

XML-Technologien

Cristina Vertan

Inhalt

- Grundlegende Technologie: SGML
- XML- Prinzipien
- XML-anwendungen
- XHTML
- XML- detaillierte Syntax
- XM-Schema

Was ist SGML ? (Standard Generalized Markup Language)

Internationaler Standard, der die Regeln beschreibt, mit denen man Strukturen eines Dokuments in diesem selbst beschreiben kann.

Legt keine spezifische Dokumentstruktur fest, sondern

definiert Regeln und die Syntax zum Aufbau strukturierter Dokumente

SGML-Dokumente sind nicht formatiert oder layoutiert

Die Darstellung der SGML-Dokumente ist Aufgabe eines Browsers

ist kein Format, sondern

Basiert nur auf 7-Bit-ASCII

ist nicht neu, sondern

Wurde 1970 entwickelt für Aufbereitung und Austausch von Rechtstexten

25.10.2004

Proseminar SemWeb WiSe04/05

3

Vorteile von SGML

- Etablierter Standard zur Strukturierung von Dokumenten
- SGML-Dokumente
 - [sind plattformneutral](#)
 - [besitzen ein einheitliches Erscheinungsbild](#)
 - [besitzen eine "Checkliste" für den Redakteur](#)
 - [leichter recherchierbar als spezifisch formatierte Texte](#)
 - [sind medienneutral aufgebaut](#)
 - [benötigen weniger Speicherbedarf als konventionelle Texte](#)
- SGML-Bestandteile sind wiederverwendbar.

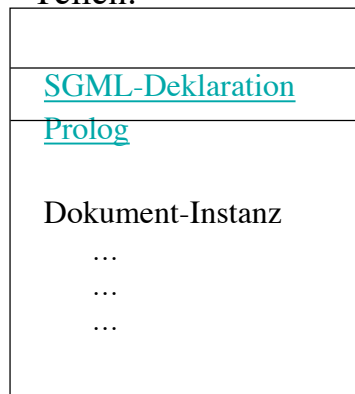
25.10.2004

Proseminar SemWeb WiSe04/05

4

Aufbau von SGML-Dokumenten

- Ein vollständiges SGML-Dokument besteht aus 3 Teilen:



Der Prolog einer Markup-Sprache hat nichts mit der Programmiersprache „Prolog“ zu tun!

25.10.2004

Proseminar SemWeb WiSe04/05

5

SGML-Deklaration

- Beschreibt formal die für ein bestimmtes Dokument verwendeten Teile des gesamten SGML-Standards,
- beinhaltet Informationen für die Weiterverarbeitung von SGML-Dokumenten (z.B. durch externe Dateien),
- wird vom SGML-System gelesen und interpretiert,
- definiert:
 - den Dokumentzeichensatz, d.h. die als Markierung zu interpretierenden Zeichen,
 - die zulässige Schachtelungstiefe von Strukturen usw.,
- Ist sehr schwer für SGML-Neulinge zu durchschauen,
- viele SGML-Systeme (z.B. FrameMaker+SGML) greifen deshalb auf eine Standard-SGML-Deklaration zurück.

25.10.2004

Proseminar SemWeb WiSe04/05



6

Der Prolog und die DTD (Dokument Type Definition)

- Der Prolog enthält die Strukturregeln (DTD) für ein SGML-Dokument
- DTD = die Definition der Struktur und der Strukturelemente für eine Klasse von SGML-Dokumenten.

DTD Datei-Beispiel:	Datei anthologie.dtd :
<!ELEMENT anthologie	-- (gedicht+)>
<!ELEMENT gedicht	-- (titel?, strophe+)>
<!ELEMENT titel	- O (#PCDATA)>
<!ELEMENT strophe	- O (reihe+)>
<!ELEMENT reihe	O O (#PCDATA) >

Der Prolog zitiert dann die Datei **anthologie.dtd** :

Prolog Beispiel:

```
<DOCTYPE Anthologie SYSTEM "c:\...\anthologie.dtd">
```

25.10.2004

Proseminar SemWeb WiSe04/05

7

Eine Dokument-Instanz

<ELEMENT anthologie	-- (gedicht+)>
<ELEMENT gedicht	-- (titel?, strophe+)>
<ELEMENT titel	- O (#PCDATA)>
<ELEMENT strophe	- O (reihe+)>
<ELEMENT reihe	- O (#PCDATA) >

<anthologie>
 <gedicht>
 <titel>An die Muse
 <strophe>
 <reihe> Was ich ohne dich wäre, ich weiß es nicht - aber mir grauet,
 <reihe> Seh ich, was ohne dich Hundert' und Tausende sind.
 </strophe>
 </gedicht>
 <!-- ...andere Gedichte ... -->
 </anthologie>

Die Syntaxkorrektheit wird
von einem Parser überprüft

25.10.2004

Proseminar SemWeb WiSe04/05

8

SGML-Konstrukte

- Erscheinen im DTD-Teil.

Elemente: legen die Bestandteile des Dokumentinhaltes fest

```
<!ELEMENT gedicht -- (titel?, strophe+)>
```

Attribute: geben zusätzliche Informationen zu einem Element

```
<!ATTLIST gedicht sprache (deutsch | englisch) deutsch>
```

In der Dokumentinstanz: `<gedicht sprache=englisch>`

Text `</gedicht>`

Entitäten: abstrakte Bezeichnung für Daten

```
<!ENTITY uuml "ü">
```

In der Dokumentinstanz: `<reihe>.. ist für... </reihe>`

25.10.2004

Proseminar SemWeb WiSe04/05

9

SGML-Tools

- SGML-Parser: überprüfen das SGML-Dokument hinsichtlich:
 - der Gültigkeit der DTD im Sinne von SGML
 - der Konformität der Dokumentinstanz bezüglich DTD
- SGML-Browser: Anzeigesysteme für SGML-Dokumente
- SGML-Editoren:
 - native SGML-Editoren (z.B. look and feel einer Datenbankoberfläche)
 - WYSIWYG - Editoren

25.10.2004

Proseminar SemWeb WiSe04/05

10

SGML-basierte Anwendungen

- MARTIF ist ein SGML-basiertes Austauschformat für terminologische Daten
 - fachsprachliche Kommunikation braucht korrekte Terminologie.
 - Abhilfe für die Probleme
 - traditionelle Medien (Fachwörterbücher, Glossare usw.) wurden durch Entwicklungen im Bereich der elektronischen Datenverarbeitung stark verdrängt
 - Terminologiedatenbanken wurden von jeder Nutzergruppe anders definiert.
 - 1997 wurde das Terminologie-Austauschformat MARTIF (MACHINE-Readable Terminology Interchange Format) definiert
 - spezifiziert die DTD des SGML-Dokuments mit den entsprechenden Tags für die Strukturierung der Daten

25.10.2004

Proseminar SemWeb WiSe04/05

11

```

<martif>
<martifHeader>
.....
</martifHeader>
<text>
<body>
<termEntry>
<descripGrp>
<descrip type='subjectFieldLevel1'>appearance of material</descrip>
<ntig lang=de>
<termGrp>
<term>Opazität</term>
<termNote type='partOfSpeech'>n</termNote>
<termnote type='grammaticalGender'>f</termnote>
</termGrp>
<descripGrp>
<descrip type='definition'> Maß für Lichtundurchlässigkeit
</descrip></descripGrp>
</ntig>

```

Martif-Beispiel

25.10.2004.....

Proseminar SemWeb WiSe04/05

12

SGML-basierte Anwendungsklassen

- SGML-basierte bibliographische Datenbank für Nachschlagewerke
- SGML-basierte Publikationprozesse im Verlag
- Semantisches Markup zur Inhaltserschließung von Agenturmeldungen
- Computerunterstützte Textanalyse (Textannotation)



25.10.2004

Proseminar SemWeb WiSe04/05

13

SGML-Dokumente sind plattformneutral

- Da es sich bei SGML-Dokumenten um reine ASCII-Texte mit Strukturauszeichnungen handelt, sind SGML-Dokumente weitestgehend plattform- und softwareunabhängig.
- Neben einer beliebigen Portierbarkeit wird somit die Langlebigkeit von Dokumenten unterstützt, da SGML eben nicht auf einem herstellereigenen Speicherformat, speziellen internen Formatierungen oder gar speziellen Hardwarevoraussetzungen basiert.



25.10.2004

Proseminar SemWeb WiSe04/05

14

SGML-Dokumente besitzen ein einheitliches Erscheinungsbild

- Durch “formatierungsunabhängige” Strukturbeschreibungen für Dokumente ist ein einheitlicher und konsistenter Aufbau der Dokumente gewährleistet. Prüfprogramme (Parser) gewährleisten die Vollständigkeit und syntaktische Korrektheit der (Instanz-) Dokumente.



25.11.2004

Proseminar SemWeb WiSe04/05

15

SGML-Dokumente besitzen eine “Checkliste” für den Redakteur

- SGML-fähige Redaktionssysteme, wie z.B. FrameMaker+SGML, unterstützen den Redakteur bei der Erstellung von Dokumenten durch einen kontextsensitiven Elementkatalog



25.11.2004

Proseminar SemWeb WiSe04/05

16

SGML-Dokumente sind leichter recherchierbar

- Gut strukturierte Dokumente ermöglichen eine viel genauere Recherchemöglichkeit von Informationen
- z.B. Eine Volltextrecherche nach dem Begriff *“Werkzeug”* ist unpräziser als die Abfrage *“alle Kapitel die in der Kapitelüberschrift den Begriff “Werkzeug” enthalten”*. Das ist eine Information, die man über das Layout allein nicht erhält.



25.11.2004

Proseminar SemWeb WiSe04/05

17

SGML-Bestandteile sind wiederverwendbar

- Da SGML-Dokumente modular aufgebaut sind, lassen sich die Komponenten eines SGML-Dokumentes einzeln ablegen, z.B. In einer SGML-Datenbank, und in unterschiedlichen Kontexten zur Erreichung von Einheitlichkeit gezielt wiederverwerten.
- Beispiele: Definitionen in einem Lehrbuch, Ergebnisse von Bundesligaspielen, Loseblattsammlung mit Verordnungen



25.11.2004

Proseminar SemWeb WiSe04/05

18

SGML-Dokumente sind trägerneutral aufgebaut

- SGML eignet sich hervorragend für eine träger- (medien-) neutrale Informationsaufbereitung.
- Ein Browser zeigt ein SGML-Dokument an, daraus kann man z.B. drucken
- SGML-Dokumente können z.B. zusammen mit einem geeigneten Browser auf eine CD gebracht und verteilt werden.
- SGML-Dokumente können ohne großen Aufwand nach HTML konvertiert werden und damit Internet-fähig sein.



25.11.2004

Proseminar SemWeb WiSe04/05

19

SGML-Dokumente benötigen weniger Speicherbedarf

- SGML-Dokumente sind nur ASCII-Dokumente deshalb ist der Speicherbedarf um ein Vielfaches geringer als bei herkömmlichen Dokumenten aus einer Textverarbeitung.



25.11.2004

Proseminar SemWeb WiSe04/05

20

SGML und das Web

- SGML ist sehr gut geeignet für "large-scale document management business".
- Nicht so gut geeignet für Web-Publikationen denn:
 - Komplexe Software ist erforderlich, denn SGML ist sehr kompliziert und erfordert einen entsprechend komplizierten Parser, der für On-line Arbeiten im Netz (z.B. Browsen) zu langsam sein würde,
 - Nach dem SGML-Standard wäre für jedes Dokument unbedingt eine DTD nötig,
 - die DTDs haben eine sehr komplexe Struktur und sind nicht ohne weiteres wiederbenutzbar oder änderbar,
 - zwei Dokumente mit zwei unterschiedlichen DTDs können nicht gemischt werden.

25.10.2004

Proseminar SemWeb WiSe04/05



21

XML versus SGML

- XML ist ein "application profile" von SGML: ein SGML-System kann XML-Dokumente lesen
- XML ist eine beschränkte Teilmenge von SGML
- XML hat keine eigene Deklaration
- XML kann SGML nicht ersetzen. SGML ist eine bessere und abstraktere Lösung für die Erzeugung und Entwicklung von komplexen Dokumenten und Datenbanken.

25.10.2004

Proseminar SemWeb WiSe04/05

22

Was ist XML ? (Extensible Markup Language)

- XML ist eine eingeschränkte Markup-Sprache für Dokumente, die strukturierte Information enthalten
- strukturierte Information ist z.B.:
 - Inhalt (Wörter, Bilder, usw.)
 - Hinweis über Inhaltbedeutung (Kapiteltitle, Überschriften, usw.)
- Die XML-Spezifikation definiert einen Standard für vereinfachte Markupentwicklung und ohne eine komplizierte Deklaration.
- Dokumente können sein:
 - Konventionelles Standarddokument
 - Vektor-Graphik
 - e-commerce-Transaktionen
 - Mathematische Formeln
 - meta-Daten, usw.

25.10.2004

Proseminar SemWeb WiSe04/05

23

XML Tools

- Editor: ein normaler ASCII-Editor
- Dokumente können mit Web-Browsern angesehen werden (z.B. Netscape Navigator ab Version 5, Internet Explorer ab Version 4.5)
- HTML-Tags können innerhalb von XML-Dokumenten benutzt werden.
- XML enthält keinen vordefinierten Tag für Hyperlinks. (dafür muss XML Linking Language - XLS - benutzt werden)
- XML-Dokumente können in HTML-Dateien eingelesen und z.B. mit JavaScript ausgewertet werden.

25.10.2004

Proseminar SemWeb WiSe04/05

24

XML Beispiel

- Syntax ist ähnlich wie SGML.

```

<?xml version="1.0"?>
<?xml-stylesheet href="style.css" type="text/css"?>
<!DOCTYPE Dokument [
<!ELEMENT Dokument (Titel, Abstrakt?, Kapitel+, Zusammenfassung, Bibliographie)>
  <!ELEMENT Titel (#PCDATA)>
  <!ELEMENT Abstrakt (#PCDATA)>
  <!ELEMENT Kapitel (Titel, #PCDATA)>
  <!ELEMENT Zusammenfassung (#PCDATA)>
  <!ELEMENT Bibliographie (Reihe+)>
  <!ELEMENT Reihe (#PCDATA)>
]>
<Dokument>
<Titel> XML Einf&uuml;hrung</Titel>
<Abstrakt> .....Text .... </Abstrakt>
....
</Dokument>
    
```

} XML-Prolog
} DTD
} Dokument-Instanz

Die Formatierung wird danach durch eine CSS - Datei (Cascading Style Sheets) erzeugt.

25.10.2004 Proseminar SemWeb WiSe04/05 25

XML- Erweiterungen

- LT-XML**
 - Entwickelt an der Universität Edinburgh (Language Technology Group)
 - XML parser + flexible API + Toolsammlung für XML marked-up Dokumenten-Verarbeitung

```

<p id='pl'>
<s id='sl'>
<w pos='prep'>In</w>
<w pos='art'>the</w>
<w pos='n'>beginning</w>
<w pos='v'>was</w>
<w pos='art'>the</w>
<w pos='n'>word</w><c>.</c>
</s>
</p>
                
```

25.10.2004 Beispiel.xml

%textonly -s 'sample.xml'
In the begining was the world.

↑
Ergebnisse der Auswertung

↓

%sgcount sample.xml

```

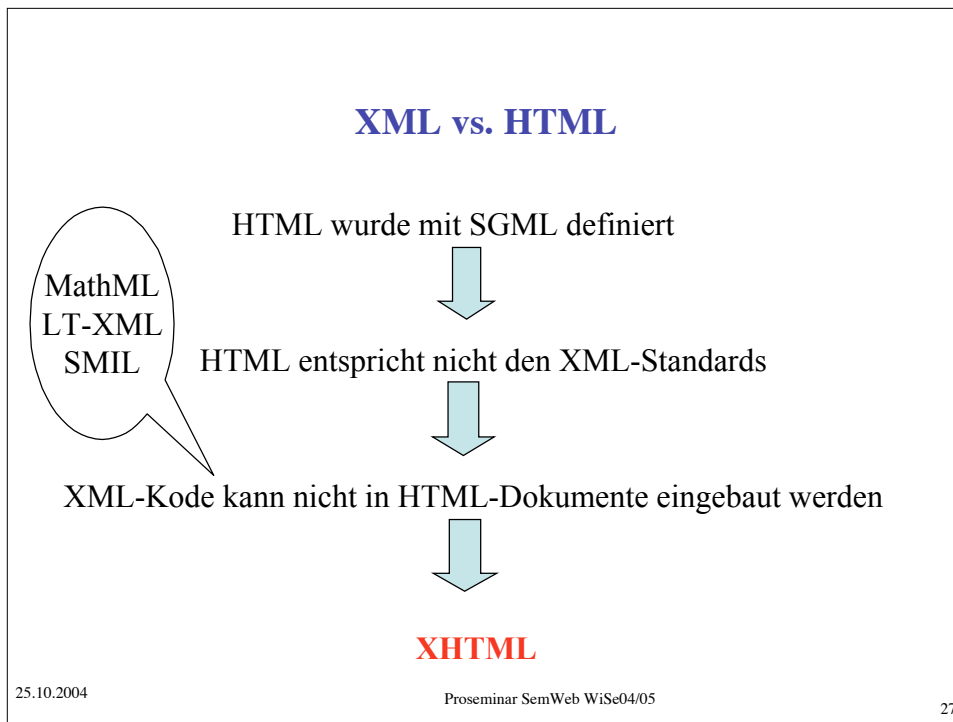
p 1
s 1
w 6
c 1
                
```

%sggrep -q '.*w[pos="n"]' sample.xml

```

<w pos='n'>beginning</w>
<w pos='n'>word</w>
                
```

Proseminar SemWeb WiSe04/05 26



Was ist XHTML ?

- Redefinition von HTML tags entsprechend der XML-Syntax
- die Dokumentstruktur bleibt dieselbe

```

<html>
  <head>
    <!-- ... Head-Inhalt ....-->
    <title> Dokumenttitel </title>
    <!-- ... Head-Inhalt ... -->
  </head>
  <body>
    <!-- ... Body-Inhalt ... -->
  </body>
</html>

```

- Einige Tags haben in XHTML **NICHT** dieselben Namen wie in HTML

25.10.2004 Proseminar SemWeb WiSe04/05 28

XHTML vs. HTML

- Alle Tag- und Attribut-Namen müssen in Kleinbuchstaben geschrieben werden (weil XML kleine und grosse Buchstaben unterscheidet)
- Es gibt keine optionalen Ende-tags. Alle Tags **müssen** ein Paar sein (also auch z.B. `<p>...</p>`)
- Alle leeren Tags enthalten wie in XML ein Leeres-Element-Tag
`<hr />`
- es gibt nur ein einziges `head`- und ein einziges `body`- Element. Stattdessen kann man nur ein einziges `frameset` -Element einfügen
- jedes `head`-Element darf nur ein einziges `title` -Element (Tag) enthalten

25.10.2004

Proseminar SemWeb WiSe04/05

29

XHTML

vs. HTML (Beispiel)

<pre> <html> <head> <title> Vorlesung CP Content</title> </head> <body> <h1> Vorlesung CP </h1> <hr /> <h2>Inhalt</h2> 01 Intro 02 Theorie </body> </html> </pre>	<pre> <HTML> <body> <h1> Vorlesung CP </h1> <hr> <h2>Inhalt</h2> 01 Intro 02 Theorie </body> </html> </pre>
---	--

Viele HTML-Editoren ergänzen HTML zu XHTML

25.10.2004

Proseminar SemWeb WiSe04/05

30

Strictly Conforming XHTML

- Entspricht einem strikten XML-Formalismus:
 - Spezifiziert, dass das Dokument vollständig XML-formatiert ist:
`<?xml version="1.0" charset="iso-8859-1" ?>`
 - benennt eine DTD
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict// EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd" >`
 - das `<html>` Element muß ein "xmlns" Attribut enthalten, um zu spezifizieren wo die Elementnamen definiert sind:
`<html xmlns="http://www.w3.org/TR/xhtml1">`
 ...
`</html>`

25.10.2004

Proseminar SemWeb WiSe04/05

31

Strictly Conforming XHTML - Beispiel

```

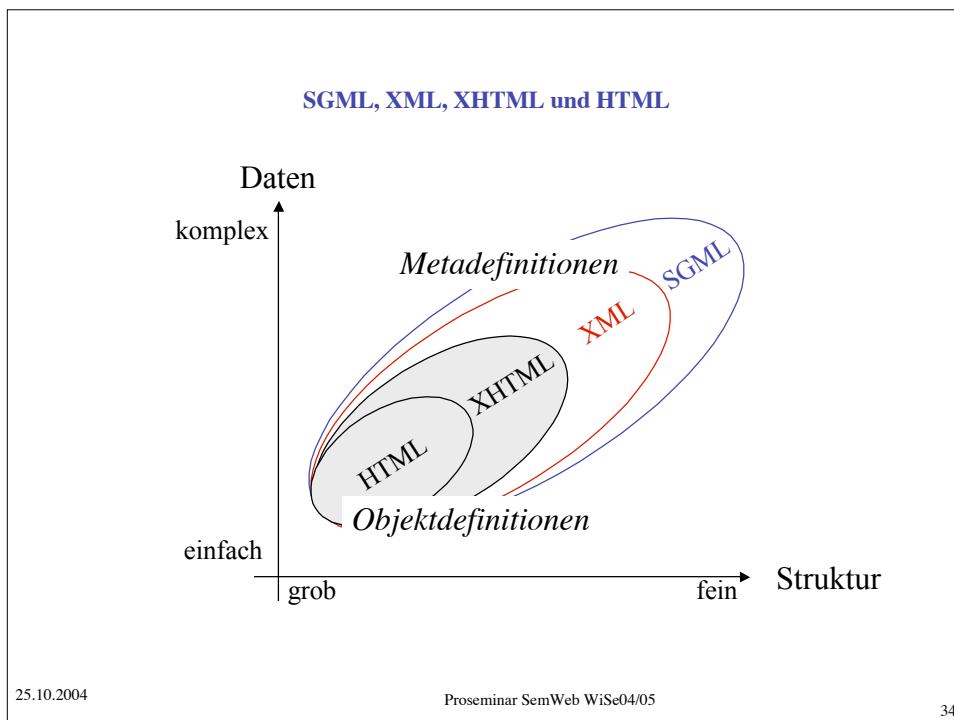
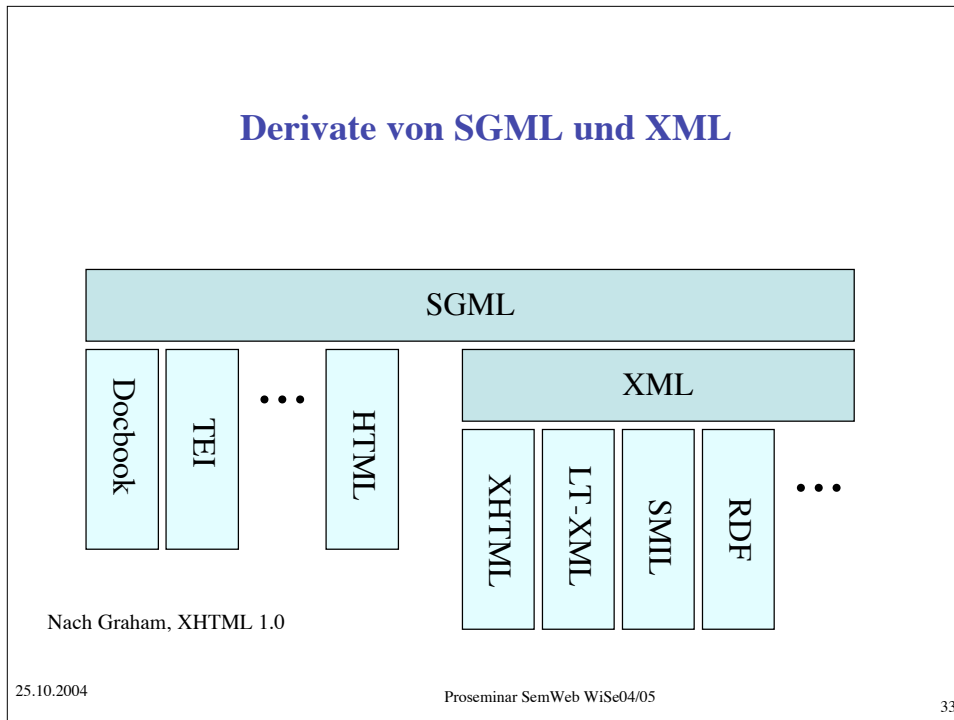
<?xml version="1.0" charset="iso-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Strict// EN"
  "http://www.w3.org/TR/xhtml1/DTD/strict.dtd" >
<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
<title> Vorlesung CP Content</title>
</head>
<body>
.....
</body>
</html>

```

25.10.2004

Proseminar SemWeb WiSe04/05

32



Bestandteile von XML

- Vorspann:
 - Version: z.B. `<?xml version="1.0">`
 - DTD: Verweis auf Grammatik: `<!DOCTYPE...>`
- Elemente (Tags): z.B. `<offer>`, `<store>`, `<book>` usw.
 - Ähnlich wie in HTML aber selbstdefiniert
- Attribute: z.B. für `>price>`: `type="retail"`.
 - Ähnlich wie in HTML aber selbstdefiniert
- Entitäten
 - Zur Wiederverwendung und modularisierung von Texten

25.10.2004

Proseminar SemWeb WiSe04/05

35

Elemente

- Elemente sind die grundlegenden Komponenten eines XML-Dokuments
- Sie werden durch beliebige Namen eingeschlossen in `<` und `>` repräsentiert.
- D.H. zur Definition eines neues Elements definiert man einen neuen Tag sowie den zugehörigen Abschlusstag
- Elemente können innerhalb von anderen Elementen definiert werden (Hierarchien)
- Namenskonventionen wie bei der Programmierung

25.10.2004

Proseminar SemWeb WiSe04/05

36

Beispiel

```
<?xml version="1.0">
<kontakt>
  <name>
    <vorname>Peter </vorname>
    <nachname>Mustermann</nachname>
  </name>
  <email>peter@firma.de</email>
  <telefon> 0236589 </telefon>
</kontakt>
```

- Wurzelement: kontakt
- Elemente: name, vorname, email, telefon
- Drei Hierarchieebenen

25.10.2004

Proseminar SemWeb WiSe04/05

37

Attribute

- Ein Element kann nicht nur Unter-elemente besitzen, sondern auch durch Attribute erweitert werden.
- Ein Attribut wird samt seinem Wert (immer in anführungszeichen) dem Start-Tag eines Elements hinzugefügt.
- Beispiel: ein Kapitel eines Buches hat einen Title und eine Nummer:

```
<chapter title="Introduction" number="4">
...
</chapter>
```

25.10.2004

Proseminar SemWeb WiSe04/05

38

Wann Element, wann Attribut?

- XML gestattet die Datenformatierung auf viele Arten; u.a. ist es praktisch immer möglich anstelle von Attributen auch Elemente zu verwenden und umgekehrt.
- Einige Tipps:
 - Wenn die information selbst wieder komplex ist verwendet man Elemente
 - Modellierung geordneter Information: Elemente
 - Leserlicher ist die Elementvariante, sie nimmt deshalb auch mehr Platz ein.
 - Ist die modellierte Information ein Bestandteil des übergeordneten Elements, verwendet man ebenfalls ein element. Ist sie eine Eigenschaft (wie etwa eine Farbe), dann verwendet man Attribute.

25.10.2004

Proseminar SemWeb WiSe04/05

39

Validierung

- Die strengen Syntaxkonventionen erleichtern die Programmierung von XML Applikationen bereits erheblich (wohlgeformte Dokumente)
- Validierung erlaubt es, die möglichen Baumstrukturen einzuschränken (valide Dokumenten)
- Validierung findet immer gegenüber einer Grammatik statt, d.h., ein valides Dokument ist:
 - Wohlgeformt und
 - Konform zu einer vorgegebenen Grammatik
- Ein grosser Teil der Fehlerabfragen kann so von der applikation an einen Parse übertragen werden

25.10.2004

Proseminar SemWeb WiSe04/05

40

Definition von Grammatiken

- Wenn zwei Anwendungen dieselbe Grammatik verwenden, haben sie dasselbe Verständnis von Daten (wenigstens die Hierarchie)
- Es werden heute zwei verschiedene Ansätze verwendet, um eine Grammatik zu beschreiben:
 - Document Type Definition (DTD)
 - XML Schema

25.10.2004

Proseminar SemWeb WiSe04/05

41

XML Schema als Alternative zur DTD

- Die DTD-Syntax ist nicht XML-konform (sondern SGML)
- Keine Datentypen, nur Strings
- Begrenzte Erweiterbarkeit (Vererbung)
- XML Schema behebt diese Probleme
 - Das Konzept bleibt dasselbe
 - Verschiedene Datentypen
 - Geschrieben in XML
- Vorgehen: schreibe ein Schema in XML, referenziere es von der XML-Datei aus.

25.10.2004

Proseminar SemWeb WiSe04/05

42

Vorteile von XML-Schema

- Jedes XML Schema ist selbst ein XML-Dokument (im Gegensatz zu DTD keine spezielle Syntax mit speziellen Verarbeitungswerkzeugen erforderlich)
- Auch komplexe Integritätsbedingungen formulierbar
- XML Schema enthält vordefinierte und eigendefinierbare Datentypen, wodurch Typprüfung möglich wird
- Bei Datentypen werden Vererbung und substitution unterstützt.
- Benennungskonflikte können durch Verwendung von XML-Namensräumen vermieden werden.

25.10.2004

Proseminar SemWeb WiSe04/05

43

Beispiel für XML-Schema -1-

```
<xsd:schema xmlns:xsd=http://www.w3.org/2002/XMLSchema">
<element name="paper" type="papertype" />
<xsd:complextyp name="papertype">
  <xsd:sequence>
    <xsd:element name="autor" type="xsd:string"/>
    <xsd:element name="titel" type="xsd:string"/>
    <xsd:element name="datum" type="xsd:gYearMonth"/>
    <xsd:element name="link" type="xsd:anyURI min Occurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="typ" type="art"/>
</xsd:complextyp>
```

25.10.2004

Proseminar SemWeb WiSe04/05

44

Beispiel für XML-Schema -2-

```
<xsd:simpleType name="art">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="proc"/>
    <xsd:enumeration value="report"/>
    <xsd:enumeration value="journal"/>
    <!-- usw. -->
  </xsd:restriction>
</xsd:simpleType>
```

25.10.2004

Proseminar SemWeb WiSe04/05

45

Parser

- Um ein XML-Dokument in einer Anwendung zu verwenden, muss man es einlesen und in interne Datenstrukturen der verwendeten Programmiersprache umwandeln
- Es werden heute wesentlichen zwei Modelle für das Parsen von XML verwendet:
 - Simple API für XML (SAX)
 - Document Object Model (DOM)
- JavaXML-library bietet Lösungen für beides an.

25.10.2004

Proseminar SemWeb WiSe04/05

46

Wie verarbeiten wir die Daten?

- Möglichkeit 1: schreibe eine anwendung für den verwendeten Parser.
 - Z.B: lese Dokument als DOM-Baum ein, bearbeite dann den Baum
- Möglichkeit 2: bearbeite das Dokument mittel vorgegebener Regeln
- Hierzu verwendet man sog. Stylesheet-Transformationen (XSL)

25.10.2004

Proseminar SemWeb WiSe04/05

47

Zusammenfassung

- XML an sich ist eine sehr grundlegende Technologie
- Die Stärken liegen in:
 - Der grossen Akzeptanz
 - Der enormen Menge an frei erhältlicher Software
 - Der Vielzahl der Anwendungsbereiche

25.10.2004

Proseminar SemWeb WiSe04/05

48