <person>, <place>, <event>
- may be grouped into <listPerson>, <listPlace>, <listEvent>
- relationships can also be modelled, explicitly using <relation> or implicitly by context

```
<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, among the foremost representatives of early-20th century Russian Futurism.</occupation>
</person>
```

Personal Names

- <persName> (personal name) a proper noun/phrase referring to a person
- <surname> a family (inherited) name
- <forename> a forename, or given name
- <roleName> a name component indicating a particular role
- <addName> (additional name) a nickname or alias
- <nameLink> a connecting phrase or link used within a name ('van der')
- <genName> a generational name component

<persName> Example

```
<persName ref="#jsbach">
  <forename type="first">Johann</forename>
  <forename type="middle">Sebastian</forename>
  <surname>Bach</surname>
</persName>
<persName ref="#cjwshumann">
  <forename type="first">Clara</forename>
  <forename type="middle">Josephine</forename>
  <surname type="maiden">Wieck</surname>
  <surname type="married">Schumann</surname>
</persName>
```

Sub-elements of <persName>

A large number of sub-elements is available within <persName> for the various parts of a name; here is an example using some of them, each with attendant attributes such as *@type* and *@sort*.

```
<persName xml:lang="bg">
  <forename type="first" sort="2">Татяна</forename>
  <forename type="middle" sort="3">Александра</forename>
  <forename type="patronym" sort="4">Василева</forename>
  <surname sort="1">Тодорова</surname>
  <addName type="nick" xml:lang="bg">Сани</addName>
  <addName type="nick" xml:lang="en">Alex</addName>
</persName>
```

Another example

```
<persName    xml:lang="zh-tw">
   <forename>夏爾</forename>
   <forename>皮耶</forename>
   <surname>波特萊爾</surname>，
<roleName>
      <placeName>法國</placeName>象徵派詩人
   </roleName>
</persName>
```

Persons

- <person> provides information about an identifiable individual
  - for example a participant in a language interaction, or a person referred to in a historical source
- <personGrp> (personal group) a group of individuals
  - treated as a single person for analytic purposes
- <listPerson> (list of persons) a list of descriptions, each of which provides information about an identifiable person or a group of people
  - for example the participants in a language interaction, or the people referred to in a historical source
- <relationGrp> (relation group) information about relationships identified amongst people, places, and organizations
  - either informally as prose or as formally expressed relation links

Example

```
<listPerson type="composers">
  <person xml:id="kfabel">
    <persName>Karl Friedrich Abel</persName>
  </person>
  <person xml:id="magricola">
    <persName>Martin Agricola</persName>
  </person>
  <person xml:id="salkan">
    <persName>Siegfried Alkan</persName>
  </person>
  <person xml:id="cpebach">
    <persName>Carl Philipp Emanuel Bach</persName>
  </person>
<!--….-->
</listPerson>
```

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, among the foremost representatives of early-20th century Russian Futurism.</occupation>
</person>
```

What elements should the TEI provide for such a purposes?

Personal names

The <persName> element is repeatable and can, like all TEI elements, take the attribute *@xml:lang* to indicate the language of the content of the element, thus making it possible to supply name forms in different languages:

```
<person  xml:id="AdMic">
  <persName  xml:lang="lt">Adomas Mickevi□ius</persName>
  <persName  xml:lang="be">Адам Міцкевіч</persName>
  <persName  xml:lang="pl">Adam Mickiewicz</persName>
</person>
```

Organizations and organizational names

- <orgName> an organizational name
  - Any named collection of people regarded as a single unit
- <listOrg> a list of elements, each of which provides information about an identifiable organization
- <org> information about an identifiable organization such as a business, a tribe, or any other grouping of people

Place Names

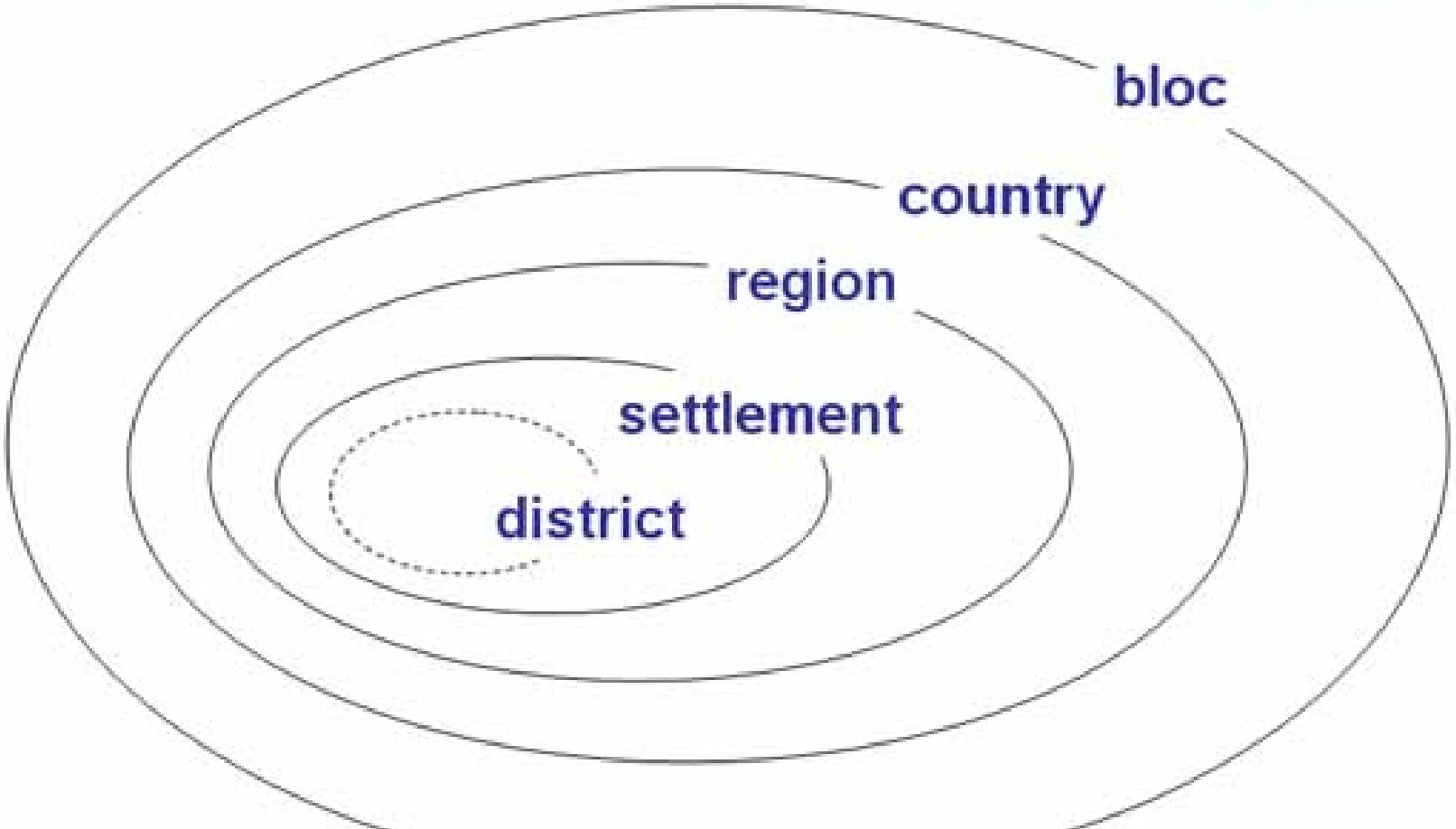
Name components:

- <placeName> an absolute or relative place name
- <geogName> geographical name a name associated with some geographical feature
  - such as Windrush Valley or Mount Sinai
  - <geogFeat> geographical feature name contains a common noun identifying some geographical feature contained within a geographic name, such as valley, mount etc.

Geo-political Place Names

- <district> the name of any kind of subdivision of a settlement
  - such as a parish, ward, or other administrative or geographic unit
- <settlement> the name of a settlement such as a city, town, or village
  - identified as a single geo-political or administrative unit
- <region> the name of an administrative unit such as a state, province, or county
  - larger than a settlement, but smaller than a country
- <country> the name of a geo-political unit, such as a nation, country, colony, or commonwealth
  - larger than or administratively superior to a region and smaller than a bloc
- <bloc> the name of a geo-political unit consisting of two or more nation states or countries

Geo-political Inclusion



Relative Place Names

- <offset> that part of a relative temporal or spatial expression which indicates the direction of the offset between the two place names, dates, or times involved in the expression
- <measure> contains a word or phrase referring to some quantity of an object or commodity, usually comprising a number, a unit, and a commodity name

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

New elements to representing data about the places those names can refer to: <listPlace> and <place>

- Geopolitical inclusion of places can be expressed by nesting and/or by stating explicit relationships: <relationGrp> and <relation>
- The conceptualization of *States*, *Traits* and *Events* inherited from the section on persons is also applied to places:
  - e.g. of traits: <population>, <climate>, <terrain>
  - the new element <location> is a kind of trait and is used to represent geographical location and geopolitical structures
  - within location, <geo> can be used to express latitude and longitude (recommended datum: WGS84)

A place example: Taipei region

```
<place xml:id="BGDT">
  <placeName xml:lang="zh-tw">士林夜市</placeName>
  <placeName xml:lang="en">Shilin</placeName>
  <location type="geopolitical">
    <country>Taiwan</country>
    <region>Taipei</region>
  </location>
  <location type="physical">
    <geo>25.0866 121.5254</geo>
  </location>
</place>
```

Places can be nested (unlike people)

```
<place xml:id="LT">
  <country>Lithuania</country>
  <country xml:lang="lt">Lietuva</country>
  <place xml:id="LT-VN">
    <settlement>Vilnius</settlement>
  </place>
  <place xml:id="LT-KA">
    <settlement>Kaunas</settlement>
  </place>
</place>
```

Outer Place Example

```
<place xml:id="leipzig" type="city">
  <placeName nymRef="#leipz">Leipzig</placeName>
  <location>
    <geo>51.333333 -12.383333</geo>
    <offset>at the confluence of</offset>
    <geogName>
      <geogFeat>River</geogFeat> Pleiße</geogName>
    <geogName>
      <geogFeat>River</geogFeat> White Elster</geogName>
    <geogName>
      <geogFeat>River</geogFeat> Parthe</geogName>
  </location>
  <location type="geopolitical">
    <district>Leipzig</district>
    <region type="state">Saxony</region>
    <country>Germany</country>
  </location>
<!-- inner place here -->
</place>
```

Example: Inner Place:

```
<place xml:id="tkirch" type="church">
  <placeName xml:lang="de" nymRef="#thomas #kirche">Thomaskirche</placeName>
  <placeName xml:lang="en" nymRef="#thomas #church">St Thomas' Church</placeName>
  <desc>
    <label>Religion</label> Lutheran church</desc>
  <desc>
    <label>Architecture</label> Gothic</desc>
  <event type="consacration" when="1496-04-10">
    <desc>The current building was consecrated on 10 April 1496 by the<persName>
        <roleName>Bishop</roleName> of <placeName>Merseburg</placeName>
      </persName>.</desc>
  </event>
  <event when="1723">
    <desc>It is most famous as the place where <persName ref="#jsbach">Johann Sebastian
Bach</persName> worked as a cantor.</desc>
  </event>
</place>
```

Example: or as explicit relation

```
<relation name="partOf" active="#tkirch" passive="#leipzig"/>
```

Sources

As for persons, responsibility and uncertainty about the sources can be asserted by using the attribute class att.editLike

```
<org type="tribe" resp="#herodotus">
  <orgName>The Maxyans</orgName>
  <country>Libya</country>
  <desc>According to Herodotus, they were a west Libyan tribe who said that they were
    descended from the men of Troy.</desc>
</org>
```

Traits, States, and Events

Information about people, places, and organizations, of whatever type, comprises a series of statements or assertions relating to:

- characteristics or *traits* which do not, by and large, change over time
  - <trait> contains a description of some culturally-determined characteristic attributed to a person or place.
- characteristics or *states* which hold true only at a specific time
  - <state> contains a description of some ongoing status or quality attributed to a person, place, or organization.
- *events* or incidents which may lead to a change of state or, less frequently, trait
  - <event> contains data relating to any kind of significant event associated with a person, place, or organization.
    - *@where* indicates the location of an event by pointing to a <place> element

Personal Traits

The model.persTraitLike class contains elements describing physical or socially-constructed characteristics or traits of a person. Members of the class comprise the following specific elements:

- <faith> the faith, religion, or belief set of a person
- <langKnowledge> (language knowledge) the state of a person's linguistic knowledge, either as prose or by a list of <langKnown> (language known) elements
- <nationality> an informal description of a person's present or past nationality or citizenship
- <sex> the sex of a person
- <age> the age of a person
- <socecStatus> (socio-economic status) an informal description of a person's perceived social or economic status

Traits Example

```
<person xml:id="fmendelssohn">
  <persName>
    <forename>Felix</forename>
    <surname>Mendelssohn</surname>
  </persName>
  <langKnowledge>
    <langKnown tag="de">German</langKnown>
    <langKnown tag="en">English</langKnown>
    <langKnown tag="it">Italian</langKnown>
    <langKnown tag="la">Latin</langKnown>
  </langKnowledge>
</person>
```

Ways of using <langKnowledge>

You can say:

```
<langKnowledge  tags="fr wo en">
  <p>Speaks fluent Wolof and French. Some knowledge of
    English.</p>
</langKnowledge>
```

or

```
<langKnowledge>
  <langKnown  level="fluent"  tag="wo">Wolof</langKnown>
  <langKnown  level="fluent"  tag="fr">French</langKnown>
  <langKnown  level="basic"  tag="en">English</langKnown>
</langKnowledge>
```

Sex

The <sex> element carries a *@value* attribute to give the ISO 5218:1977 values, i.e. 1 for male, 2 for female, 9 for non-applicable and 0 for unknown.

```
<sex value="2">female</sex>
```

A generic element for traits

The generic <trait> element has a *@type* attribute and can contain an optional <label> element, which can be used to provide a human-readable specification for the category of feature concerned, followed by a <desc> element; alternatively, the description of the feature is supplied within one or more <p> elements.

```
<trait  type="ethnicity">
  <label>Ethnicity</label>
  <desc>Ethnic Albanian.</desc>
</trait>
```

Personal States

The model.persStateLike class contains elements describing changeable characteristics of a person which have a definite duration, for example occupation, residence, or name. Members of this class comprise:

- <persName> (personal name) a proper noun or proper-noun phrase referring to 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

States Example

```
<person xml:id="fmendelssohn2">
  <persName>
    <forename>Felix</forename>
    <surname>Mendelssohn</surname>
  </persName>
  <education>He began taking piano lessons from his mother when he was six, and at seven was tutored by Marie Bigot in Paris. From 1817 he studied composition with Carl Friedrich Zelter in Berlin. Zelter introduced Mendelssohn to his friend and correspondent, the elderly Goethe. He later took lessons from the composer and piano virtuoso Ignaz Moscheles. Besides music, Mendelssohn's education included art, literature, languages, and philosophy.</education>
</person>
```

A generic element for states

The <state> element has the same content model as <trait>. The example given here describes the first living held by the Icelandic clergyman and poet Jón Oddsson Hjaltalín:

```
<state type="office" from="1777-04-07" to="1780-07-12">
  <p>Jón's first living, which he apparently
     accepted rather reluctantly, was at <name type="place">Háls í Hamarsfirði</name>,
<name type="place">Múlasýsla</name>, to which he
     was presented on 7 April 1777. He was ordained
     the following month and spent three years at
     Háls, but was never happy there, due largely
     to the general penury in which he was forced
     to live. In June of 1780 the bishop
     recommended that Jón should <q xml:lang="da">promoveres
          til andet bedre kald, end det hand hidindtil
          har havt</q>, and on 12
     July it was agreed that he should exchange
     livings with <name type="person" key="#ThorJon">sr. Þórður Jónsson</name> at
  <name type="place">Kálfafell á Síðu</name>,
  <name type="place">Skaftafellssýsla</name>. <bibl>ÞÍ,
          Stms I.15, p. 733.</bibl>
    <bibl>ÞÍ, Stms
          I.17, p. 102.</bibl>
  </p>
</state>
```

Assigning dates and date ranges

All states, and name elements like <persName> are in both the specific and the generic forms, are members of the *datable* attribute class, which means they can be limited in terms of time, as in the following example, where the person originally named David Jones has changed his name in 1966 to David Bowie:

```
<person xml:id="DB">
  <persName notAfter="1966">David Jones</persName>
  <persName notBefore="1966">David Bowie</persName>
</person>
```

Personal Events

The model.persEventLike class contains elements describing specific events in a person's history, for example birth, marriage, or appointment. These are not characteristics of an individual, but often cause an individual to gain such characteristics, or to enter a new state. Members of this class comprise:

- <birth> information about a person's birth, such as its date and place
- <death> (information about a person's death, such as its date and place)
- <event> (information about an event in a person's life)

Example

```
<person xml:id="rwagner">
  <persName>Richard Wagner</persName>
  <event type="marriage" when="1836-11-24">
    <desc>On 24 November 1836, Wagner married
        actress <persName>Christine Wilhelmine
      <forename full="abb">Minna</forename> Planer </persName>. </desc>
  </event>
  <event type="move">
    <desc>They moved to the city of <placeName>Riga</placeName>, then in the
    <bloc>Russian Empire</bloc>.</desc>
  </event>
</person>
```

Assigning certainty and responsibility

All the generic elements are also members of the *editLike* class, which, as its name implies, was originally intended to provide attributes 'describing the nature of an encoded scholarly intervention or interpretation of any kind' and which makes available the attributes *@cert*, to indicate the degree of certainty, *@resp*, the agency responsible, and *@evidence*, the nature of the evidence used. In this way it is possible, in the case of multiple and conflicting sources, to provide more than one view of what happened, as in the following example, where a different place and date of birth for an individual has been claimed by two scholars, one thought to be more reliable than the other:

```
<event type="birth" resp="#Smith2005" cert="high">
  <p>Born in <name type="place">Brixton</name> on 8 January 1947.</p>
</event>
<event type="birth" resp="#Watson1996" cert="low">
  <p>Born in <name type="place">Berkhamsted</name> on 9 January 1947.</p>
</event>
```

Assigning certainty and responsibility

The *@resp* attribute can also be used to distinguish between things which are explicitly stated in the source and those which, though not explicitly stated, can be inferred from the information available, as in the following example, from a collection of data assembled by Sebastian Rahtz on the people buried in the Protestant Cemetery in Rome based on the information on their tombstones:

```
<person xml:id="WHH">
  <persName>
    <forename>Winchcombe</forename>
    <forename>Henry</forename>
    <surname>Hartley</surname>
  </persName>
  <sex value="1" resp="#SPQR"/>
  <birth
      notBefore="1773-02-22"
      notAfter="1774-02-21"
      resp="#SPQR"
      cert="low"/>
  <death when="1847-02-21">21 February 1847</death>
  <residence>
    <placeName>Belvedere</placeName>, near <placeName>Bath</placeName>, <placeName>
      <region>Somersetshire</region>
    </placeName>
  </residence>
  <nationality resp="#SPQR">British</nationality>
  <occupation>Judge of the Admiralty Court at the Cape of Good
    Hope</occupation>
  <occupation>Fellow of Merton College, Oxford</occupation>
</person>
```

Personal Relationships

- <relationGrp> (relation group) provides information about relationships identified amongst people, places, and organizations
- <relation> (relationship) describes any kind of relationship or linkage amongst a specified group of participants

@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="jsbach">
  <persName>Johann Sebastian Bach</persName>
</person>
<person xml:id="cdbach">
  <persName>Catharina Dorothea Bach</persName>
</person>
<person xml:id="ghbach">
  <persName>Gottfried Heinrich Bach</persName>
</person>
<!--⋯.-->
<relationGrp type="children" subtype="first-marriage">
  <relation name="parent" active="#jsbach" passive="#cdbach"/>
<!--⋯.-->
</relationGrp>
<relationGrp type="children" subtype="second-marriage">
  <relation name="parent" active="#jsbach" passive="#ghbach"/>
<!--⋯.-->
</relationGrp>
```

Elements for events

The class of elements for events contains specific elements for <birth> and <death> and the generic <event> element; it has the same content model as <state> and <trait> and can be used to describe any event in the life of an individual. In the example the event is the wedding of Jane Burdon to the English writer, designer and socialist William Morris.

Example of <event>

```
<event type="marriage">
  <label>Marriage</label>
  <desc>
    <date when="1859-04-26">26 April 1859</date>
    <name type="person" ref="#WM">William Morris</name>
     and Jane Burden were married at <name type="place">St Michael's Church, Ship
          Street, Oxford</name> on <date when="1859-04-26">26
          April 1859</date>. The wedding was conducted by
     Morris's friend <name type="person" ref="#RWD">R. W. Dixon</name>
     with <name type="person" ref="#CF">Charles Faulkner</name> as the
     best man. The bride was given away by her father,
  <name type="person" ref="#RB">Robert Burden</name>. According to the
     account that <name type="person" ref="#EBJ">Burne-Jones</name> gave
  <name type="person" ref="#JWM">Mackail</name>
    <q>M. said to Dixon beforehand <q>Mind you don't
              call her Mary</q> but he did</q>. The entry in the
     Register reads: <q>William Morris, 25, Bachelor
          Gentleman, 13 George Street, son of William Morris
          decd. Gentleman. Jane Burden, minor, spinster, 65
          Holywell Street, d. of Robert Burden, Groom.</q> The
     witnesses were Jane's parents and Faulkner. None of
     Morris's family attended the ceremony. Morris
     presented Jane with a plain gold ring bearing the
     London hallmark for 1858. She gave her husband a
     double-handled antique silver cup.</desc>
  <bibl>J. W. Mackail, <title>The Life of William
          Morris</title>,
     1899.</bibl>
</event>
```

Use of @key attribute

In the example the keys on the various <persName> elements point to the <person> elements for the other people named. The <relation> element can then be used to link them in a more meaningful way:

```
<relation  name="spouse"   mutual="#WM #JBM"/>
<relation  name="friend"   mutual="#WM #RWD"/>
<relation  name="parent"   active="#RB"   passive="#JBM"/>
```

Nyms

The elements <listNym> and <nym> have been introduced to define canonical name or name-part of any kind

- <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="lipsk">
  <form>Lipsk</form>
  <etym>From <lang>Slavic</lang>; it means <gloss>settlement where the lime trees
        stand</gloss>. </etym>
</nym>
```

Dates and Periods

The support for dates in TEI P5 has concentrated on enabling greater use of international standards (W3C and ISO)

- <date> contains a date in any format
- <time> contains a phrase defining a time of day in any format

Example

```
<place xml:id="leipzig-univ">
  <placeName>University of Leipzig</placeName>
  <event type="foundation">
    <desc>The university was founded on <date when="1409-12-02">December 2, 1409</date>.
    </desc>
  </event>
</place>
```

W3C Date Formats

Thanks to the mapping to W3C (att.datable.w3c) and ISO date formats, automatic processing and validation of expression of dates and times are now allowed

att.datable.w3c provides attributes for normalization of elements that contain datable events using the W3C datatypes

@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 W3C standard form for dates is YYYY-MM-DD.

Example

```
<place xml:id="leipzig-univ2">
  <placeName>University of Leipzig</placeName>
<!--···-->
  <event type="opening" notBefore="1409-09-09">
    <desc>The <foreign xml:lang="la">Alma mater Lipsiensis</foreign> opened in 1409,
        after it had been officially endorsed by Pope Alexander V in his Bull of
        Acknowledgment on (September 9 of that year).</desc>
  </event>
</place>
```

ISO Date Fomats

For some uses the subset of ISO 8601 which is used by the W3C might not be enough, so the TEI provides an optional att.datable.iso class to give the following attributes if needed:

@when-iso	the value of a date or time in a standard form
@notBefore-iso	the earliest possible date for the event
@notAfter-iso	the latest possible date for the event
@from-iso	the starting point of the period
@to-iso	the ending point of the period
@dur-iso	the length of this element in time

The ISO standard, for example, allows specifying dates and durations with a precision by omitting some digits to the left, while the W3C datatypes require in most cases conformance to a stricter precision.

Example

```
<p>He arrived <time   when="12:00:00">around noon</time>. He arrived
<time   when-iso="12">around noon</time>.</p>
```

Time Periods and Relative Chronology

Time periods and relative chronology can also be defined under <encodingDesc> and <classDecl>.

```
<taxonomy xml:id="periods">
  <category xml:id="hellenistic">
    <catDesc>
      <ref
          target="http://www.wikipedia.com/wiki/Hellenistic"> Hellenistic</ref>. Commonly treated
as <date notBefore="-0323" notAfter="-0031"/>. </catDesc>
  </category>
<!--···-->
</taxonomy>
<!--···-->
<p> The city was built near a marble quarry which was extensively exploited in the
<date period="#hellenistic">Hellenistic</date> and <date period="#roman"> Roman</date>
periods.</p>
```