

# NAMES, DATES, PEOPLE, AND PLACES

---

Martina Scholger

Centre for Information Modelling - Austrian Centre for Digital Humanities  
University of Graz

Thanks to Lou Burnard, James Cummings, Sebastian Rahtz,  
and the rest of the TEI community providing workshop material.



# Names, people and places

Names and other references to objects appear in most texts. How they appear can differ significantly...



# Names, people and places

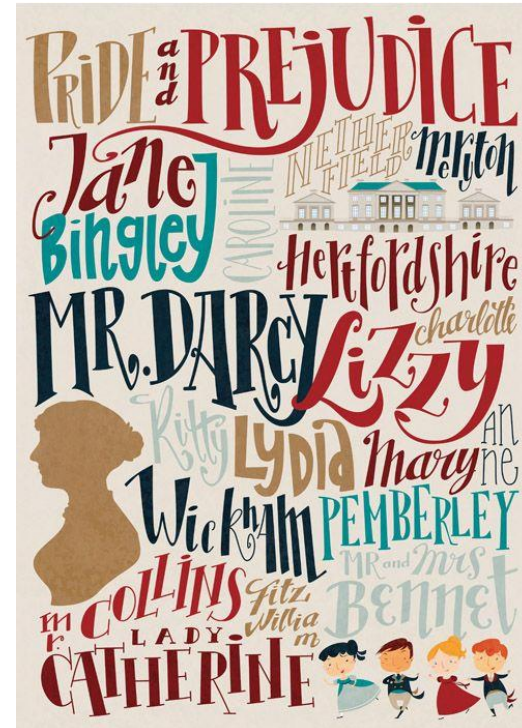
Encoding of persons, places, and referring strings:

*"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.



# Names, people and places

Encoding of **persons**, places, and referring strings:

*"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.

# Names, people and places

Encoding of **persons**, **places**, and referring strings:

*"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.

# Names, people and places

Encoding of **persons**, **places**, and **referring strings**:

*"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.

# References are not the entities which they refer to

A person, place or organisation might be known by many different names or might be referred to by some other description.

*"Why, my dear, you must know, Mrs. Long says that **Netherfield Park** 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?"*

*"**Bingely**."*

# Names in TEI

The TEI provides several ways of marking up names and nominal expressions.

Already known from the „core“ module:

- `<rs>` contains a string which refers to a person or place, e.g. ‚his lady‘, ‚she‘, etc.
- `<name>` contains a proper noun or noun phrase, e.g. ‚William Shakespeare‘, ‚Stratford-upon-Avon‘
- `@type` distinguish persons and places



# Encoding names

- Encoding of names using the `<name>` element

"My dear `<name>`Mr. Bennet`</name>`", said his lady to him one day, "have you heard that `<name>`Netherfield Park`</name>` is let at last?" `<name>`Mr. Bennet`</name>` replied that he had not.

"But it is," returned she; "for `<name>`Mrs. Long`</name>` has just been here, and she told me all about it."  
`<name>`Mr. Bennet`</name>` made no answer.

# Distinguish different types of names

- Add the `@type` attribute to be more precise

"My dear `<name type="person">Mr. Bennet</name>`", said his lady to him one day, "have you heard that `<name type="place">Netherfield Park</name>` is let at last?" `<name type="person">Mr. Bennet</name>` replied that he had not.

"But it is," returned she; "for `<name type="person">Mrs. Long</name>` has just been here, and she told me all about it."

`<name type="person">Mr. Bennet</name>` made no answer.

# 13 Names, Dates, People, and Places

The **namesdates** module allows to encode names and other phrases *descriptive of persons, places, or organizations* in more detail and supplies a detailed *sub-structure*.

- **<persName>**, **<placeName>**, **<orgName>**:  
,spice up‘ the **<name>** element from the core
- **<surname>**, **<forename>**, **<geogName>**, **<geogFeat>** etc.:  
components of nominal expressions

# Personal names and place names

- `<persName>` and `<placeName>`

"My dear `<persName>`Mr. Bennet`</persName>`", said `<rs>`his lady`</rs>` to `<rs>`him`</rs>` one day, "have you heard that `<placeName>`Netherfield Park`</placeName>` is let at last?" `<persName>`Mr. Bennet`</persName>` replied that `<rs>`he`</rs>` had not.

"But `<rs>`it`</rs>` is," returned `<rs>`she`</rs>`; "for `<persName>`Mrs. Long`</persName>` has just been here, and `<rs>`she`</rs>` told me all about it."

`<persName>`Mr. Bennet`</persName>` made no answer.

# Components of personal names

Components of `<persName>`

- `<forename>` contains a forename or given name
- `<surname>` contains a family name

```
<persName>  
  <forename>Wilfred</forename>  
  <forename>Edward</forename>  
  <forename>Salter</forename>  
  <surname>Owen</surname>  
</persName>
```

# More components of personal names

- `<roleName>` a name component indicating the role or position in society (e.g. Emperor, President, General)
- `<genName>` generational or dynastic information (e.g. the Elder, Jr., II)
- `<addName>` an additional name such as a nickname, or alias (e.g. the Red, the Great)
- `<nameLink>` connecting phrase of name parts (e.g. van, of, de)

```
<persName>  
  <roleName>Mme</roleName>  
  <nameLink>de la</nameLink>  
  <surname>Rochefoucault</surname>  
</persName>
```

# Reference theory

- Reference is a fundamental semiotic concept
  - We use natural language to talk about the real world using different forms for specific objects: *signifier* and signified.
  - Proper names and technical terms are canonical examples of this real world.
  - Marcel Duchamp = Rrose Selavy = R. Mutt = Duchamp, M.
  - All of these are different designations of the same concept: Marcel Duchamp
  - When we translate between natural languages, often the proper names don't change, or are equivalent (e.g. William Shakespeare – de = en = fr = es).



# Names and their referenced entities

Consequently, the TEI distinguishes between the signs in the text (occurrence of a name) and the thing to which they refer (the real world object).

- `<person>` corresponds with `<persName>`
- `<place>` corresponds with `<placeName>`
- `<org>` corresponds with `<orgName>`



# Storing information about named entities

Detailed information about a person is usually stored in the TEI Header

- `<person>` groups together information about a person
- `<personGrp>` a group of individuals regarded as a single entity (e.g. „the audience“)
- `<listPerson>` contains a series of `<person>` and `<personGrp>` elements and is supplied within the `<particDesc>` element in the `<profileDesc>` element of the TEI header.

# Named entities: places

```
<profileDesc>
  <particDesc>
    <listPerson>
      <person xml:id="WS1564">
        <persName xml:lang="en">
          <forename>William</forename>
          <surname>Shakespeare</surname>
        </persName>
        <persName xml:lang="ru">
          <forename>Уільям</forename>
          <surname>Шэкспір</surname>
        </persName>
        <birth when="1564">1564
          <placeName ref="#SuA">Stratford-upon-Avon</placeName>
        </birth>
        <death when="1616">1616
          <placeName ref="#SuA">Stratford-upon-Avon</placeName>
        </death>
      </person>
    </listPerson>
  </particDesc>
</profileDesc>
```

# Why?

- To facilitate a more detailed and explicit encoding of source documents.
- To support the encoding of „data-centric“ documents (e.g. authority files, biographical or geographical dictionaries and gazetteers).
- To represent implicit data in a uniform way.

# Representing the association

- **@ref** locating a full definition for the entity being named by means of one or more URIs.

```

<profileDesc>
  <particDesc>
    <listPerson>
      <person xml:id="WS1564">
        <persName xml:lang="en">
          <forename>William</forename>
          <surname>Shakespeare</surname>
        </persName>
      </person>
    </listPerson>
  </particDesc>
</profileDesc>

```

```

<text>
  <body>
    <p><persName ref="#WS1564">W.S.</persName>
      was born in 1564</p>
  </body>
</text>

```

# Pointing mechanisms

The `@ref` attribute can take any kind of pointer:

- Entity defined within the same XML document

This year marks the 400th anniversary of `<name ref="#WS1">Shakespeare's</name>` death

- In some other place, referred to by means of an URI

This year marks the 400th anniversary of `<name ref="http://www.example.com/persons.xml#WS1">Shakespeare's</name>` death

- Reference an existing authority file

This year marks the 400th anniversary of `<name ref="http://viaf.org/viaf/96994048">Shakespeare's</name>` death

# Nyms

The elements `<listNym>` and `<nym>` contain the canonical form of a name or a name component.

- `<nym>` the definition for a canonical name
  - may nest a number of `<nym>` elements
  - can contain elements of the `model.entryPart` class (e.g. `<form>`, `<orth>`, `<etym>`)
- `<listNym>` a list of canonical names
- `@nymRef` has been added to the attribute class `att.naming` to refer to canonical names.

# Nyms

```
<forename nymRef="#N45">William</forename>
```

```
<listNym>
  <nym xml:id="W45">
    <form xml:lang="la">Gulielmus</form>
    <nym xml:id="W450">
      <form xml:lang="en">William</form>
      <nym xml:id="W4501">
        <form>Willy</form>
      </nym>
      <nym xml:id="W4502">
        <form>Billy</form>
      </nym>
    </nym>
    <nym xml:id="W455">
      <form xml:lang="de">Wilhelm</form>
    </nym>
    <nym xml:id="W453">
      <form xml:lang="es">Guillermo</form>
    </nym>
  </nym>
</listNym>
```

# Places





# Components of place names

- `<placeName>` contains an absolute or relative place name
- `<geogName>` for representing topographical features (e.g. mountains)
- `<geogFeat>` a term for some particular kind of geographical features e.g. „Mount“, „Lake“

```
<placeName>  
  <geogName>Humbleton Hill</geogName>  
  <offset>near</offset>  
  <placeName>Wooler</placeName>  
</placeName>
```

# Geo-political place names



**bloc**

**country**

**region**

**settlement**

**district**

# Components of place names

- <district> subdivision of a settlement (e.g. parish, ward)
- <settlement> name of a settlement (e.g. city, town, village)
- <region> administrative unite (e.g. state, province, county)
- <country> geo-political unite (e.g. nation, country)
- <bloc> geo-political unit containing two or more nation states (e.g. Southeast Asia)

# Location

A `<location>` element can contain

- geographical coordinates
  - place name components, or
  - an address
- 
- `<geo>` supplies the coordinates of a `<location>`

```
<place xml:id="SUA">  
  <placeName>Stratford-upon-Avon</placeName>  
  <settlement>Warwickshire</settlement>  
  <country>United Kingdom</country>  
  <location>  
    <geo>52.1925 -1.709722</geo>  
  </location>  
</place>
```

# Store information about named entities

Detailed information about a place is usually stored in the TEI Header, within the `<place>` element:

- `<place>` groups together information about a place

```

<profileDesc>
  <settingDesc>
    <listPlace>
      <place xml:id="SUA">
        ....
      </place>
      <place xml:id="SUA">
        ....
      </place>
    </listPlace>
  </settingDesc>
</profileDesc>

```

# Named entities: places

```

<profileDesc>
  <settingDesc>
    <listPlace>
      <place xml:id="SUA">
        <placeName>Stratford-upon-Avon</placeName>
        <settlement>Warwickshire</settlement>
        <region>England</region>
        <country>United Kingdom</country>
        <location>
          <geo>52.1925, -1.709722</geo>
        </location>
      </place>
    </listPlace>
  </settingDesc>
</profileDesc>

```

```

<text>
  <body>
    <p>W.S. was born in <placeName
ref="#SUA">Stratford-upon-Avon</placeName></p>
  </body>
</text>

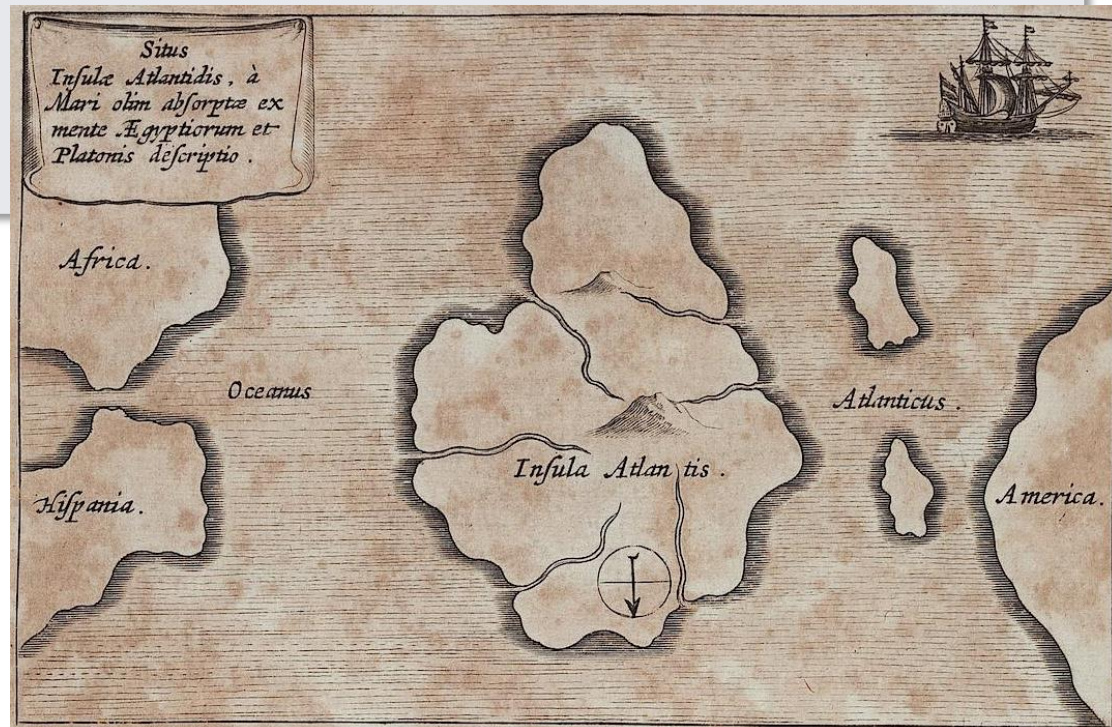
```

# Places can be fictional...

```

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

```



# Nested places

```
<place>  
  <country>United Kingdom</country>  
  <place>  
    <settlement>London</settlement>  
  </place>  
  <place>  
    <settlement>Oxford</settlement>  
  </place>  
</place>
```



# Organizational names

Organizations are named collections of people regarded as a single unit: university institutions, broadcasting corporations, businesses, racial or ethnic groupings, political factions, etc.

- `<orgName>` contains organizational names
- `@ref` for pointing to an `<org>` element

`<p>The <orgName>Bodleian Library</orgName> provides a digital facsimile of the First Folio of Shakespeare's plays from Arch. G c.7.</p>`

# Organisational data

- `<org>` provides information about an organization (a business, a tribe, a party)
- `<listOrg>` contains a list of `<org>` elements
- `<listOrg>` contains a series of `<org>` elements and is supplied within the `<particDesc>` element in the `<profileDesc>` element of the TEI header.

```
<particDesc>
  <listOrg>
    <org xml:id="BL">
      <orgName>Bodleian Library</orgName>
      <desc>The mission of the Bodleian Libraries is to provide an
        excellent service to support the learning, teaching and research
        objectives of the University of Oxford</desc>
    </org>
  </listOrg>
</particDesc>
```

# States, traits

More information for persons, places and organizations:

- `<state>` status or quality in relation to a specific time or date range

```
<state notBefore="1590-01-01">
  <label>Partner</label>
  <desc>A By the early 1590s William Shakespeare was a managing
partner in the Lord Chamberlain's Men.</desc>
</state>
```

- `<trait>` status or quality independent of the volition or action of the holder

```
<trait type="physical">
  <label>eye color</label>
  <desc>blue</desc>
</trait>
```

# Traits of a person

Some typical traits of a person:

- `<faith>` faith, religion, or belief of a person
- `<langKnowledge>` linguistic knowledge of a person
- `<nationality>` a person's present or past nationality
- `<sex>` a person's gender
- `<age>` a person's age
- `<socecStatus>` the socio-economic status

```
<person xml:id="F-1h4-hn4">  
  <persName type="standard">Henry IV, King of  
England</persName>  
  <socecStatus>Noble</socecStatus>  
  <sex value="m"/>  
</person>
```

# States of a person

A state describes some status or quality in relation to a specific time or date range:

- <occupation> a person's trade, profession, or occupation
- <residence> a person's place of residence
- <affiliation> a person's affiliation with some organization
- <education> a person's education
- <floruit> a person's period of activity

# Traits of a place

- <climate> information about the physical climate of a place
- <location> describes where a place is
- <population> information about the population of a place
- <terrain> information about the physical terrain of a place

# Events

For persons two elements are pre-defined to describe specific events:

- `<birth>` a person's birth date
- `<death>` a person's death date
- For all other events, the generic `<events>` element in combination with the `@type` attribute is used.
- Events in the real world may lead to a change in a state or trait (e.g. "marriage" for a person, "war" for a place, ...)

# Events

```
<person xml:id="WS">
  <persName>William Shakespeare</persName>
  <birth when="1564-04-23">April 23, 1564</birth>
  <event when="1582" type="marriage">
    <p>In late 1582, William Shakespeare married Anne
Hathaway.</p>
  </event>
  <event when="1585" type="children">
    <p>In early 1585, the couple had twins, Judith and
Hamnet, completing the family.</p>
  </event>
</person>
```



# Personal relationships

- The `<relation>` element describes any kind of relationship amongst entities.
- We distinguish mutual (e.g. sibling) and non-mutual or directed (e.g. parent-of) relationships

The following attributes are available:

- `@name` supplies a name for the kind of relationship
- `@active` the ,active‘ participants in a non-mutual relationship
- `@passive` identifies the ,passive‘ participants in a non-mutual relationship
- `@mutual` a list of participants in an equal relationship

# Personal relationships

```
<listPerson>
  <person xml:id="pp1">
    <persName>William Shakespeare</persName>
  </person>
  <person xml:id="pp2">
    <persName>Susanna Hall</persName>
  </person>
  <person xml:id="pp3">
    <persName>Judith Quiney</persName>
  </person>

  <relation name="parent" active="#pp1" passive="#pp2
#pp3"/>

</listPerson>
```

# Times & Dates

*Better three hours too soon,  
than one minute too late*

*William Shakespeare.*



# Dates and times

Known elements from the core module:

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

The *namesdates module* provides encoding mechanisms for relative dates and times:

- `<offset>` contains a relative temporal or spatial expression

Just `<offset>before</offset>` his death in  
`<date>1616</date>...`

# Dating attributes

Attributes (att.dateable.w3c) for normalizing elements that contain dates, times, etc. in standard form (e.g. yyyy-mm-dd):

- **@when** value of the date or time
- **@notBefore** the earliest possible date
- **@notAfter** the latest possible date
- **@from** indicates the starting point of a period
- **@to** indicates the ending point of a period

```
<p>  
  <date when="2016-06-24">June 24</date>,  
  a day that changed Europe  
</p>
```



# Time attributes

Absolute temporal expressions denoting times which are given in terms of seconds, minutes, hours, or of well-defined events (e.g. 'noon', 'sunset') may be represented using the `<time>` element:

The train leaves for Boston at

`<time when="13:45:00">a quarter to two</time>` and will arrive at `<time type="occasion">sunset</time>`.

Thank you for your attention!

Any questions?