CENTER FOR TEXTUAL STUDIES AND
DIGITAL HUMANITIES

DIGH 400 - Introduction to Digital Humanities Research

Fall Semester 2013

Week 11

Today's Class

- Presentation
- Conceptual Project and 2nd Presentation
- Text Encoding Initiative (TEI) continued
- XSLT

Conceptual Project

conceptual project design (40%)

- choose your own preferred material, text, work (you'll need to be able to justify your selection)

- helps visualise project management and development

- beneficial for future development and preparation of grant proposals

- does NOT require actual project development, simply conceptual planning and design

NEH guidelines overview

Project Presentation 2

26th November 2013 - ~10 mins per paper & questions from the class

- conference style short paper
- describe the work or material you have selected for the conceptual project

For example:

- briefly outline and describe the material
- describe the project and proposed final outcomes
- any possible research output
- any similar works, publications, or projects
- how the project will contribute to DH research and development...

Text Encoding Initiative

More about Textual Phenomena - Analytical Features

Notes and Annotations

- <note> can be used to record a textual annotation
- @type attribute used to differentiate notes
- @resp attribute used to assign responsibility
- position of the note can be referenced using the @place attribute

<note n="1" place="foot" type="editorial" resp="NJH">a new note...</note>

Text Encoding Initiative

More about Textual Phenomena - Analytical Features

Index Entries

- pre-existing in <front> or <back> using <list> inside
- new index using <term> inside <index> at the location of the index item

<index>
<term>new index term...</term>
</index>

Text Encoding Initiative

More about Textual Phenomena - Analytical Features

Errors

- indicated using <sic>
- corrected using <corr>
- combine <sic> and <corr> within <choice>
- use @cert and @resp attributes to encode degree of certainty and editor responsible

<corr cert="high" resp="#NJH">correction...</corr>

Text Encoding Initiative

More about Textual Phenomena - Analytical Features

Regularisation and Normalisation

- <reg> element for regularisation
- <orig> for original, non-normalised form
- use <reg> in isolation or combined with <orig> within <choice>

```
<choice>
<orig>thou</orig>
<reg resp="#NJH">you</reg>
</choice>
```

Text Encoding Initiative

More about Textual Phenomena - Analytical Features

Additions, Deletions & Omissions

- <gap> element used for omission, both material and editorial
- @reason attribute used to indicate reason for omission
- @extent and @unit attributes can be used to record extent of omission
- editorial omissions should be recorded using <editorialDecl> inside <editorialDesc> in <teiHeader>
- <gap> may be empty or include a <desc> of the material omitted
- <add> and <del> may also be used for words and phrases
- @rend attribute may also be used with <add> and <del>
- <addSpan/> and <delSpan/> for longer passages
- <subst> to contain <add> and <del> with causal relationship
- <unclear> with @reason attribute for difficult to read deletions in the text

Text Encoding Initiative

Non-Textual Phenomena

- graphics such as illustrations, diagrams, drawings, artwork...
- anchor in a text using <graphic/> and optional @url attribute

<graphic url="http://www.somewhere.com/image.jpg"/>

- use <figure> element as parent to create a full listing for a graphic

Text Encoding Initiative

Global Attributes

There are currently 6 global attributes:

- @n
- @rend
- @rendition
- @xml:base
- @xml:id
- @xml:lang

Text Encoding Initiative

TEI P5 Guidelines

XSL

Overview

- consists of three parts
    - XSLT: transforms XML documents
    - XPath: navigates XML documents
    - XSL-FO: formats XML documents

XSLT

- XSL Transformations
- considered most important part of XSL
- transforms XML into another XML document, or another type of document
- transforms each XML elements into an (X)HTML element for browser viewing
- uses XPath for XML navigation
- W3C recommendation
- manipulate and transform an XML document
    - add or remove elements
    - rearrange and sort elements...

XSLT - Browser Support (W3 list)

Mozilla Firefox
Firefox supports XML, XSLT, and XPath from version 3.

Internet Explorer
Internet Explorer supports XML, XSLT, and XPath from version 6.
Internet Explorer 5 is NOT compatible with the official W3C XSL Recommendation.

Google Chrome
Chrome supports XML, XSLT, and XPath from version 1.

Opera
Opera supports XML, XSLT, and XPath from version 9. Opera 8 supports only XML + CSS.

Apple Safari
Safari supports XML and XSLT from version 3.

XSLT - Process Overview

- select the XML document you want to transform into XHTML
- create an XSL style sheet with a transformation template
- link the XSL style sheet to the XML document
- XSLT compliant browser will transform XML into XHTML

Agatha Christie Example

Initial Basic XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalogue>
  <book>
    <title>Evil Under the Sun</title>
    <author>Agatha Christie</author>
    <country>UK</country>
    <publisher>Collins Crime Club</publisher>
    <price>7</price>
    <year>1941</year>
  </book>
.
.
</catalogue>
```

XSLT - <xsl:template>

- one or more sets of rules called templates
- a template contains rules for a specified matched node
- <xsl:template> element is used to build templates
- 'match' attribute associates a template with an XML element

<xsl:template match="/">

<?xml-stylesheet type="text/xsl" href="christie-full.xsl"?>

DIGH 400 - Introduction to Digital Humanities Research

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>Collection</h2>
  <table>
    <tr>
      <th>Title</th>
      <th>Author</th>
    </tr>
    <tr>
      <td>.</td>
      <td>.</td>
    </tr>
  </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

Example

XSLT - <xsl:value-of>

- used to extract the value of an XML element and add to transformation output

<xsl:value-of select="catalogue/book/title"/>
<xsl:value-of select="catalogue/book/author"/>

- 'select' attribute's value contains an XPath expression
- XPath expression navigates to the given position in the XML

- the value of an attribute can be found using an expression such as

<xsl:value-of select="catalogue/book/title/@id"/> or
<xsl:value-of select="catalogue/book/title/attribute::id"/>

DIGH 400 - Introduction to Digital Humanities Research

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>Collection</h2>
  <table>
    <tr>
      <th>Title</th>
      <th>Author</th>
    </tr>
    <tr>
      <td><xsl:value-of select="catalogue/book/title"/></td>
      <td><xsl:value-of select="catalogue/book/author"/></td>
    </tr>
  </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

Example

XSLT - <xsl:for-each>

- used to select every element from a specified set

```
<xsl:for-each select="catalogue/book">
<xsl:value-of select="title"/>
<xsl:value-of select="author"/>
</xsl:for-each>
```

DIGH 400 - Introduction to Digital Humanities Research

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>Collection</h2>
  <table>
    <tr>
      <th>Title</th>
      <th>Author</th>
    </tr>
    <xsl:for-each select="catalogue/book">
    <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="author"/></td>
    </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

Example

XSLT - <xsl:for-each> filtering

<xsl:for-each select="catalogue/book[author='Agatha Christie']">

- filter XML output using additional operators such as

= (equal)
!= (not equal)
< (less than)
> (greater than)

<xsl:for-each select="catalogue/book[author='Agatha Christie' and year='1941']"
>

DIGH 400 - Introduction to Digital Humanities Research

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>Collection</h2>
  <table>
    <tr>
      <th>Title</th>
      <th>Author</th>
    </tr>
    <xsl:for-each select="catalogue/book[year='1937']">
    <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="author"/></td>
    </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

Example 1 - Year =1937 Example 2

DIGH 400 - Introduction to Digital Humanities Research

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>Collection</h2>
  <table>
    <tr>
      <th>Title</th>
      <th>Author</th>
      <th>Year</th>
    </tr>
    <xsl:for-each select="catalogue/book[year!='1937' and year&lt;1942]">
    <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="author"/></td>
      <td><xsl:value-of select="year"/></td>
    </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

XSLT - <xsl:sort>

- sort returned data from XML

```
<xsl:sort select="title"/>
```

- we can then add the sort filter to the for-each option

```
<xsl:for-each select="catalogue/book">
<xsl:sort select="title"/>
```

- default sort order is ascending (NB: for numbers this will always be the first number regardless of natural value - eg: 1, 10, 100, 2, 3, 31, 4...)

DIGH 400 - Introduction to Digital Humanities Research

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>Collection</h2>
  <table>
    <tr>
      <th>Title</th>
      <th>Author</th>
    </tr>
    <xsl:for-each select="catalogue/book">
    <xsl:sort select="title"/>
    <tr>
      <td><xsl:value-of select="title"/></td>
      <td><xsl:value-of select="author"/></td>
    </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>

</xsl:stylesheet>
```

Example