2012-11-17

Creating Digital Editions: An Introduction to the Text Encoding Initiative (TEI)

Welzenbach, Rebecca; Hawkins, Kevin; Schaffner, Paul

http://hdl.handle.net/2027.42/109403
Introduction to XML/TEI
Why encode a text in TEI?

You need...

- a system that allows annotations of the text in order to label the various components

- "a simple data format that balances the needs of people to read/write data with the needs of machines to read/write data."

- a format for interchanging bits of structured data

- a standard way of annotating these documents.

- ...to make explicit certain features of text in order to aid the processing of that text by computer programs.
What features?

Almost anything! Have to make choices based on your project's goals and resources:

- Making the information conveyed by a page layout readable to a machine? (capturing paragraphs as <p>, tagging decorative initial capitals)

- Adding scholarly interpretation that makes the text richer? (adding notes, or glosses, adding explanation for why/how various phenomena occur, associating a pronoun with the name of the person being referenced)

- Standardizing bits of the text so they can be processed more easily? (expanding abbreviations, associating dates and place names with standard formats)
To what end?

The goal is always to make the text readable to a machine—so what you choose to encode depends on what you want an application to be able to read.

Think about what aspects of the text are not susceptible to machine processing in their current format, and what the benefits would be of addressing that (dates, names, places)

Can be tricky: if you're encoding for unknown future uses.

Let's look at how some real life websites make use of the various kinds of encoding options we talked about.
Encoding text structure
- Enables sensible rendering of the text online
- Enables searching restricted to specific regions of the text
Encoding text structure

- Enables sensible rendering of the text online
- Enables searching restricted to specific regions of the text
Normalizing non-standard forms

- Enables thorough searching and sorting
- Allows simultaneous fidelity to the source and ease of processing
- Enables creation of visualizations like maps and timelines

1. James the 2nd by ye Grace of God of England, Scotl: France & Ireland King, Defender of ye faith &c. To ye most Reverend Father in God our right trusty & right well-beloved Counsellor Wm Ld Archbishop of Cant: Primate of all Engl: & Metropolitan & to our right Trusty & right well-beloved Counsellor George Ld Jeffreys Ld. Chf. of Engl: & to our right Trusty & right well-beloved Cosin & Counsellor Laurence E. of Rochester, Ld high Treasurer of Engl: & to our right Trusty & right well-beloved Cosin & Counsellor Rob E. of Sunderland, Prsident of our Counsell & our principal Secretary of State, Nathaniel Ld. BP of Durham, & to ye right Reverend Father in God our Trusty & well-beloved Counsellor. Sr Edw: Herbert Kt. Cheif Justice of the Pleas before us to be holden assign'd Greeting.

1. James the 2nd by the Grace of God of England, Scotland France & Ireland King, Defender of the faith &c. To the most Reverend Father in God our right trusty & right well-beloved Councillor William Lord Archbishop of Canterbury Primate of all England & Metropolitan & to our right Trusty & right well-beloved Councillor George Lord Jeffreys Lord Chancellor of England & to our right Trusty & right well-beloved Cosin & Councillor Laurence Earl of Rochester, Lord high Treasurer of England & to our right Trusty & right well-beloved Cosin & Councillor Robert Earl of Sunderland, President of our Councill & our principal Secretary of State, Nathaniel Lord Bishop of Durham, & to the right Reverend Father in God our Trusty & well-beloved Councillor Sir Edward Herbert Knight Cheif Justice of the Pleas before us to be holden assign'd Greeting.
Adding interpretation/glossing

- Enables references between text and external factors such as people or known events
- Richly encoded file becomes the basis of your edition
XML in brief (1)

Open, non-proprietary standard

Stored in plain text but usually thought of as contrasting with it

Marks beginning and ends of spans of text using tags:
<sentence>This is a sentence.</sentence>
XML in brief (2)

Spans of text must nest properly:

Wrong:
<sentence>Overlap is <emphasis>not allowed!</sentence></emphasis>

Right:
<sentence>Overlap is <emphasis>not allowed!</emphasis></sentence>
Elements (tags), attributes, values, content

<sentence type="declarative">This is a sentence.</sentence>

<sentence type="interrogative">Is this a sentence?</sentence>
Elements (tags), attributes, values, content

Elements may have one attribute, many attributes, or none, but each attribute on any given element must be unique.

Valid: <sentence type=“declarative”>This is a sentence.</sentence>

Valid: <sentence type=“interrogative” xml:lang=“en”>Is this is a sentence?</sentence>

Valid: <sentence>This is a sentence.</sentence>

Invalid: <sentence type=“declarative” type=“true”>This is a sentence.</sentence>
XML as a tree

- We use family tree terms: parent, child, sibling, ancestor, and descendant.
- *Remember, everything must nest properly!*

```xml
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <paragraph>
    <sentence>It was a dark and stormy night.</sentence>
    <sentence>The wind howled and leaves blew off the trees.</sentence>
    <sentence>A dog barked in the distance.</sentence>
  </paragraph>
  <paragraph>
    <sentence>Inside, the flames in the fireplace struggled to resist the draft through the window.</sentence>
    <sentence>Jane struggled to stay warm under a blanket.</sentence>
  </paragraph>
</document>
```
Structure, not appearance

Most people use XML to describe the structure of a document rather than its appearance. Information about how to render various components of the document is usually stored separately, in a stylesheet.
Questions?