Grundlagen der Textauszeichnung

Begrüßung

mit XML

Montag, 10.2., 14:00-15:30

Organisatorisches

- Danksagung
  - Institutionen: BUW, Leopoldina, BBAW, GRK2196, IZED
  - Menschen: Ann-Kathrin Haustein, Angela Kump, Jana Klinger
- Lehrteam
  - PS, Torsten Roeder, Markus Schnöpf, Nadine Sutor, Nadine Dorscheid
- Programm, Abendprogramm
  - Mo: Café Simons, Untergrünewalderstraße 3
  - Di: Stadtführung? 18:30-19:30
- Kursordner: https://tinyurl.com/wsde20
- oXygen läuft?
- Zertifikate, Scheine
- Alumni-Mailing

------START-LICENSE-KEY------

Registration_Name=Bergische Universit\u00E4t Wuppertal

Company=Bergische Universit\u00E4t Wuppertal

Category=Academic-Classroom

Component=XML-Editor, XSLT-Debugger, Saxon-SA

Version=21

Number_of_Licenses=1

Date=04-16-2019

Subscription=416

SGN=MCwCFGpKDkb70pWJZHzjNjRoFB+1+DuWAhREarIL0QlWiveCoQGVwvhiLiixUA\=\=

-------END-LICENSE-KEY-------

Notfalllizenz

Das Programm der School

1. Grundlagen der Wissensmodellierung: **XML**
2. Eine Sprache für Dokumente und Texte: **TEI**
3. Mit XML-/TEI-Daten umgehen können: **XPath, XSLT, Architekturen, Werkzeuge**
4. Digitale Editionen: **Theorien und Methoden; Best Practice (Abendvortrag); Evaluation**

Sinn? Unsinn? Bedarfe?

Diese Stunde

1. Das Internet! HTML
2. XML ist wie HTML, nur anders. Erst die Praxis ...
   a. oXygen als Editor
   b. Transkription und Textauszeichnung
   c. Wissensmodellierung
3. … dann die Theorie: XML als Datenmodell und -standard.
   a. XML im Kontext
   b. XML-Syntaxregeln
   c. Die X-Familie

Das Internet? Alles ist HTML!

- Im Prinzip ist jede Internet-Seite eine HTML-Seite
- Rechte Maustaste: Quelltext ansehen | Untersuchen
- HTML - Hypertext Markup Language

```
<html>
  <head> … </head>
  <body>
    <h1>Ich bin eine Überschrift</h1>
    <p>Ich bin ein Absatz. Dies ist ein <a href="http://abc.de">Link</a>
    <img src="dateiname.jpg" alt="ein schönes Bild"/>
  </body>
</html>
```

XML ist wie HTML, nur anders

- oXygen öffnen
- Strg+n - eine neue Datei anlegen
- Erste Zeile verstehen
- Eine Transkription einfügen
- “Auszeichnen” = Wissen explizit machen (Tip: Strg+e)
- Diskutieren: Was? Wie? Auszeichnen?
- Buchstaben, Umbrüche, Positionen, Funktionen, Korrekturen, Kommentare, etc etc

Eine kleine Oxygen-Einführung: https://www.i-d-e.de/wp-content/uploads/2019/11/02_Oxygen.pdf

- Noch ein Ansatz. Neue Datei anlegen. Überlegen: was ist ein Brief? Was wäre ein Briefmodell?

XML und oXygen, wir merken uns …

- XML ist Text, man kann es mit beliebigen Editoren schreiben
- oXygen ist ein XML-Editor mit extremen Leistungsumfang
- Man braucht sehr lange nur einen kleinen Teil der Möglichkeiten
- Vertraut auf die Vorschläge / Autocompletion
- Achtet auf die roten Kringellinien (etwas läuft schief)
- Fehlermeldungen sind Euer Freund
- Rechte Laufleiste: Grün? Gelb? Rot?
- Ja, es gibt andere Ansichten als den Code. Aber aus didaktischen Gründen ...

XML Syntaxregeln

- <?xml version="1.0" encoding="UTF-8"?> = XML-Deklaration
  - Kann von einem Prolog (Verarbeitungsanweisungen) gefolgt werden
- <elementname> … </elementname>
  - Elemente müssen geschlossen werden!
  - <elementname/> - leeres Element
  - Elementnamen sind Schreibungs-sensitiv: <element> ≠ <Element>
  - Regeln für Namen (vertraut oXygen)
- <elementname attributname="attributwert"/>
  - beliebig viele Attribute, aber keine Wiederholung!
  - Attributwert = beliebige Zeichenketten
- <!-- ein Kommentar →
  - Kommentare werden bei der Verarbeitung ignoriert

XML Syntaxregeln

<.../> … die spitzen Klammern haben eine besondere Bedeutung

→ Spitze Klammern im *normalen* Gebrauch: < und >

<name attribut="wert">...</name> … Anführungszeichen mit besonderer Bedeutung → Anführungszeichen im *normalen* Gebrauch: " '

→ Das Kaufmanns-Und hat eine besondere Bedeutung

→ das Kaufmanns-Und (engl.: ampersand) im *normalen* Gebrauch: &

Das allgemeine Konzept der “entities” → &entity;

XML Syntaxregeln

Leerraum / Spaces / Tabs / Zeilenumbrüche:

- … spielen bei XML keine Rolle
- mehrfache spaces, tabs, Zeilenumbrüche werden zu einem space zusammengefasst
- … dienen nur der besseren Lesbarkeit (→ “pretty print”, Strg+Shift+P)

XML-Terminologie

<tagname>Zeichendaten <leeresElement attribut="wert"/> Zeichendaten</tagname>

- Tag, Starttag / Endtag, öffnendes / schließendes Tag
- Element, Elementname, Elementinhalt; Element = Starttag + Inhalt + Endtag
- Attribut = Attributname + Attributwert
- Elemente enthalten Text oder andere Elemente
- Mixed Content (ein Element enthält Elemente und Zeichendaten)
- Elemente sind sauber verschachtelt (they are “nested”)
- Keine Überlappung (kein <a>........)
- Sie bilden eine Hierarchie, einen Baum
- Es gibt ein Wurzelelement
- Eltern, Kinder, Geschwister, Vorfahren, Nachfahren

XML im Kontext

Genese

- Textauszeichnung → Buchdruck
- Markup Languages
- GML - Generalized Markup Language - 1969
- SGML - Standard Generalized Markup Language - 1986
- XML - eXtensible Markup Language - 1998
- Softwareunterstützung, umliegende Standards, Verbreitung, Nutzung
- Konkurrierende Ansätze (ERM, JSON, Graphen)
- Was kommt nach XML?

XML im Kontext

XML - eXtensible Markup Language

- Standard (recommendation) des W3C seit 1998 (version 1.0)
- Andere Versionen? XML 1.1 (2006) ?
- XML ist eine “Auszeichnungssprache”
- XML ist eine “Metasprache”
- XML ist eine “Syntax”
- XML ist ein “Datenmodell”
- XML ist ein Paradigma der Wissensmodellierung

XML als allgemeine Grundlage

- XML ist zeichenbasiert (i.d.R. Unicode), einfach, plattformunabhängig, ein internationaler Standard ()W3C), langzeitarchivierbar ...
- XML ist eine “Metasprache”, die eine allgemeine Syntax und allgemeine Regeln zur Verfügung stellt
  - XML sagt: “Bilde deine eigene Sprache”
- Es gibt viele “Anwendungssprachen”
  - HTML ist eine Seitenbeschreibungssprache für das Internet
  - TEI ist eine Sprache für Dokumente und Texte im Allgemeinen
  - viele weitere Sprachen …
    - DocBook, METS, EAD, SVG, MathML, etc etc etc
  - ... XML is everywhere (?)

XML und konkrete Sprachen

Erfinde Deine eigene Sprache. Oder verabschiede einen Standard ...

- Lexikon (welche Elemente) und Grammatik (wie benutzt) können kodifiziert werden

  → Schemata → Schemasprachen (mehrere verschiedene)

Terminologie:

- Ein XML-Dokument ist **wohlgeformt** (well-formed), wenn es den allgemeinen XML-Regeln folgt (es ist ist dann parse-bar und bildet einen Baum)
- Ein XML-Dokument ist **gültig** (valid), wenn es den Regeln eines Schemas entspricht

Die X-Familie

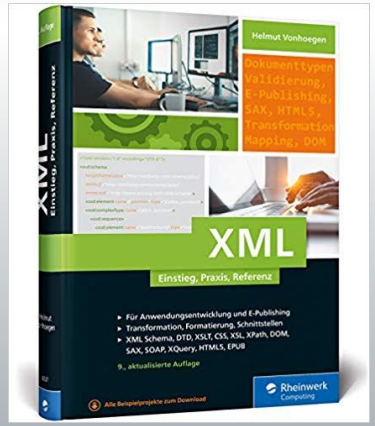
XML ist einfach. Zugleich ist es Teil eines mächtigen Systems weiterer Standards

- XML Infoset
- XLink, XPointer, XML Base, **XInclude**
- **XPath**
- **XSLT**, XSL-FO, **XQuery**, XUpdate, **XProc**
- XForms
- **XML Schema**, DTD, Relax NG, Schematron, **ODD**

Offizieller Startpunkt: https://www.w3.org/XML/

XML lernen

- Bitte beachten: XML ist seit 1998 unverändert - die XML-Praxis aber nicht! Neue Materialien sind besser als alte!
- Extrem niederschwelliges Tutorial: https://www.w3schools.com/xml/
- XML-Tutorials aus [season] Schools, z.B. https://www.i-d-e.de/aktivitaeten/schools/ oder https://digital.humanities.ox.ac.uk/dhoxss (Archiv)
- Immer noch das brauchbarste Buch: Helmut Vonhoegen: XML - Einstieg, Praxis, Referenz. ^{9}2018



Inside XML

- XML ist eine “Metasprache”
- XML ist eine “Syntax”
- XML ist ein “Datenmodell”
- XML ist ein Paradigma der Wissensmodellierung

Zum Charakter von XML …

- XML ist zugleich eine strikte hierarchische (Baum-)struktur und ein sequentieller Strom von Zeichendaten (Text)
- Man unterscheidet zwischen datenzentriertem und dokumentorientierten XML
- Präskriptive und deskriptive Modellierung
- XML ist stark bei semistrukturierten, komplexen, textorientierten Daten

Von der Technik zur Methode: Warum benutzen wir XML?

Scheinbare Nachteile von XML:

- “geschwätzig” (verbose)
- zu komplex: speicherintensiv, langsam in der Verarbeitung
- nicht komplex genug: keine Überlappung
- Keine echte Standardisierung: jedem seine eigene Sprache?
- XPath, XSLT, XQuery - andere Logik als bei anderen Programmiersprachen
- Zuviel “implizites” Wissen. (Paradox: XML = Wissensexplizierung)

Alternativen:

- Relationale Datenbanken (ERM, RDBMS)
- JSON
- Graphen

Von der Technik zur Methode: Warum benutzen wir XML?

- ERM ist für komplexe Textdaten, deskriptive Modellierung und semistrukturierte Daten ungeeignet
- JSON ist wie XML ohne Mixed Content
- Graphen müssen *alles* explizit machen, sind nicht menschenlesbar, modellieren Aussagen - nicht Texte
- XML ist eine allgemeine Sprache, in der wir unser Wissen um komplexe, semistrukturierte Dokumente und Texte gut ausdrücken können
- Wir benutzen XML nur, bis wir etwas besseres gefunden haben ...