

Sommersemester 2006 • Institut für Germanistik I

Vorlesung Computerphilologie

Themenfeld Dokument-Management

„Wie kann man Dokumente systematisch und langfristig für die Computernutzung bearbeiten?“

Warum Dokument-Formate?

Dokument-Formate sind den Internet-Formaten verwandte Textauszeichnungen (ohne Hypertext-Eigenschaften). Wozu ist das gut?

- Um Texte zeitbeständig speichern zu können,
- Um Texte im Detail vergleichbar zu machen,
- Um große strukturierte Textmengen leicht zugänglich sammeln zu können,
- Um Formatierungsinformation getrennt vom Inhalt nutzen zu können,
- Um Texte leicht verschiedenen Layoutumgebungen anpassen zu können.

Dokument-Formate

Zur Dokumentstrukturierung:

- SGML und Anwendungen Martif, TEI
- XML mit Beispiel

(Zu TeX siehe unter „Höhere Textverarbeitung“)

Bildformate

- EPS, GIF, JPEG, PICT,
TIFF, BMP

Die Folien über SGML, TEI und XML
hat Dr. C. Vertan vorbereitet

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

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
 - SGML-Bestandteile sind wiederverwendbar.
 - sind medienneutral aufgebaut
 - benötigen weniger Speicherbedarf als konventionelle Texte

Aufbau von SGML-Dokumenten

- Ein vollständiges SGML-Dokument besteht aus 3 Teilen:

<u>SGML-Deklaration</u>
<u>Prolog</u>
Dokument-Instanz



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

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.



Der Prolog und die DTD (**D**okument **T**ype **D**efinition)

- 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:

```
<!ELEMENT anthologie
<!ELEMENT gedicht
<!ELEMENT titel
<!ELEMENT strophe
<!ELEMENT reihe
```

Datei anthologie.dtd:

```
- - (gedicht+)>
- - (titel?, strophe+)>
- O (#PCDATA)>
- O (reihe+)>
O O (#PCDATA) >
```

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

Prolog Beispiel:

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


Eine Dokument-Instanz

<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>

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

Die Syntaxkorrektheit wird
von einem Parser überprüft

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>`

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

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

```

<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

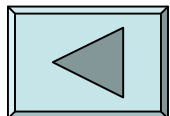
Anwendungsklassen für SGML

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



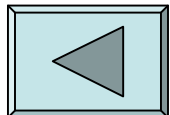
SGML-Dokumente sind plattformneutral

- 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 herstellerspezifischen Speicherformat, speziellen internen Formatierungen oder gar speziellen Hardwarevoraussetzungen basiert.



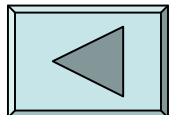
SGML-Dokumente besitzen ein einheitliches Erscheinungsbild

- 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.



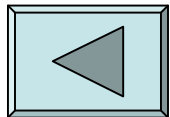
SGML-Dokumente besitzen eine “Checkliste” für den Redakteur

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



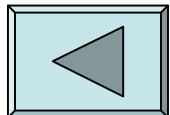
SGML-Dokumente sind leichter recherchierbar

- 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.



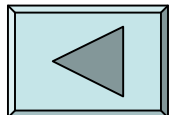
SGML-Bestandteile sind wiederverwendbar

- 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 (Textsorten) zur Erreichung von Einheitlichkeit gezielt wiederverwerten.
- Beispiele: Definitionen in einem Lehrbuch, Ergebnisse von Bundesligaspielen, Loseblattsammlung mit Verordnungen



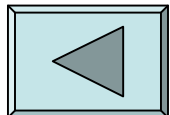
SGML-Dokumente sind trägerneutral aufgebaut

- 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.



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.



Was ist TEI ?

(Text Encoding Initiative)

- Eine philologische Textbeschreibungssprache, basierend auf SGML
- ein Markup-System, um die Struktur von Texten in elektronischem Format zu beschreiben
- legt die Art der Darstellung auf dem Bildschirm oder dem Papier nicht fest

Warum TEI ?

Die Aktualität und Brauchbarkeit elektronischer Editionen wird häufig durch mehrere Faktoren eingeschränkt,

Es galten die typischen Probleme wie auch für SGML, als da sind:

- die grundsätzliche Abhängigkeit von einem bestimmten Programm in einem bestimmten Betriebssystem
- die kurze Lebensdauer der Betriebssysteme oder deren Versionen, unter denen die Darstellungsprogramme laufen
- die kurze Lebensdauer der Programme, die zu ihrer Darstellung verwendet werden,
- Abhängigkeit von der logischen Einrichtung des Datenträgers (auf dem sich die Edition befindet)

TEI Entwicklung

die Entwicklung wurde gefördert von:

- Association for Computational Linguistics (ACL),
- Association for Literary and Linguistics Computing (ALLC)
- Association for Computing and the Humanities (ACH)

Die erste Version erschien 1996

Ziele

- Festlegung der zu kodierenden wissenschaftlichen Textmerkmale, ansonsten normale Ziele von SGML,
- Art der Codierung dieser Merkmale ist so, dass
 - die Daten
 - plattformunabhängig und
 - austauschbar werden
 - wichtige Meta-Informationen nicht verloren gehen.

TEI und SGML

- TEI ist eine SGML-basierte Anwendung
- TEI stellt dem Textwissenschaftler mehrere hundert Auszeichner (tags) zur Verfügung, mit denen man jeden gewünschten Aspekt eines Textes kennzeichnen und mit Zusatzinformation versehen kann
- Daten, die entsprechend den TEI-Richtlinien ausgezeichnet worden sind, können mit jedem Programm, das SGML-Daten verarbeiten kann, dargestellt, umgewandelt oder durchsucht werden.

TEI- Auszeichner

- Es gibt zwei unterschiedliche Auszeichnermengen:
 - eine enthält Elemente, die in den Dokumenten erscheinen können (Grundmenge der tags)
 - die zweite definiert die Elemente im “header” (ähnlich wie ein header in einer e-mail)

TEI Header

- Stellt Information über Nachschlagewerke und Dokumentationen zur Verfügung. Sehr wichtig für
 - Austausch von verschiedenen Quelltexten
 - Langzeit-Benutzer
- 4 Teile:
 - Dateibeschreibung (titel, intellectual responsibility, distribution)
 - Quellenbeschreibung
 - Kodierungsbeschreibung (welches Tag-Set von TEI)
 - “revision history” (Änderungen)

TEI - Grund-Auszeichnermenge

- Die Elemente, die z.B. in einem Wörterbuch erscheinen sind völlig andere als in einen Drama,
- Daher muß man für einen TEI-DTD-Aufbau eine Grund-Auszeichnermenge wählen,
- Zur Zeit sind sechs davon definiert, für Prosa, Lyrik, Drama, Wörterbücher, “transcribed speech” und für terminologische Datenbanken,
- Die Wahl einer Grund-Auszeichnermenge definiert die Struktur aller Dokumente, die diese Menge benutzen,

TEI - Ressourcen

- Die TEI - Webseite :
 - <http://www-tei.uic.edu/orgs/tei>
 - Wichtige Dokumente:
 - “Guidelines for the encoding and interchange of machine-readable texts”
 - “An introduction to the TEI encoding scheme”
- Beispiel für eine TEI-Codierung :
 - “Der Junge Goethe in seiner Zeit” - TEI konforme Fassung des Textes unter:
 - <http://www.jgoethe.uni-muenchen.de/teireadme.html>

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,
 - 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.



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.

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 (Kapiteltitel, Überschriften, usw.)
- Die XML-Spezifikation definiert einen Standard für vereinfachte Markupentwicklung ohne eine komplizierte Deklaration.
- Dokumente können sein:
 - Konventionelles Standarddokument
 - Vektor-Grafik
 - e-commerce-Transaktionen
 - Mathematische Formeln
 - Meta-Daten, usw.

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, Firefox)
- HTML-Tags können innerhalb von XML-Dokumenten benutzt werden.
- XML enthält keinen vordefinierten Tag für Hyperlinks. (dafür muss XML Linking Language - XSL - benutzt werden)
- XML-Dokumente können in HTML-Dateien eingelesen und z.B. mit JavaScript ausgewertet werden.

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 (Cascaded Style Sheets) style.css oder eine spezielle Style-Sprache (XSL) erzeugt.

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>

```

Beispiel.xml

Auswertungstools

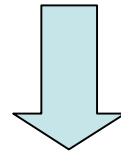
```
%textonly -s ' <sample.xml
In the begining was the world.
```

```
%sgcount <sample.xml
  p 1
  s 1
  w 6
  c 1
```

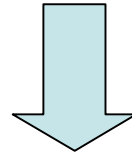
```
%sggrep -q '.*</w[pos="n"]' sample.xml
<w pos='n'>beginning</w>
<w pos='n'>word</w>
```

XML vs. HTML

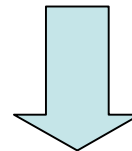
HTML wurde mit SGML definiert



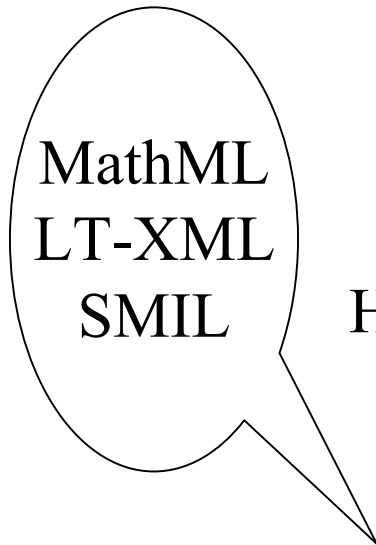
HTML entspricht nicht den XML-Standards



XML-Code kann nicht in HTML-Dokumente eingebaut werden



XHTML



Was ist XHTML ?

- Re-definition 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

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 (textlosen) Tags enthalten wie in XML ein Leeres-Element-Tag
`<hr />` (statt `<hr></hr>`)
- es gibt nur ein einziges `head`- und ein einziges `body`- Element, man nur auch nur ein einziges `frameset` -Element einfügen.
- jedes `head`-Element darf nur ein einziges `title` -Element (Tag) enthalten

XHTML

vs.

HTML (Beispiel)

```
<html>
<head>
<title> Vorlesung CP Content</title>
</head>
<body>
<h1> Vorlesung CP </h1>
<hr / >
<h2>Inhalt</h2>
<ul>
  <li>01 Intro </li>
  <li> 02 Theorie </li>
</ul>
</body>
</html>
```

```
<HTML>
<BODY>
<h1> Vorlesung CP </h1>
<hr>
<h2>Inhalt</h2>
<UL>
  <li>01 Intro
  <li> 02 Theorie
</ul>
</body>
</HTML>
```

Viele HTML-Editoren ergänzen HTML zu XHTML

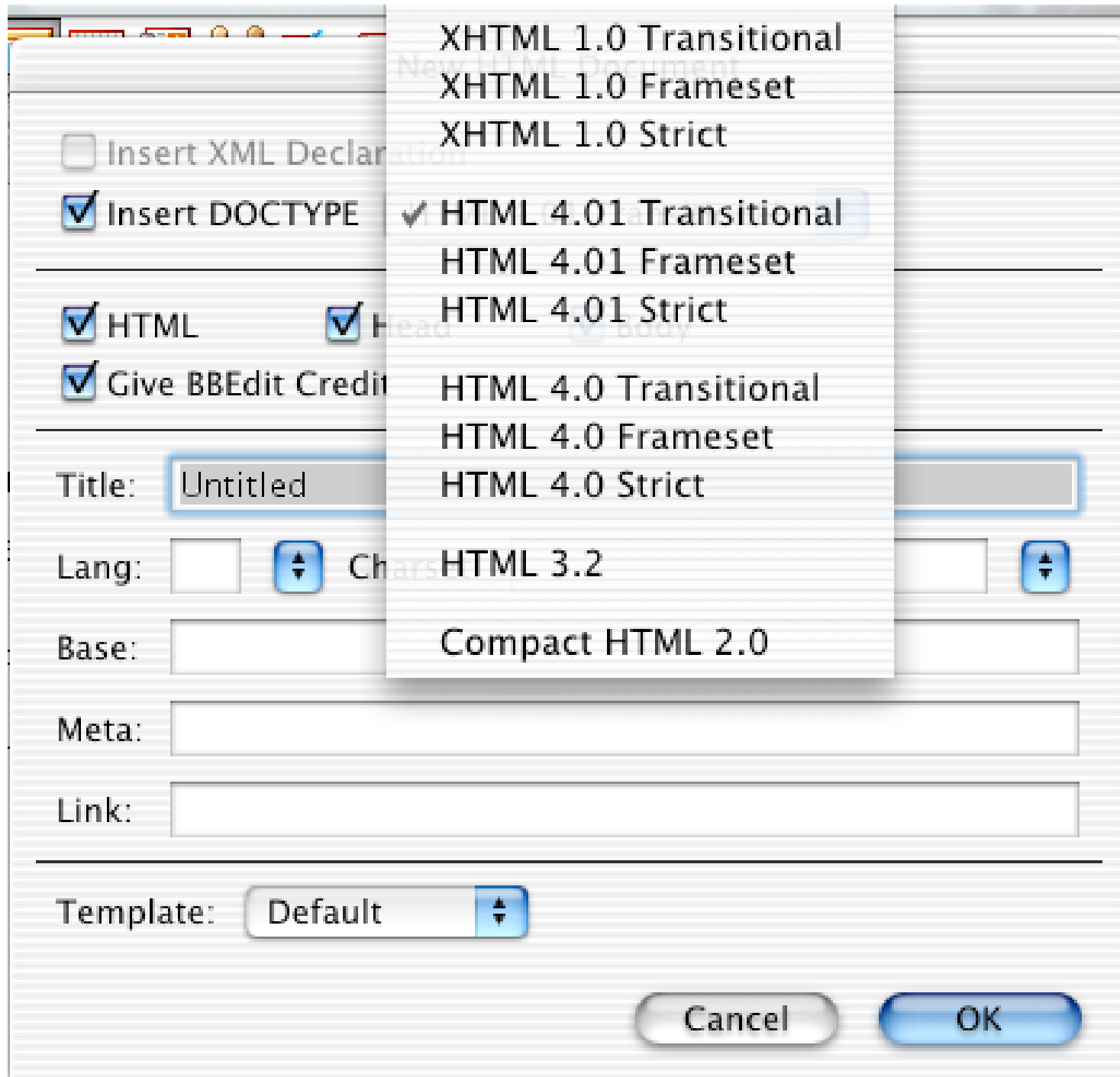
Strictly Conforming XHTML

- 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>`

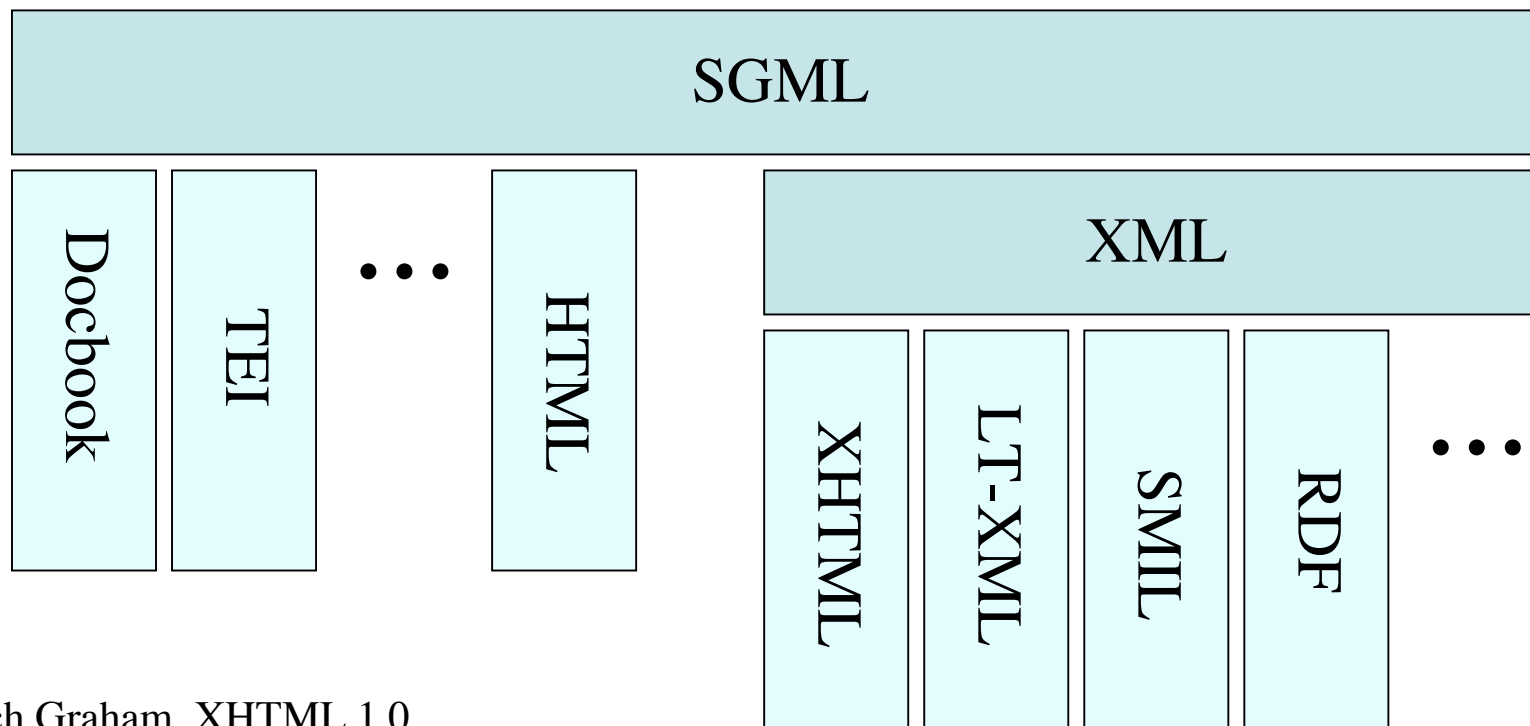
Strictly Conforming XHTML - Beispiel

```
<?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>
```

BBEdit

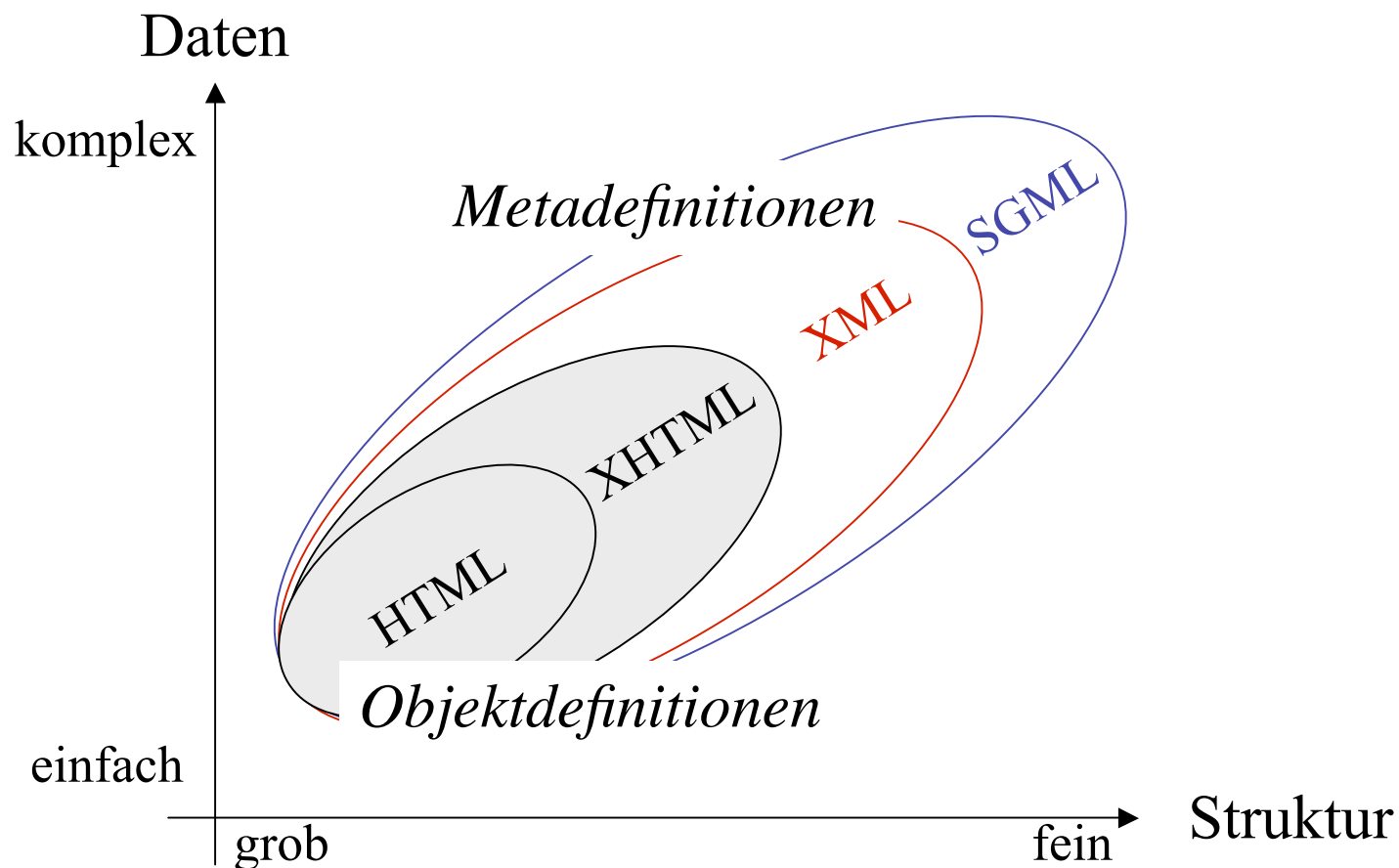


Derivate von SGML und XML



Nach Graham, XHTML 1.0

SGML, XML, XHTML und HTML



Literatur

- “*FrameMaker + SGML 5.5*”, M. Bollenbach, T. Rüppel, A. Rucker, Addison-Wesley, 1999
- “*SGML und XML - Anwendungen und Perspektiven*”, W. Möhr, I. Schmidt (Hrsg.), Elektronischer Publishing, 1999
- “*XML Complete*”, S. Holzner, McGraw-Hill, 1998
- “*XML - Das Einsteigerseminar*”, M. Seeboerger-Weichselbaum, bhv Verlag, 1999
- “*Enabling Structured Documents - A short guide to LT XML*”, H. S. Thompson, ElseneWS 8.2, Juni 1999, S. 4-5
- “[A Gentle Introduction to SGML](#)”, in “[Guidelines for Electronic Text Encoding and Interchange](#)” (Kapitel 2), <http://etext.virginia.edu/TEI.html>
- “[The SGML Primer](#)”, http://www.sq.com/resources/content_sgml_primer.html
- “[What is XML ?](#)” N. Walsh, <http://www.xml.com/pub/a/98/10/guide1.html>
- <http://www.xml.org>
- XHTML 1.0 Language and Design Sourcebook. I.S Graham. Wiley 2000.
- Als kritische Stellungnahme siehe:
- <http://cliki.tunes.org/TUNES%20vs%20the%20WWW>

Bildformate

- <http://www.bildformate.de/>

Beschreibt alle bekannten Bildformate „für den Hausgebrauch“

Dazu gehören die für uns interessantesten:

BMP	das Standardformat unter Windows
EPS	Eingekapseltes Postskript Format
GIF	Graphic Interchange Format
JEPG	Joint Pictures Experts Group
PICT	Standard-Grafikformat von Apple Macintosh
TIFF	Tagged image file format

EPS (.eps)

In .ps Dateien

<http://www.bildformate.de/>

Schreibt:

Kurzbeschreibung	Weitere Beschreibung
EPS-Daten können CMYK-, Grau- oder Strichdaten sowie Komprimierungen und Vectordaten enthalten. Sie enthalten PostScript-Kode, der unabhängig vom Ausgabegerät alle erforderlichen Daten zur Beschreibung einer Seite oder einer Darstellung enthält. Viele Text- oder Grafikprogramme können EPS-Daten lesen, aber nicht weiter bearbeiten.	Je nach Anwendung werden beim Erstellen auch "Macintosh Resource Forks" erzeugt, in denen sich ein 72 dpi-Übersichtsbild befindet, da nur wenige Programme das enthaltene Postscript-Format interpretieren können.

GIF (.gif)

Internet-geeignet

<http://www.bildformate.de/>

Schreibt:

Kurzbeschreibung	Weitere Beschreibung
<p>Datentiefe beliebig. (maximal 256 Farben) Hauptanwendung liegt im Internet. Ein Bild, das z.B. 28 Farben verwendet wird mit 5 Bit Farbtiefe (=maximal 32 Farben) anstatt in 8 Bit Farbtiefe gesichert. Durch diese Option werden die Bilddateien etwas kleiner. Die Bildzeilen können hintereinander oder interlaced gesichert werden. Das Sichern im Interlaced-Modus ist nur sinnvoll, falls die GIFs in Online- Diensten verwendet</p>	<p>GIF-Version 87a Diese Version kann von allen Programmen importiert werden, die GIF-Daten verarbeiten können. Unterstützt KEINE Kommentare und KEINE Transparenz.</p> <p>Version 89a Dies ist die aktuelle Version des GIF-Formates. Hier sind auch Kommentare und Transparenzen möglich.</p>
<p>werden.</p>	

JEPG (.jpg)

<http://www.bildformate.de/>

Schreibt:

Oft praktisch, erzeugt
kleine Dateien

Kurzbeschreibung	Weitere Beschreibung
Standard-Kompressionsverfahren für digitale Bilder. Die JPEG-Kompression ist ein verlustbehaftetes Kompressionsverfahren.	eine niedrigere Qualität führt zu einer kleineren Dateigröße oder aber stärkerer "Kachelung" der Bilder

PICT (.pict)

<http://www.bildformate.de/>

Schreibt:

Kann auf Mac beliebig eingesetzt werden

Kurzbeschreibung	Weitere Beschreibung
PICT's können Pixel- und Vectordaten enthalten. Meist sind die Daten komprimiert abgelegt	Macht auf anderen Rechnern manchmal Probleme

TIFF (.tif)

Ist eine Art „Rohformat“
und erzeugt meist riesige Dateien

<http://www.bildformate.de/>

Schreibt:

Kurzbeschreibung	Weitere Beschreibung
<p>Ein verbreitetes Datenformat für Bilddaten. Innerhalb des TIF-Formats gibt es verschiedene Klassifizierungen, die in der Datei selbst gespeichert sind. Beim Öffnen eines TIFF-Bildes wird zuerst dieser Datenteil gelesen und interpretiert</p>	<p>Motorola(Mac)- oder Intel(PC)-Format. Falls häufig TIFFs mit DOS-Benutzern ausgetauscht werden, sollte Intel verwendet werden, da einige DOS-Programme nur Intel-TIFFs einlesen können.</p>

Formatkonvertierung

Fast alle Bildbearbeitungsprogramme können eingelesene Dateien in einem anderen Format wieder speichern. Daher ist das die einfachste Methode der Dateiwandlung:

- Bildbearbeitungsprogramm (z.B. sehr geeignet: GraphicConverter, nur auf Macintosh) starten
- Unter „Öffnen“ die Bilddatei laden
- Unter „Speichern als“ die Datei in anderem Format wieder ablegen.

Wenn man allerdings einmal in einem verlustbehafteten Format (etwa JPEG) speichert, kann man die ursprüngliche Information durch Wandeln in z.B. TIFF nicht mehr zurück erhalten.