Textencoding with XML

Gerlinde Schneider

CLARIAH-AT
TEI Introductory School
10.-13. September 2019

Venue:
Centre for Information Modelling (ZIM)
Elisabethstraße 59/III, SR 81.31
Programm

Tuesday 10.9.
9:00-10:30 *Digital Scholarly Editing*
Roman Bleier, Martina Scholger

11:00-12:30 *Textencoding with XML*
Hans Clausen, Gerlinde Schneider

14:00-15:30 *TEI – General Introduction*
Hans Clausen, Gerlinde Schneider

16:00-17:30 *Practice: XML/TEI*
Hans Clausen, Gerlinde Schneider
Schedule

★ What is XML?
★ Why Text Encoding?
★ XML Basics
★ XML Syntax and Rules
★ Using the Oxygen XML editor
★ A beginners exercise

Goal: To understand
• how XML works and the relationship between XML and TEI
What is XML, actually?

XML stands for eXtensible Markup Language

A generic standard for the description and exchange text documents / textual data

W3C Standard

- 1.1 (Second Edition) - currently in use (4th edition of version 1.0) - 2006; 12 years ago

https://www.w3.org/TR/2006/REC-xml11-20060816/
Why XML

- System and platform independent
- Human and machine-readable
- Low-threshold
- Supported by a wide range of software
- International user and developer community
- Encompasses a whole range of accompanying standards

... separates structure and presentation

XML is extensible:
No predefined structure or names of elements and attributes,
Easily adaptable to the needs of specific domains and use cases
Model structured datasets

- Often used in software development and Information Sciences
- Data exchange and storage

E.g. Configuration files, metadata records ...

Source: British Library
Model narrative text

Very important for text-based humanities

Mixed content: Elements can contain strings without markup + other elements

```
<text>
  <body>
    <p>The American poet, critic and publisher <persName>T S Eliot</persName> was born into a comfortable and historically distinguished family in <placeName>St. Louis</placeName>, <placeName>Missouri</placeName> in <date>1888</date>. He studied at Smith Academy and then Harvard, where he undertook an eclectic range of courses before settling on a BA in what would now be called Comparative Literature and an MA in English Literature.</p>
    <p>He spent a year studying at the Sorbonne in Paris, and returned to Harvard to work on the philosophy of consciousness. This can be seen as influential in his earlier poetry, much of which is concerned with fractured perceptions and mental illness.</p>
  </body>
</text>
```
Text encoding

There is much more information in a text than can be expressed by character encoding.

Implicit content information or text structure are made explicit (machine-readable) by markup.

Different interpretations of a text and different readings, can also be explicated with markup.
Basic syntax

The basic unit is the XML **Element**

- An element is data, surrounded by a tag
- Elements must have a starting and a closing tag
- Elements can contain other elements, text or both

<element>Content (Element value)</element>

Start tag

End tag
Elements

- Empty elements don’t have content and are represented by a special tag

<element />
Attributes

Attributes give additional information to an element

- Assigned to the start tag of an element
- Name/Value pair
- Elements can take an unlimited number of attributes,
- but only one with one name
- Values must be quoted

<element attribute="value">Content</element>
Attribute or element

<w type="adjective" lemma="specific">specific</w>
Rules for XML names

Apply to element and attribute names

- Names can contain any alphanumeric characters, hyphens, dots, or underscores.
- Names must begin with an alphabetical character, underscore or colon.
- Names must not start with a number.
- Names are case sensitive: they distinguish between uppercase and lowercase:
  \(<\text{title}>\neq<\text{Title}>\)
- Names can be of any length.
- The usage of \(<\) & ' and " is not allowed.
Entity references

For reserved characters

&lt;  <  less than
&gt;  >  greater than
&amp;  &  ampersand
&apos;  '  apostrophe
&quot;  "  quotation mark

Always escape them in your content and don’t use them in XML names
Root Element and Nesting

- An XML document has **one single root element**!

Having elements within another element is called **nesting**

```
<p>A paragraph includes a specific
   <w>word</w> and other words.
   <s>And some sentences.</s>
   <s>And some sentences.</s>
</p>
```

No cross-nesting!
Tree structure

- The nesting of the elements, beginning with one root results in a tree structure

- Ordered Hierarchy of Content Objects (OHCO)
  - Content objects (nodes/elements)
  - Hierarchical (Relation between nodes)
  - ordered (Sequence of nodes)
Example

Hamlet

<?xml version="1.0" encoding="UTF-8"?>
<div n="1">
  <speech>
    <speaker>HAMLET</speaker>
    <line>I would not hear your enemy say so.</line>
    <line>Nor shall you do mine ear that violence.</line>
    <line>To make it truer of your own report</line>
    <line>Against yourself: I know you are no truant.</line>
    <line>But what is your affair in Elsinore?</line>
    <line>We'll teach you to drink deep ere you depart.</line>
  </speech>
  <speech>
    <speaker>HORATIO</speaker>
    <line>My lord, I came to see your father's funeral.</line>
  </speech>
  <speech>
    <speaker>HAMLET</speaker>
    <line>I pray thee, do not mock me, fellow-student;</line>
    <line>I think it was to see my mother's wedding.</line>
  </speech>
</div>
XML Document structure

- XML declaration
  `<?xml version="1.0" encoding="UTF-8"?>`
- Processing instructions
  `<?xml-stylesheet type="text/xsl" href="transformation.xsl"?>`
- Root element + Nested elements
  `<root> ... </root>`
- Comments
  `<!-- This is a comment -->`
XML Document structure

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="transformation.xsl"?>

<!-- Metadata starts here -->

<item>
  <fullTitle>The waste land: a facsimile and transcript of the original drafts, including the annotations of Ezra Pound / T. S. Eliot ; edited by Valerie Eliot</fullTitle>
  <published>
    <publicationPlace>London</publicationPlace>
    <publicationDate>1971</publicationDate>
  </published>
  <publisher>Faber &amp; Faber</publisher>
  <created>before October 1922</created>
  <format>Facsimile / Manuscript / Typescript</format>
  <language>English</language>
  <creator>
    <name>T S Eliot</name>
    <name>Ezra Pound</name>
    <name>Vivienne Eliot</name>
  </creator>
</item>

<!-- Another comment -->
XML Processing

XML: Encode information
Schema: Check data quality
XPath: Navigate and query data
XSLT: Transform data

Source: Fritze, 2017
Oxygen editor

Oxygen is a text editor/development environment specifically for the editing of XML Documents.

- Creating and editing of XML
- Checking and validating XML
- Processing XML
Oxygen editor

- Platform-independent
- Subversion client, Add-on for Git integration
- Supports TEI
- Supports all popular schema languages
- Syntax completion
- Integrated documentation
- Built-in XSLT and FOP processors
Is your document well-formed?

A well-formed XML document fulfills the rules of the standard:

1. There is exactly one root element
2. Each element has a start-tag and an appropriate closing-tag
3. Elements have to be properly nested - no overlapping structures
4. Attribute values have to be quoted
5. An element cannot have two attributes with the same name
6. Reserved characters have to be escaped

Harold/Means, 2004
Is your document well-formed?

- `<name>Franch Kafka</name>`
- `<name><forename>Franch</forename><surname>Kafka</surname></name>`
- `<name><forename>Franch</forename><surname>Kafka</surname></name>`
- `<name type="person">Franch Kafka</name>`
- `<name type="person">Franch Kafka</name>`
- `<name type="person">Franch Kafka</name>`
- `<name type="person">Franch Kafka</name>`
- `<name>Franch Kafka</name>`
- `<name type="person" type="schriftsteller">Franch Kafka</name>`
- `<name type="person author">Franch Kafka</name>`
Validation

In addition to being well-formed a document can also be checked for being valid.

A valid XML document fulfills a set of rules defined in a specific schema, attached to it, which for example defines:

- the vocabulary (element and attribute names) used
- the structure of a document and the sequence of elements

Different schema languages, e.g. Document Type Definition (DTD) or XMLSchema, allow different types of validation.
Document check

```
<poem version="1.0" encoding="UTF-8">
  <The Waste Land>
  By T. S. Eliot
  FOR EDRA POEDD
  IL MIGLIOI FABBRO
  I. The Burial of the Dead
  April is the cruellest month, breeding
  Lilacs out of the dead land, mixing
  Memory and desire, stirring
  Dull roots with spring rains.
  Winter kept us warm, covering
  Earth in forgetful snow, feeding
  A little life with dried tubers.
  Summer surprised us, coming over the starnberg pepper
  With a shower of rain: we stopped in the colonnades,
  And went on in sunlight, into the Hofgarten,
```

Document is well formed.
Exercise

The Wasteland
by T. S. Eliot

There are many interesting entities and phenomena to markup in this poem!

Source: British Library
Exercise

- Start the Oxygen XML editor
- Open a new XML Document


- Create a root element
- Copy the text from the File ‘TheWasteland.txt’ into your root element
- Describe the structural elements and layout of the first part of the poem (*I. The Burial of the Dead*) with meaningful XML elements and attributes

Brainstorming:
Which relevant structural and content-related elements of the text can you identify?
Which metadata could be added to the text?
Exercise
Material

- Manuscript Facsimile at the British Library:
  https://www.bl.uk/collection-items/manuscript-of-t-s-eliots-the-waste-land-with-ezra-pounds-annotations

- Full text at the Poetry Foundation:
  https://www.poetryfoundation.org/poems/47311/the-waste-land
Finished?

- Check if your XML document is well-formed.
  Ctrl + Shift + w | ✔

- Save your work on your computer
  Ctrl + s | 📖

- Upload your files to the ‘Exercises’ folder so that we can compare them - [https://tinyurl.com/Textencoding-with-XML](https://tinyurl.com/Textencoding-with-XML)

  - What did you mark up?
  - Did you use elements or attributes?
References

XML in a Nutshell by Elliotte Rusty Harold, W. Scott Means 2004 by O'Reilly Media, Third edition

Textkodierung mit XML, Summer School “Digitale Edition” 2017, Christiane Fritze


[http://www.w3schools.com/xml/](http://www.w3schools.com/xml/)