

Computerlexikographie-Tutorium 23.05.2008

- Information:
 - E-Buch: Seitenangaben zum E-Buch auf den Tutoriumsfolien sind leider nicht immer treffend, weil ich eine andere Ausgabe des Buches habe.
→ Bitte die Kapitelangaben beachten.
- Thema für heute:
Strukturbeschreibende Auszeichnung der Wörterbuchartikelstrukturen
 - XML
 - DTD

Auszeichnung von Wörterbuchstrukturen

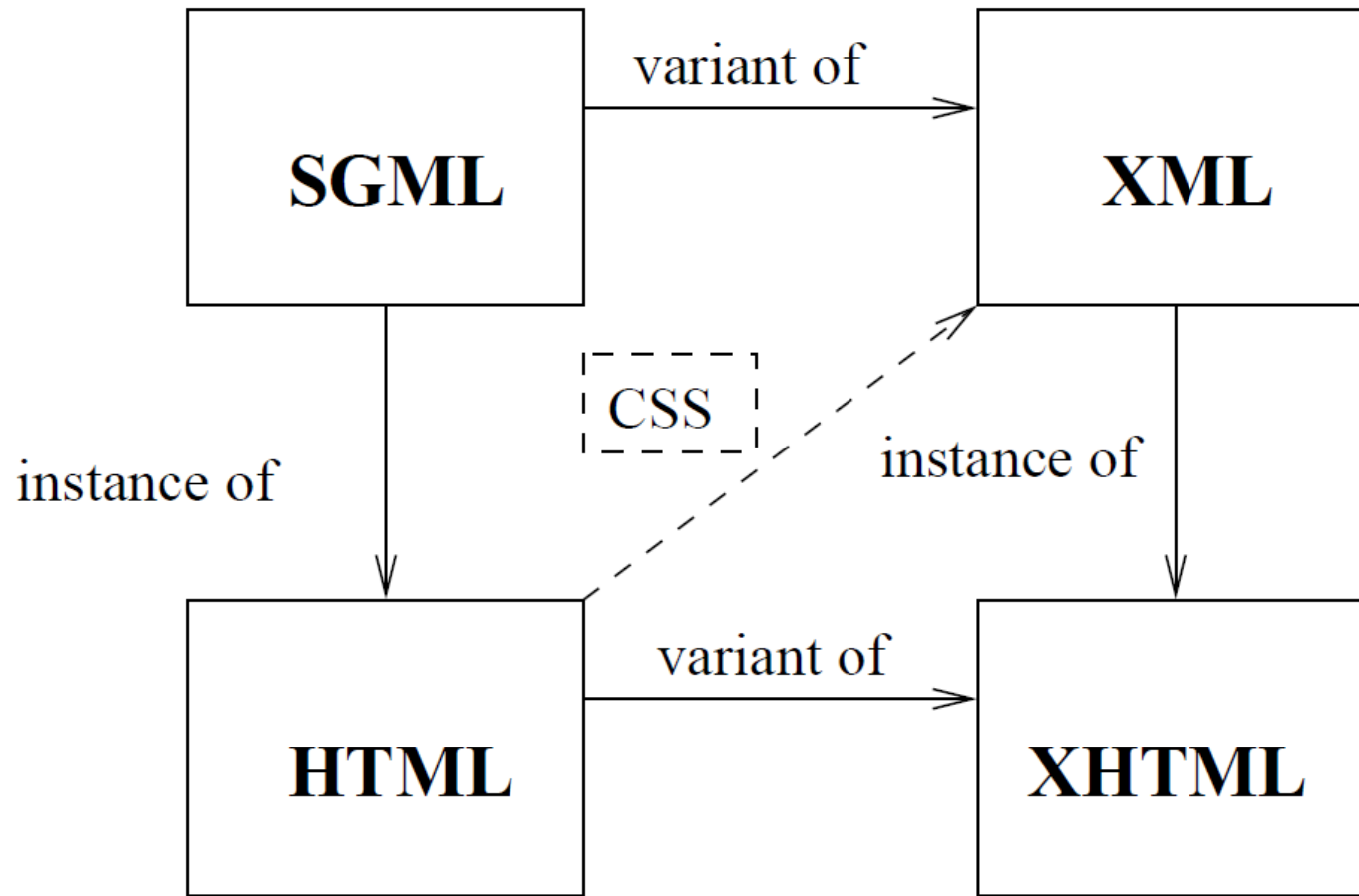
- Die Daten (z.B. ein Wörterbuchartikel) werden in sogenannte **Datenfelder** aufgeteilt, die den Inhalt des Textes als hierarchische Struktur erfassen. Das ist die sogenannte logische Struktur.
- Das **Layout** und die **logische Struktur** können somit getrennt betrachtet werden.
 - Mithilfe von **Auszeichnungssprachen** (markup languages) wird die logische Struktur gespeichert.
 - Das Layout kann mit einem externen Stylesheet definiert werden.
- Die Suche nach bestimmten Informationen erfolgt typischerweise anhand der logischen Struktur.

Auszeichnung von Wörterbuchstrukturen

- **TEI = Text Encoding Initiative (1987):** Standardisierung der Verwendung von Auszeichnungssprachen für alle Arten von Dokumenten.
(Beispiel: <http://www.freedict.org/howto/ch06.html>)
- **SGML = Standard Generalized Markup Language = Normierte Verallgemeinerte Auszeichnungssprache:** ist eine Metasprache, mit deren Hilfe man verschiedene Auszeichnungssprachen für Dokumente definieren kann. (<http://de.wikipedia.org/wiki/SGML>, <http://www.w3.org/MarkUp/SGML/>)
- **XML = Extensible Markup Language = Erweiterbare Auszeichnungssprache:** ist ein einfaches, flexibles Textformat, das auf SGML basiert.
(<http://www.w3.org/XML/>)
- **DTD = Document Type Definition = Dokumenttypdefinition:** beschreibt die Struktur einer Klasse gleichartiger Dokumente in Form von einer kontextfreien Grammatik.

SGML und XML

- (Lemnitzer: XML fundamentals, Folie 9)



XML: Beispiel (vgl. Lemnitzer: XML Fundamentals, 19)

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- movie data base -->
<mdb> ...
  <movie ID="movie2333" genre="crime/drama">
    <title> The Godfather, Part II
  </title>
  <plot outline> The early life and career of Vito
    Corleone is portrayed ...
  </plot outline>
  <cast overview>
    <actor appearance="1" name="Al Pacino" role="Don
      Michael Corleone" />
  </cast overview>
</movie> ...
</mdb>
```

XML: Wohlgeformtheit

- In der ersten Zeile eines XML-Dokumentes sollte (muss nicht) die **XML-Deklaration** stehen:

`<?xml version="1.0" encoding="utf-8" standalone="yes"?>`

- version: 1.0, 1.1
- encoding (z.B.): `iso8859-1`, `utf-8`, `ascii`
- standalone: (optional)
 - **yes**: ohne externe DTD-Deklaration (siehe später)
 - **no**: Wenn es eine externe DTD-Deklaration zum XML-Dokument gibt.
 - Wenn kein standalone-Attribut angegeben wird, wird es standardmäßig auf **no** gesetzt.
- Nach der XML-Deklaration steht – falls eine DTD vorhanden – die **DTD-Deklaration**:

`<!DOCTYPE mdb SYSTEM "mdb.dtd">`

XML: Wohlgeformtheit

- Ein XML-Dokument hat eine **Baumstruktur** (wie auch HTML).
- Knoten des Baumes: Die Elemente, die mit sogenannten **Tags** (~ Etiketten) ausgezeichnet werden.
 - Elemente mit Inhalt:
 - öffnende Tags: `<TAGNAME [ATTRIBUTE="ATTRIBUTWERT"]*>`
 - Attribute sind optional und werden am öffnenden Tag spezifiziert.
 - schließende Tags: `</TAGNAME>`
 - **TAGNAME** der öffnenden und schließenden Tags müssen übereinstimmen.
 - Leere Elemente: `<TAGNAME [ATTRIBUTE="ATTRIBUTWERT"]*/>`
- Kanten ergeben sich von der Einbettung der Tags oder Texteinheiten in andere Tags.

XML: Elemente

- **Inhalt der Elemente:**

- leer: leere Elemente

- `<actor appearance="1" name="Al Pacino" role="Don Michael Corleone" />`

- enthalten

- Text

- `<title> The Godfather, Part II </title>`

- andere Elemente (eingebettet)

- `<cast overview>`

- `<actor appearance="1" name="Al Pacino"`
`role="Don Michael Corleone" />`

- `</cast overview>`

- Text und andere Elemente gemischt (nicht empfohlen)

XML: Wohlgeformtheit

- Baumstruktur:
 - Die Wurzel ist das alles umfassende Tag.
 - Alle anderen Knoten werden in den Wurzelknoten eingebettet.
 - Kreuzung oder Überlappung der Knoten ist nicht erlaubt.
- Tagnamen sind *nicht vordefiniert* ($\leftrightarrow$ HTML), es sollten „sprechende Namen“ gewählt werden.
- Untereinheiten im Tag werden mit Leerzeichen getrennt.
- Tagnamen sind case-sensitive.
- Endtags dürfen nicht weggelassen werden.

XML: Wohlgeformtheit

- **XML-Namenskonvention:**

- XML-Namen dürfen grundsätzlich alle alphanumerischen Charakter enthalten, auch Nicht-ascii-Charakter, Ziffern oder Ideogramme. Sie dürfen kein Leerzeichen enthalten.
- XML-Namen dürfen von den Satzzeichen nur `_` – und `.` enthalten (*kein* `;` `/` `'` `"` `\` `$` `%` `^`) (`:` ist erlaubt, aber nicht empfohlen, weil reserviert für Namensräume).
- Tagnamen, Attributnamen und ID-Attributwerte dürfen nur mit Buchstaben, Ideogrammen oder mit einem Unterstrich beginnen (also *nicht* mit einer Ziffer, einem Bindestrich oder Punkt).
- Der String „XML...” in allen seiner Variationen ist reserviert für XML-Funktionen.

XML: Wohlgeformtheit

- Beispiele für erlaubte und unerlaubte Tagnamen:

☺ `<name>Allan</name>`

☹ `<Name>Allan</name>`

☺ `<Drivers_License_Number>98 NY 32</Drivers_License_Number>`

☹ `<Driver's_License_Number>98 NY 32
</Driver's_License_Number>`

☹ `<Drivers License Number>98 NY 32</Drivers License
Number>`

☺ `<month-day-year>7/23/2001</month-day-year>`

☹ `<month/day/year>7/23/2001</month/day/year>`

☺ `<_4-lane>I-610</_4-lane>`

☹ `<4-lane>I-610</4-lane>`

☺ `<téléphone>011 34 23 11 45</téléphone> <!-- Bemerkung:
encoding muss stimmen! -->`

XML: Wohlgeformtheit

- **Sonderzeichen** müssen geschützt werden – dafür gibt es vordefinierte Entities:
 - `<` `<` less than
 - `>` `>` greater than
 - `&` `&` ampersand
 - `'` `'` apostrophe
 - `"` `"` quotation mark
- Beispiel: `<math>12 + 3x < 100</math>`
- (Entities kann man auch selbst definieren.)
- **Kommentare** dürfen überall im Dokument stehen, nur innerhalb eines Tags nicht.
 - `<!-- Das ist ein Kommentar -->`
 - Kommentare dürfen keinen anderen Kommentar einbetten und keine Sequenz „--“ enthalten.

XML: Attribute

- **Attribute:** sind Attribut-Wert-Paare, die die Elemente näher spezifizieren.
 - **ATTRIBUTNAME="WERT"**
- Attribute müssen innerhalb eines Tags einzig sein.
 - `<word pos="verb">record</word>` oder auch:
 - `<word pos="verb noun">record</word>` aber nicht:
 - ☹ `<word pos="verb" pos="noun">record</word>`
- Man kann eine Eigenschaft sowohl als eingebettetes Element, als auch als Attribut kodieren:
 - `<word><literal>record</literal><pos>verb</pos></verb></word>` oder
 - `<word literal="record" pos="verb"/>`
- Man verwendet Attribute, wenn man es für passend(er) hält, besonders
 - für Metaeigenschaften oder
 - wenn die Werte aus einer definierten, abzählbaren Menge sind.

XML: Validität

- Nur wohlgeformte Dokumente können validiert werden.
- Wohlgeformte Dokumente sind nur dann valid, wenn sie anhand einer DTD validiert werden (können).
- Zum Validieren des Dokumentes:
 - W3 XML Validator:
 - http://www.w3schools.com/xml/xml_validator.asp
 - Brown University Scholarly Technology Group:
 - <http://www.stg.brown.edu/service/xmlvalid/>
 - Validierer von Richard Tobin:
 - <http://www.cogsci.ed.ac.uk/~richard/xml-check.html>

DTD: Beispiel (Lemnitzer: Markup Languages, 10)

```
<!-- DTD for our movie database sample document -->
<!-- Last update: October 2003 -->
<!ELEMENT mdb (movie)+>
<!ELEMENT movie (title, subtitle?, genre?, plot outline?,
    director?, cast overview)>
<!ATTLIST movie ID ID #REQUIRED>
<!ELEMENT cast overview (actor, role+)+>
<!ELEMENT actor (#PCDATA)>
<!ATTLIST actor appearance CDATA #IMPLIED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT subtitle (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT plot outline (#PCDATA)>
<!ELEMENT director (#PCDATA)>
<!ELEMENT actor (#PCDATA)>
<!ELEMENT role (#PCDATA)>
```

DTD

- DTD definiert,
 - welche **Elemente** ein Dokument enthält, insbesondere welches Element das Wurzelelement ist.
 - wie das **Inhaltsmodell** dieser Elemente aussieht (Namen, Art, Abfolge, Inhalt).
 - ob und was für welche **Attribute** die Elemente enthalten.
 - (+ Entities)
- DTD steht entweder in der XML-Datei oder sie ist als externe DTD definiert.
- **Interne DTD:**

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<!DOCTYPE mdb [
```

```
...           ← Da steht die ganze DTD.
```

```
<mdb> ... </mdb> ← Da steht die ganze XML-Datei mit Wurzel mdb.
```

DTD

- **Externe DTD:**

- mdb.xml:

```
<?xml version="1.0" encoding="utf-8" standalone="no"
  ?>
```

```
<!DOCTYPE mdb SYSTEM "mdb.dtd">
```

```
<mdb> ... </mdb>
```

- XML-Deklaration mit standalone="no"
- Die DTD-Deklaration gibt das Wurzelelement (mdb) und den Pfad zur DTD-Datei an.

- mdb.dtd:

```
<!ELEMENT mdb ...>
```

```
...
```

DTD: Elemente

- **Syntax:** `<!ELEMENT ELEMENTNAME (INHALTSMODELL) >`
- **Inhaltsmodell** der Elemente gibt an, welche anderen Elemente in welcher Reihenfolge eingebettet sind oder ob Text eingebettet ist.
 - **EMPTY:** Für leere Elemente.
 - `<!ELEMENT anmerkung EMPTY>`
 - **#PCDATA** = Parsed Character Data: Für Text.
 - `<!ELEMENT beschreibung #PCDATA>`
 - **Aufzählung der Kindknoten:** Für eingebettete Elemente:
 - `<!ELEMENT name (A, B)>`
 - **ANY:** Alle Elemente, die in der DTD definiert werden, sind erlaubt.
 - `<!ELEMENT knoten ANY>`
 - **anderes Element und Text gemischt (nicht empfohlen):**
 - `<!ELEMENT A (#PCDATA | B | C) *>`

DTD: Elemente

- Die Elementnamen müssen der XML-Namenskonvention folgen.
- Die Operatoren *, ?, +, |, () werden wie in regulären Ausdrücken verwendet.
 - **<!ELEMENT name ((A, B?) | (C, D?))+>**
- Wenn möglich, muss gekürzt werden:
 - **<!ELEMENT name (A, (B | D))>**
 - ☹ **<!ELEMENT name ((A, B) | (A, D))>**

DTD: Attribute

- **Syntax:** `<!ATTLIST ELEMENTNAME ATTRIBUTNAME ATTRIBUTTYP DEFAULTWERT (E)>`
 - Beispiel: `<!ATTLIST word pos CDATA #REQUIRED>`
- Innerhalb eines Elementes darf ein Attributname nicht mehrmals vorkommen.
- Die Attributnamen müssen der XML-Namenskonvention folgen.
- Zu einem Element können auch mehrere Attribute in einer ATTLIST-Deklaration definiert werden:
 - `<!ATTLIST word
 pos (noun | verb | adj) #REQUIRED
 infl CDATA #IMPLIED
 var CDATA #IMPLIED
 >`

DTD: Attribute

- **Attributtypen:**

- **CDATA** = Character Data: Für XML-wohlgeformten Text. Der am meisten verwendete Attributtyp.

- `<!ATTLIST word pos CDATA #REQUIRED>`

- XML: `<word pos="verb">record</word>`

- **NMTOKEN / NMTOKENS**. = Name Token(s): XML Name Tokens müssen der XML-Namenskonvention nicht folgen, aber sie dürfen kein Leerzeichen enthalten. NMTOKENS ist eine Liste von NMTOKEN.

- `<!ATTLIST performances dates NMTOKENS #IMPLIED>`

- XML: `<performances dates="24-09-2008 23-10-2008">
The Film</performances>`

- **Aufzählen der Werte**: Wenn es eine definierte, abzählbare Menge der erlaubten Werte gibt.

- `<!ATTLIST person sex (male | female) #REQUIRED>`

- XML: `<person sex="male">Allan</person>`

DTD: Attribute

- **Attributtypen:**

- **ID:** Der Wert dieses Attributtypes muss der XML-Namenskonvention folgen. Ein bestimmter Wert darf innerhalb des Dokumentes nur einmal vorkommen.
 - `<!ATTLIST word id ID #REQUIRED
pos CDATA #REQUIRED>`
 - XML: `<word id="word232" pos="verb">record</word>`
- **IDREF / IDREFS:** Ein Referenz bzw. eine Liste von Referenzen auf ein Element-ID innerhalb des Dokumentes. Ein IDREF-Wert darf im Dokument auch öfters vorkommen.
 - `<!ATTLIST translate trans IDREF #REQUIRED>`
 - XML: `<translate trans="word232">aufnehmen
</translate>`
- + (ENTITY / ENTITIES, NOTATION)

DTD: Attribute

- **Defaultwerte:**
 - **#IMPLIED:** Das Attribut ist optional.
 - `<!ATTLIST person home CDATA #IMPLIED>`
 - **#REQUIRED:** Das Attribut ist obligatorisch.
 - `<!ATTLIST word pos CDATA #REQUIRED>`
 - **#FIXED:** Ein vordefinierter Attributwert ist angegeben.
 - `<!ATTLIST dictionary lang CDATA #FIXED „english“>`
 - **Literal:** Ein standardmäßiger Attributwert ist als String angegeben. Dieser Wert kann durch ein explizit erscheinendes Attribut-Wert-Paar überschrieben werden.
 - `<!ATTLIST word pos CDATA "other">`

Quellen

- Gossner, Peter & Michael Bunk (2005): *FreeDict HOWTO: Chapter 6. Writing Text Encoding Initiative XML files*. Link: <http://www.freedict.org/howto/ch06.html> (Stand: 08.05.2008, 15:13)
- Harold, Elliotte Rusty & W. Scott Means (2004): *XML in a nutshell*. 3. Auflage. Beijing / Cambridge / Farnham / Köln / Paris / Sebastopol / Taipei / Tokyo : O'Reilly.
- Kunze, Claudia & Lothar Lemnitzer (2007): *Einführung in die Computerlexikographie*. Tübingen : Narr.
- Lemnitzer, Lothar (o.J.): *Markup Languages: XML fundamentals*. Link: <http://www.cl.uni-heidelberg.de/kurs/ss06/texttech/slidesMLang.pdf> (Stand: 18.05.2008, 23:31)
- Lemnitzer, Lothar (o.J.): *Markup Languages: Document Grammars*. Link: <http://www.cl.uni-heidelberg.de/kurs/ss06/texttech/slidesDG.pdf> (Stand: 18.05.2008, 23:33)
- Weitere Internetseiten: <http://www.freedict.org/howto/ch06.html>, <http://de.wikipedia.org/wiki/SGML>, <http://www.w3.org/MarkUp/SGML/>, <http://www.w3.org/XML/>
- XML-Validierer: http://www.w3schools.com/xml/xml_validator.asp, <http://www.stg.brown.edu/service/xmlvalid/>, <http://www.cogsci.ed.ac.uk/~richard/xml-check.html>