WWW – Grundlagen und Technologie

SGML – HTML

Erik Wilde

Übersicht

- Markup-Sprachen Grundlagen
- *Standard Generalized Markup Language (SGML)*
  - Grundlage für HTML und XML (auf anderen Ebenen!)
- *Hypertext Markup Language (HTML)*
  - einfaches Beispiel und syntaktischer Aufbau
  - Formulare auf HTML-Seiten
  - neue Features von Forms in HTML 4.01
  - Clickable Images
  - Frames
- *Web Accessibility Initiative (WAI)* des W3C
- Zusammenfassung

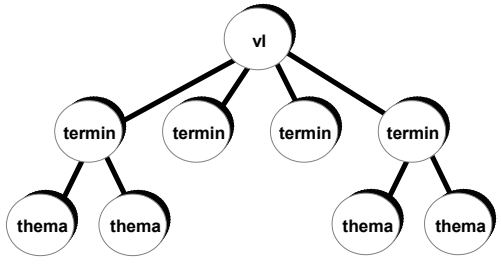
Inhalt und Präsentation (I)

• Vorlesungs-Titel	World Wide Web	• Fett, zentriert, 24pt
• Vorlesungs-Beschreibung	Das *World Wide Web (WWW)* ist ein verteiltes, hypermediales Dokument.	• Abstand, 16pt
• Kapitel, Titel	1. Einführung	• Abstand, Nummer, 20pt
• Absatz	In der Einführung werden das Semesterprogramm und allgemeine Grundlagen zu Internet und Hypermedien besprochen.	• Abstand, 16pt
• Unterabschnitt, Titel	Semesterprogram	• Abstand, Nummer, 18pt
• Absatz	Das Programm der Vorlesung teilt sich in gesamt 14 Termine auf, die einen jeweils in sich abgeschlossenen Aspekt des WWW besprechen.	• Abstand, 16pt
• Unterabschnitt, Titel	Grundlagen des Internet	• Abstand, Nummer, 18pt
• Absatz	Das Internet bietet die technische Infrastruktur, auf der das WWW aufsetzt. Es bietet unter anderem Dienste zum Datentransport und zur Benennung von Rechnern.	• Abstand, 16pt

Inhalt und Präsentation (II)

- Inhalt eines Dokumentes
  - Menge von logischen Einheiten
  - unabhängig von der Präsentation
  - Konzepte wie Absatz, Überschrift, Kapitel, Teil, ...
- Präsentation eines Dokumentes
  - Menge von Layout-Einheiten
  - Resultat einer Formatierung des Dokumentes
  - viele mögliche Präsentationen eines Dokumentes
  - Konzepte wie Block, Seite, Abstand, Schrifttyp, ...
- abhängig von Umgebung und Anforderungen
  - *Festlegung* der Präsentation durch den Autor
  - *Freiheiten* für die Präsentation, Autor liefert Inhalt

Document Tree



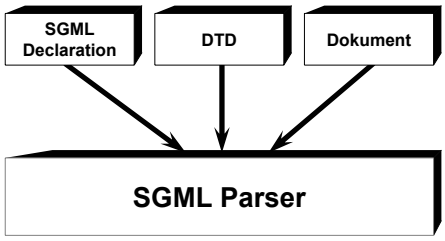
SGML

- *Standard Generalized Markup Language (SGML)*
- definiert in ISO Standard 8879 (1986)
- Markup Language trennt Struktur und Text
  - Struktur ist syntaktisch identifizierbar
  - Text ist an definierten Stellen der Struktur erlaubt
- Vorteile einer Markup Language
  - einfache Repräsentation (Erzeugung "von Hand")
  - Austauschbarkeit als rein textbasierte Dokumente
- Definition beliebiger Dokumenttypen
  - anwendungsspezifische Dokumenttypen
  - gemeinsamer zugrundeliegender Mechanismus
  - gemeinsam verwendbare Software

Aufbau von SGML

- die *SGML Declaration*
  - konkrete Syntax
  - SGML Features (wie *Markup Minimization*)
- die *Document Type Definition (DTD)*
  - erlaubte Elemente
  - Kombination der erlaubten Elemente
  - Attribute der erlaubten Elemente
  - Abkürzungen (*Entities*), z.B. für Sonderzeichen
- das *Dokument* selber
  - Struktur des Inhaltes gemäss der DTD
  - Inhalt der Struktur (reiner Text)

SGML Parser



SGML Document

- Instanz eines bestimmten Dokumententyps
  - Kennzeichnung zu Beginn des Dokuments
  - kann nur mit Hilfe der DTD interpretiert werden
- Einhaltung der Regeln der SGML Declaration
- Einhaltung der Regeln der DTD
- stellt einen *document tree* dar
- Erstellung eines SGML Dokumentes
  - als Textdatei (ursprüngliches Modell)
  - mit SGML-Tools (zunehmend verbreitetes Modell)

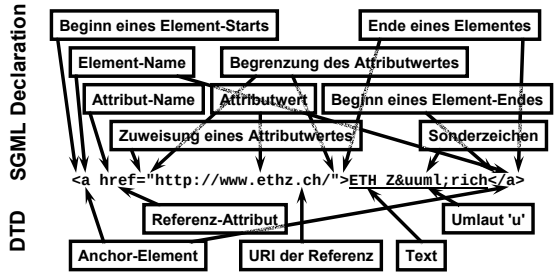
SGML Document Type Definition

- Festlegung einer Grammatik
  - bestimmt die *Wörter* einer Sprache
  - bestimmt die *Regeln zur Satzbildung*
- Definition der Elemente
  - Elementnamen (frei wählbar)
  - Attribute, Attributtypen und -werte
- Definition zur Kombination der Elemente
  - Vorkommen der Elemente in einem Dokument
  - *Model Groups*: `<!ELEMENT UL (LI)+ >`
  - *Exceptions* (*Inclusions*|*Exclusions*)

SGML Declaration

- bezieht sich auf mehrere Dokumente
- Festlegung der konkreten Syntax
  - SGML selber verwendet abstrakte Syntax
  - Zeichen mit Sonderbedeutung
  - HTML: `< > </ <! <? = " & ;`
- Festlegung von Zeichensätzen
- Festlegung von Kapazitäten
  - Länge von Namen, Schachtelungstiefen, ...
- SGML Features
  - *tag omission* (Weglassen von Tags)
  - *short tags* (Abkürzen von Tags)

SGML-Teile des HTML Standards



HTML und Verwandte

- *Hypertext Markup Language (HTML)*
  - festgelegter Dokumententyp
  - basierend auf allgemeiner Sprache (SGML)
- *Standard Generalized Markup Language (SGML)*
  - Mechanismus zur Definition von HTML
  - erlaubt Definition beliebiger Dokumententypen
  - HTML ist eine Anwendung von SGML
- *Extensible Markup Language (XML)*
  - Web-spezifisches Profile von SGML
  - leicht vereinfachte Version (weniger kompliziert)
  - keine Einschränkung der Allgemeinheit
  - optimiert auf Anwendbarkeit auf dem Web

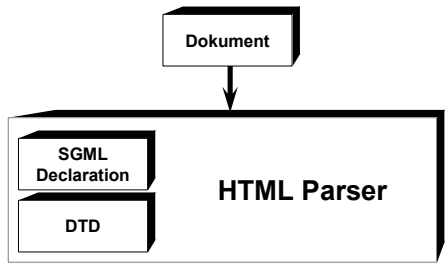
Hypertext Markup Language

- Anwendung (*Application*) von SGML
- HTML *SGML Declaration*
  - bestimmt die konkrete Syntax von HTML
  - bestimmt die SGML Features von HTML
- HTML *SGML Document Type Definition (DTD)*
  - drei verschiedene Varianten der DTD
  - Definition der Elemente und Attribute
- zusätzliche Definitionen (kein SGML!)
  - Einschränkungen von Attributwerten
  - Bedeutungen von Elementen und Attributen
  - Grossteil des Umfangs der HTML-Spezifikation

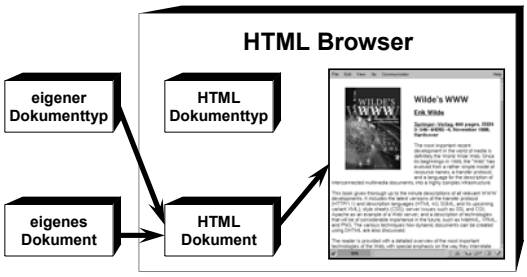
HTML als SGML Anwendung

- jeder Browser implementiert HTML Parser
  - mehr als 90% aller HTML Seiten sind kein SGML
  - Browser implementieren fehlertolerante Parser
  - kein Browser ist eine formal korrekte SGML Implementierung
  - HTML ist mehr als formal korrektes SGML
- Verbreitung generierter HTML Seiten
  - HTML Editoren (oftmals "eigenwilliges" HTML)
    - nicht unbedingt sinnvolle Wahl der Elemente
    - Verwendung eigener HTML-Erweiterungen
  - Generierung durch Skripte oder Programme
    - on-line bei der Abfrage (PHP, ASP, JSP, ColdFusion)
    - off-line bei der Generierung von Web-Sites

HTML Parser



Publishing mit HTML



HTML Validierung ist schwierig!

- keine guten Daten verfügbar bzw. veröffentlicht
- einfache Validierung nach DTD ungenügend
- korrektes SGML, aber kein korrektes HTML
  `<table><tr><th colspan="zwei"> … </table>`
- korrektes HTML aber kaum zu validieren
  `<html><head><title>Beispielseite</title>`
  `<access author="dret" permission="640">`
- HTML ist zu schwierig zu verarbeiten
  - zu tolerant gegen "Missbrauch"
  - als Folge werden die Tools immer nachlässiger
  - Konsequenz: der *Internet Engineering* Ansatz leidet

HTML und Internet Engineering

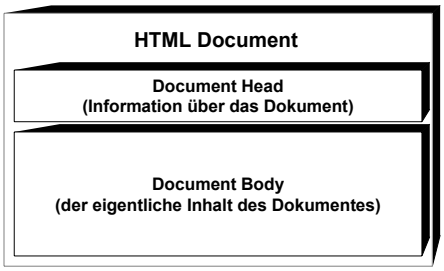
"Be liberal in what you accept and be rigorous in what you produce."

- Kompatibilität muss in zwei Richtungen gehen!
- Rückwärtskompatibilität (oft beachtet)
  - neue Applikationen mit alten Daten
  - erfordert Cleverness beim Design der Applikationen
- Vorwärtskompatibilität (oft ignoriert)
  - neue Daten mit alten Applikationen
  - erfordert Cleverness beim Design der Daten

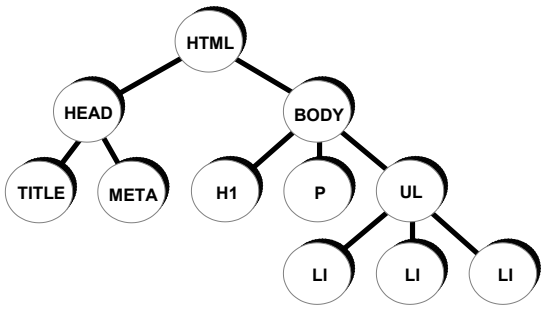
Internet Engineering at work

- HTML ist erweiterbar
  - unbekannte Elemente müssen ignoriert werden
  - unbekannte Attribute müssen ignoriert werden
  - HTML 4.01 geht (fast...) als HTML 3.2 durch
- CSS ist erweiterbar
  - unbekannte Selektoren werden ignoriert
  - unbekannte Properties werden ignoriert
- eines der Axiome von Web-Standards!
- Problem sind eher falsche Implementierungen
  - Feature wird nicht ignoriert, sonder erzeugt Fehler
  - Regeln zum liberalen Interpretieren beachten!

Aufbau einer HTML-Seite (I)



Aufbau einer HTML-Seite (II)



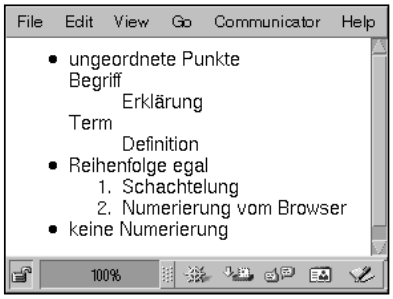
Elemente des Document Head

- Dokumententitel: **<TITLE>**
  - üblicherweise als Fenstertitel und in Listen
- Basisadresse: **<BASE>**
  - wichtig für die Interpretation relativer URIs
- Links zu anderen Ressourcen: **<LINK>**
  - für Beziehungen zwischen Ressourcen
- Style Sheets: **<STYLE>**
- Scripte für Scripting Languages: **<SCRIPT>**
- Meta-Information: **<META>**

Elemente des Document Body

- Überschriften (auf verschiedenen Ebenen)
- Absätze (normaler Fliesstext)
- verschieden formatierter Text
- Listen (numeriert und unnumeriert)
- Verweise auf andere Ressourcen (Hyperlinks)
- horizontale Begrenzungslinien
- Tabellen
- Graphiken (in verschiedenen Formaten)
- allgemeine externe Objekte (z.B. Applets)
- Formulare (im Browser auszufüllen)

Formatierung von Listen



Beispiel für HTML-Listen

```
<UL><LI>ungeordnete Punkte
<DL><DT>Begriff<DD>Erkl&auml;rung
<DT>Term<DD>Definition
</DL>
<LI>Reihenfolge egal
<OL><LI>Schachtelung
<LI>Numerierung vom Browser
</OL>
<LI>keine Numerierung
</UL>
```


Listen in HTML

- gutes Beispiel für Inhalt statt Layout
- ungeordnete Listen: **<UL>** (unordered lists)
  - TYPE-Werte DISC, SQUARE oder BULLET
  - einzelne Punkte mit **<LI>**
- geordnete Listen: **<OL>** (ordered lists)
  - Darstellung mit TYPE 1, a, A, i oder I
  - Festlegung der Numerierung mit START
  - einzelne Punkte mit **<LI>**
- Definitionslisten: **<DL>** (definition lists)
  - der Begriff wird in **<DT>** eingeschlossen
  - die Definition wird in **<DD>** eingeschlossen
  - Formatierung je nach Browser unterschiedlich
- es sollten Style Sheets verwendet werden

Tabellen

- Folge von Zeilen, Folge von Zellen
  - **<TABLE>**, **<TR>**, **<TH>**, **<TD>**
- Layoutbeeinflussung durch Attribute
  - ALIGN, ersetzbar durch **<DIV>**
  - Umrandung von Tabelle und Zellen mit **BORDER**
  - Breite in Pixel oder Prozent (**WIDTH**)
  - Abstand der Tabellenzellen (**CELLSPACING**)
  - Abstand Zelleninhalt und Border (**CELLPADDING**)
- Positionierung des Inhaltes der Zellen
  - **ALIGN** und **VALIGN** für **<TR>**, **<TH>** oder **<TD>**
- **COLSPAN** und **ROWSPAN** (**<TH>**, **<TD>**)

Tabellen in HTML 4.01 vs. 3.2

- Ausrichtung an Zeichen (z.B. Dezimalpunkte)
  - ermöglicht die Formatierung mit weniger Spalten
- bessere Ausrichtung innerhalb der Cells
- mehr Varianten für Borders
- Strukturierung in *Head*, *Body*, *Foot*
  - bessere Formatierung beim Scrollen oder Drucken
- Strukturierung in *Column Groups*
  - bessere Lesbarkeit und nicht-graphische Darstellung
- keine Unterstützung in *CSS1*, aber in *CSS2*

HTML Forms

- ermöglichen die Eingabe von Daten
- brauchen Datenübermittelung (HTTP/CGI)
- mit dem <FORM> Element spezifiziert
  - ACTION="URI" bestimmt die Verarbeitung
  - METHOD="GET/POST" bestimmt HTTP Methode
- für Eingaben werden Elemente verwendet
  - <INPUT> definiert viele Eingabemöglichkeiten
  - <TEXTAREA> für mehrzeiligen Text
  - <SELECT> definiert Optionslisten
- andere HTML-Elemente sind ebenso erlaubt

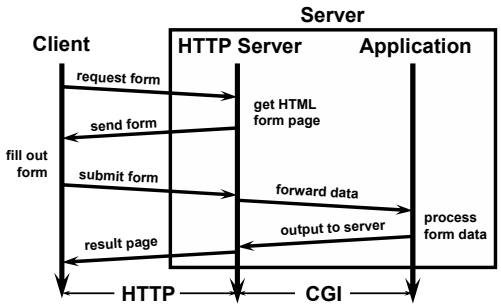
Verwendung eines Forms

/cgi-bin/form.pl?A=H&N=Wilde&V=Erik

Forms in HTML

```
<FORM ACTION="http://w3.org/cgi-bin/form.pl" METHOD="GET">
  <TABLE>
  <TR>
      <TD COLSPAN="2">
          <INPUT TYPE="RADIO" NAME="A" VALUE="H"> Herr
          <INPUT TYPE="RADIO" NAME="A" VALUE="F"> Frau
  <TR>
      <TD>Name:
      <TD><INPUT TYPE="TEXT" NAME="N" SIZE="20">
  <TR>
      <TD>Vorname:
      <TD><INPUT TYPE="TEXT" NAME="V" SIZE="20">
  </TABLE>
  <INPUT TYPE="SUBMIT" VALUE="Absenden">
</FORM>
```


Form Handling



HTML 4.01 Features von Forms (I)

- TABINDEX wählt Reihenfolge der Felder
  - für tastaturgesteuerte Eingabe in Forms
- ACCESSKEY wählt Tastatur-Abkürzungen
  - werden vom Browser hervorgehoben angezeigt
- DISABLED macht Elemente insensitiv
  - Verwendung für konsistentes Interface-Design
- TITLE enthält Hilfs-Texte und Erläuterungen
  - wird vom Browser angezeigt
- ONCHANGE für Skript zur Eingabeprüfung
  - Felder können schon beim Client getestet werden
- ACCEPT für Limitierung erlaubter File-Typen
  - nur für <INPUT FILE> definiert

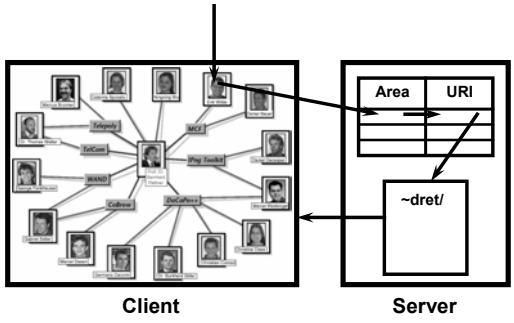
HTML 4.01 Features von Forms (II)

- <FIELDSET> für strukturierte Forms
  - noch nicht allgemein unterstützt (neu in HTML 4.0)
  - Gruppierung nach Themengruppen
  - besonders nützlich für nicht-visuelle Browser
- <LEGEND> für Benennung der Bereiche
  - Text oder andere Inline Elemente
  - sollte auch ohne <FIELDSET> sinnvoll sein
  - z.B. "Angaben zur Person" und "Bestellung"
- Ausnutzung der HTML-Konvention
  - Ignorieren unbekannter Elemente und Attribute
  - trotzdem Darstellung des Inhaltes

Clickable Images (Image Maps)

- Zuordnung von Links zu Bildbereichen
- ermöglicht "graphische" Interfaces
- verbleibende Limitierungen (siehe DHTML…)
  - keine dynamischen Menüs (pop-up, scroll-down)
- Design der Benutzungs-Oberfläche als Bild(er)
- Zuordnung sensitiver Bereiche zu Bildelementen
  - einfache geometrische Bereiche
  - URLs, die zu diesen gehören
- können auch für Forms verwendet werden

Server-Side Image Map

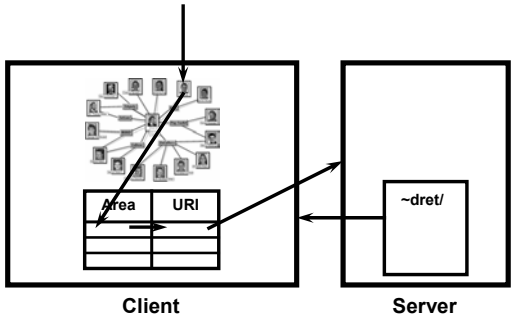


Server-Side Image Map - HTML

```
<A HREF="http://www.ethz.ch/cgi-bin/picture">
<IMG ALT="Dienste-Auswahl" ISMAP="ISMAP"
 SRC="dienste.gif" WIDTH="200" HEIGHT="30"></A>
```

- Script zur Verarbeitung der Koordinaten
  - Cient sendet *query URI* "...bin/picture?153,21"
- oft Map-Verarbeitung in den Server integriert
  - typischerweise **.map** Files mit Areas und URIs
  - Scripte sind langsam und fehleranfällig, besser Servermechanismus verwenden
- "funktioniert" auch mit alten Browsern
  - senden aber keine Koordinaten an den Server

Client-Side Image Map



Client-Side Image Map - HTML

```
<IMG SRC="img003.gif" USEMAP="#Objmap"
  WIDTH="500" HEIGHT="370" BORDER="0">

<MAP NAME="Objmap">
<AREA SHAPE="POLYGON"
  COORDS="87, 261, 405, 261, 405, 281, 87, 281"
  HREF="http://www.tik.ee.ethz.ch/~dret/WWW-VL/" >
</MAP>
```

- Beispiel aus generierten Frames (PowerPoint)
- Map-Verarbeitung in den Client integriert
- ermöglicht neue Features (ändernde Pointer)
- "funktioniert" auch mit alten Browsern
  - alternative Navigationsmöglichkeiten anbieten

Fenster im Fenster

Verwendung mehrerer Fenster

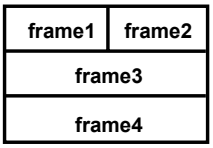
- Zuordnung eines Links zu einem Fenster
- ermöglichen das Erzeugen neuer Fenster
  - **<A HREF="Doku" TARGET="Win1">Link</A>**
  - **<BASE TARGET="Fenster2">**
  - **<FORM TARGET="Fenster3">**
  - reserviert: **_blank**, **_self**, **_parent**, **_top**
- Frames ermöglichen das Unterteilen eines Fensters ("Fenster im Fenster")

```
<FRAMESET COLS="25%,75%">
   <FRAME SRC="Inhalt.html">
   <FRAME SRC="Pre.html" NAME="Text">
</FRAMESET>
```

Darstellung von Framesets

- Schachtelung möglich

```
<FRAMESET ROWS="33%,33%,33%">
 <FRAMESET COLS="50%,50%">
  <FRAME SRC="" NAME="fr1">
  <FRAME SRC="" NAME="fr2">
 </FRAMESET>
 <FRAME SRC="" NAME="fr3">
 <FRAME SRC="" NAME="fr4">
</FRAMESET>
<NOFRAMES><BODY>
...contents to display in
non-frame-capable user agent...
</BODY></NOFRAMES>
```



Probleme mit Frames

- lange Zeit kein offizielles HTML
- Darstellung zum Teil problematisch
  - unübersichtlich durch Scrollmöglichkeiten
  - insbesondere Darstellung auf kleineren Bildschirmen
- Probleme mit Bookmarks
  - dynamische Seitenstruktur, kein fester Status
- Links auf eigene Seiten schwer möglich
- Probleme mit Search Engines
  - Framesets werden meistens ignoriert
  - Seiten auch ohne Frames zugänglich machen
- Fazit: Frames mit Bedacht verwenden

Web Accessibility Initiative (WAI)

- Accessibility heute stark unterbewertet
  - Überschneidungen mit dem Thema Usability
  - Usability kommerziell wesentlich wichtiger
- Accessibility und Usability sind wichtig
  - wichtig aus Sicht der Benutzer
  - wichtig aber auch aus Sicht der Anbieter
- verschiedenen WAI Guidelines
  - *Web Content Accessibility Guidelines*
  - *Authoring Tool Accessibility Guidelines*
  - *User Agent Web Content Accessibility Guidelines*
- Jakob Nielsen's **`useit.com`** als guter Einstieg

Zusammenfassung

- im WWW verwendete Markupsprachen
  - *HTML* als Sprache für Web-Pages
  - *SGML* als Grundlage von HTML
  - *XML* als WWW-Variante von SGML
- grundlegender Aufbau von HTML
- Elemente des Document Head
- Elemente des Document Body
  - Tabellen, Formulare, Images, Frames, Image Maps