



3 Hypermedia

Hypertext



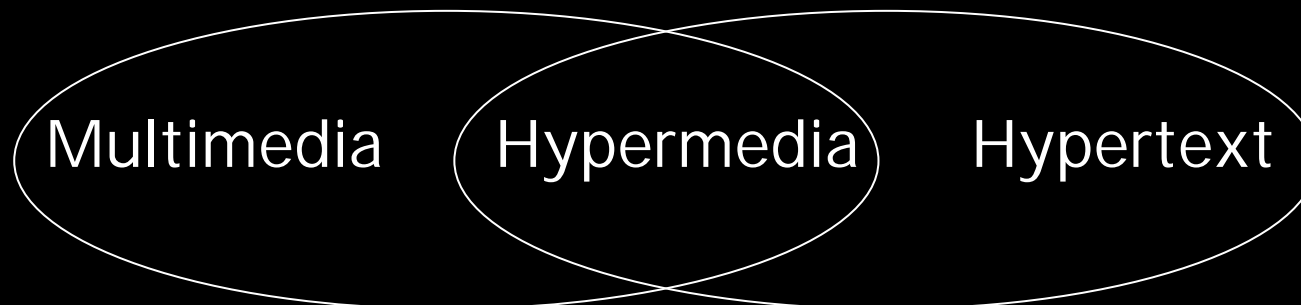
- Ein Organisationskonzept für Information
- Motivation für Hypertext
 - Wissen besteht aus vernetzten Informationseinheiten, nicht-linear
 - Verfassung von Dokumenten: Linearisierung des zu vermittelnden Wissens
 - Lesen: Delinearisierung zur Reproduktion des Wissens
→ Navigation



Hypertext - Hypermedia



- Hypertext-Dokumente
 - Die inhärente Vernetzung der Information soll im Dokument aufrechterhalten bleiben
- Hypermedia
 - In den Dokumenten sind kontinuierliche Medientypen enthalten



Hypertext Wurzeln



- Memex
 - Vannevar Bush: "As we may think", 1945
 - beschreibt "Memory Expander"-Maschine
 - assoziative Speicherung/Zugriff auf Information
 - persönliche Annotationen
- Xanadu
 - Ted Nelson, 1965/1981
 - Begriff Hypertext
 - Docuverse: globales Hypertext-System, Pay per View
- Augment/NLS (oNLine System)
 - Douglas Englebart, 1968
 - Shared Hypertext Document Spaces



Hypermedia Wurzeln



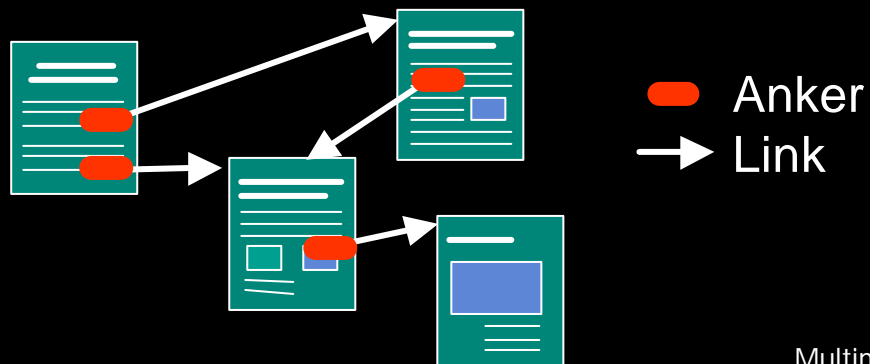
- Aspen Movie Map
 - Andrew Lippman, MIT
 - Virtuelle Reise durch Aspen
 - Bilder von allen Straßen wurden im Abstand von 3 Metern aufgenommen
 - Speicherung auf einer Videodisk
 - Steuerung mit einem Joystick
 - Maximale Fahrgeschwindigkeit: 2 Bilder/Sekunde (entspricht 21,6 km/h)
 - vgl. www.muenchen4d.de



Hypermediakomponenten



- Struktur
 - Ein Hypermedia-Dokument ist ein gerichteter Graph
- Komponenten
 - Die Knoten sind die Informationseinheiten
 - Ein Anker ist ein Informationsfragment innerhalb eines Knotens, an dem ein Link anknüpft
 - Links sind Verweise zwischen den Knoten



Knoten



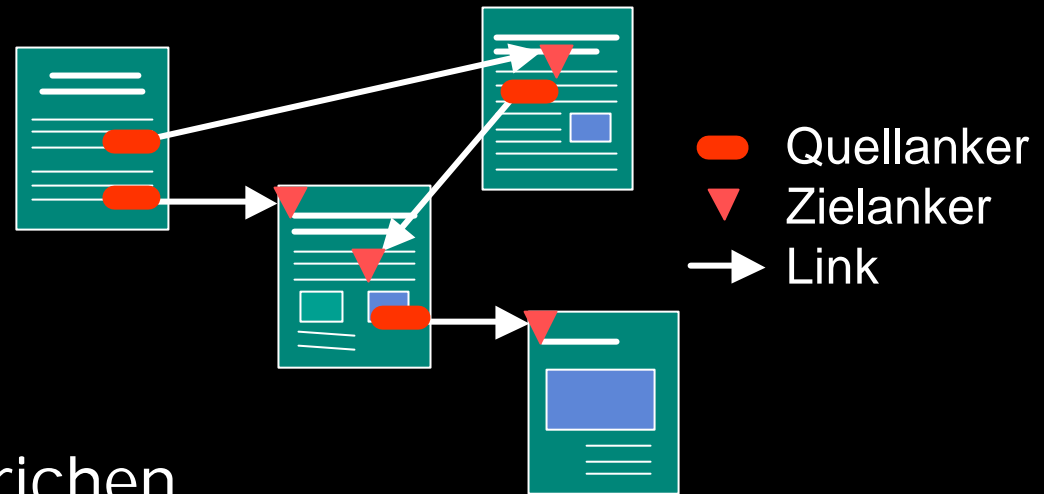
- Single Media Knoten
 - nur ein Medientyp pro Knoten
- Mixed Media Knoten
 - mehrere Medientypen pro Knoten möglich
 - Alternativ, Kombiniert
- begrenzte Inhaltsmenge
 - z.B. HyperCard
- unbegrenzte Inhaltsmenge
 - Möglichkeit zur internen Navigation notwendig
z.B. Scrolling



Anker



- Ankertypen
 - Quellanker
 - Zielanker
- Darstellung als
 - Button
 - Icon
 - Text, z.B. unterstrichen
 - Hidden
 - animiert
 - ...



- Darstellung von Quellanker meist in Verbindung mit Link
- Darstellung von Zielanker meist „Hidden“



Hypermedia Systemklassifikation



Knoten

Anker innerhalb von Knoten

Links zwischen Ankern

Standardattribute für Knoten, Anker und Links

Hypermediasystem
1. Ordnung

benutzerdefinierte Knoten- und Linktypen
benutzerdefinierte Attribute

Hypermediasystem
2. Ordnung

abstrakte Knoten- und Linkstrukturen
Darstellung und Manipulation der
Knoten- und Linksemantik

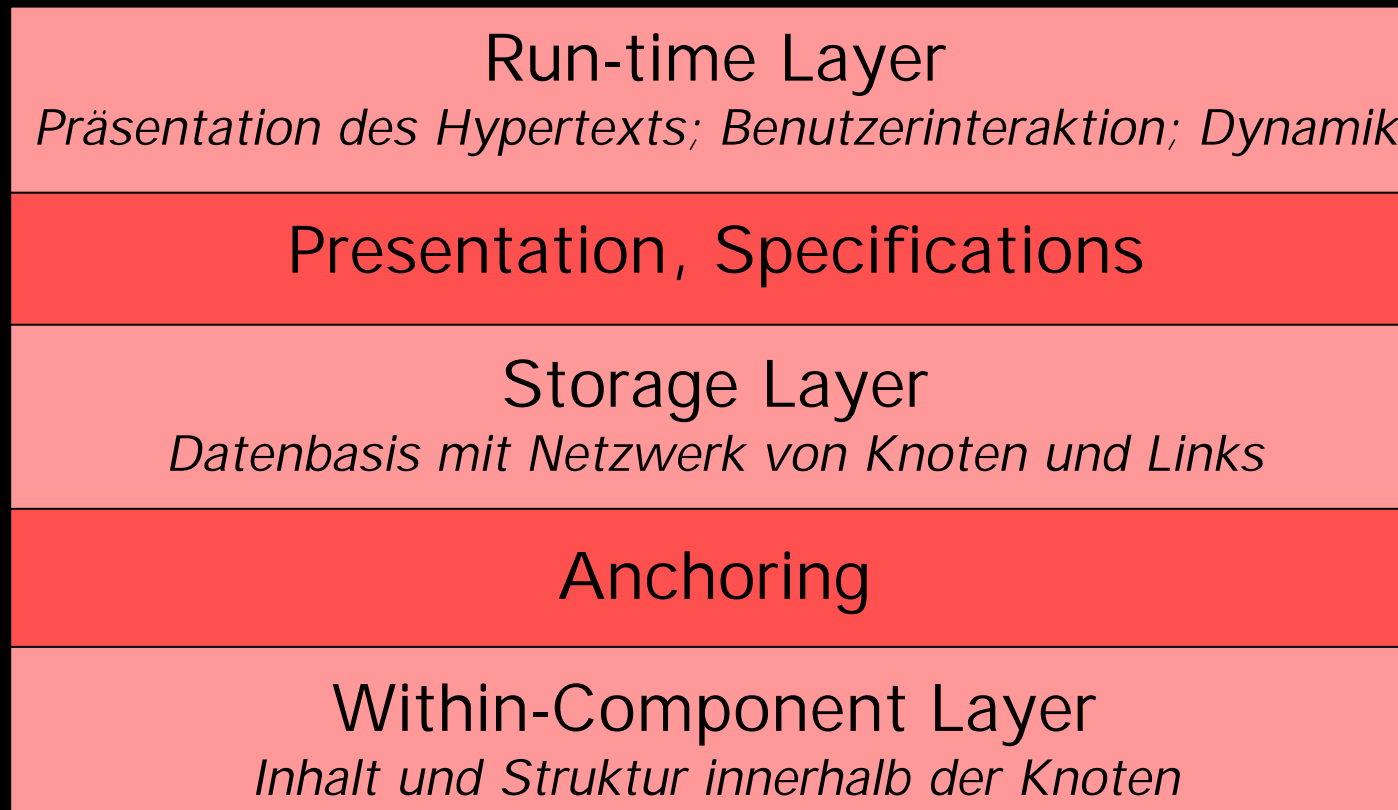
Hypermediasystem
3. Ordnung



Modelle von Hypermediasystemen



- Dexter Hypertext Reference Model
 - Halasz und Schwartz, 1994



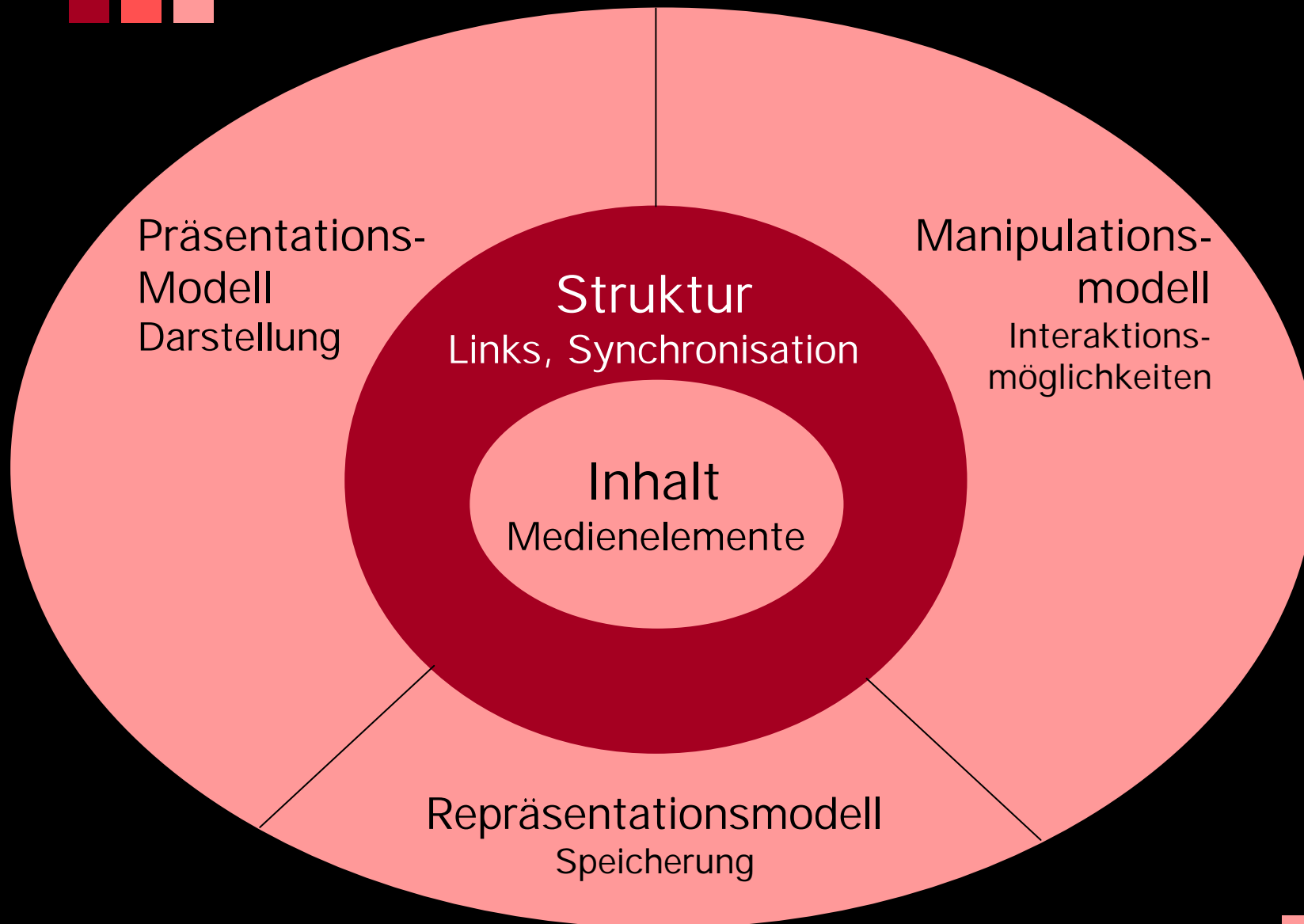
Modelle von Hypermediasystemen



- AHM: Amsterdam Hypermedia Model
 - Erweiterung der bisherigen Modelle um Realtime Multimedia
 - Hardman, Bulterman, van Rossum, 1993/94
 - Dexter-Modell plus Zeit und Kontextattribute
 - Mehrfache Ströme entlang einer Zeitlinie
 - Links sind nicht nur Verweise, sondern können auch Synchronisationskanten sein



Verallgemeinertes Modell



Dokumentsprachen - Markup



- Ursprung des Markup liegt im "Computerized Typesetting"
 - Typesetting-Makros eingebettet in ASCII
 - Direktiven für das Layout (Beispiel T_EX)
 - Presentational Markup
 - Anweisung zur Formatierung
- Semantisches Markup
 - Autoren versehen Inhalte mit vereinbarten Annotationen - Verleger entscheiden über das Layout
 - deskriptives Markup
 - beschreibend, nicht vorschreibend
 - Markup unterstützt die Suche in Dokumenten
 - Markup für Maschinen vs. Markup für Menschen



Markup - ISO-Definitionen



- markup
 - text that is added to the data of a document in order to convey information about it
- descriptive markup
 - markup that describes the structure and other attributes of a document in a non-system-specific way, independently of any processing that may be performed on it
- processing instruction (PI)
 - markup consisting of system-specific data that controls how a document is to be processed



Descriptive Markup - Processing Instruction



- Descriptive Markup
 - editoriale Struktur, inhaltliche Struktur
- Processing Instructions
 - Layout
- Beispiel

```
< kopf spalten= "1" >  
< titel> Vorlesungsankündigung </titel>  
</kopf>  
< rumpf spalten= "2" >  
  < absatz> In der Vorlesung  
    < vorlesung> < kursiv> Multimediasysteme </kursiv>  
    </vorlesung>  
  wird < dozent> Michael Weber </dozent> ...  
  </absatz>  
</rumpf>
```



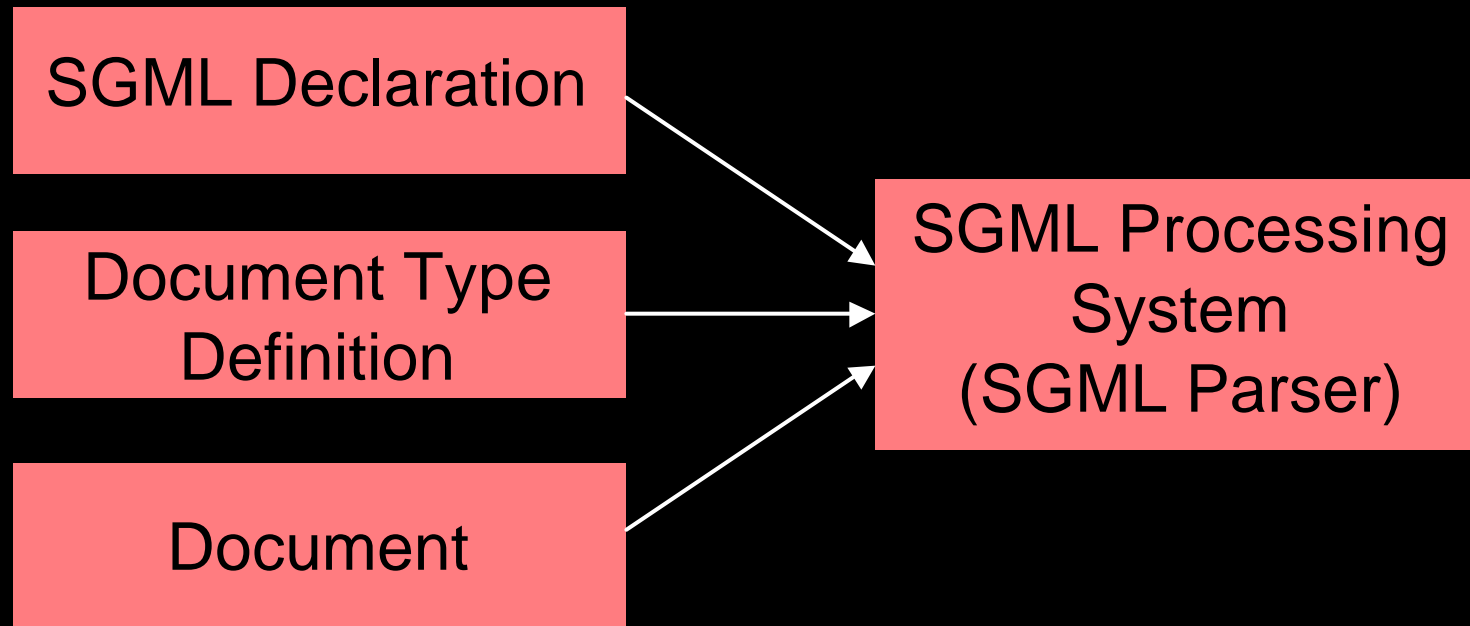
SGML



- Standard Generalized Markup Language
 - ISO-Standard [ISO/IS 8879, 1986]
 - entstanden aus GML, IBM 1969 (nach Goldfarb, Mosher, Lorie)
 - Trennung von Inhalt und Präsentation
- Dilemma konkreter Markup-Sprachen
 - welche Tag-Menge soll vorgegeben sein?
- Generalized Markup
 - beschreibe Dokumente in drei Teilen
 1. SGML Declaration:
Abbildung der abstrakten SGML Syntax auf konkrete Syntax
 2. Document Type Definition, DTD:
Definition der Tags und ihrer Bedeutung
 3. eigentliches Dokument mit Markup



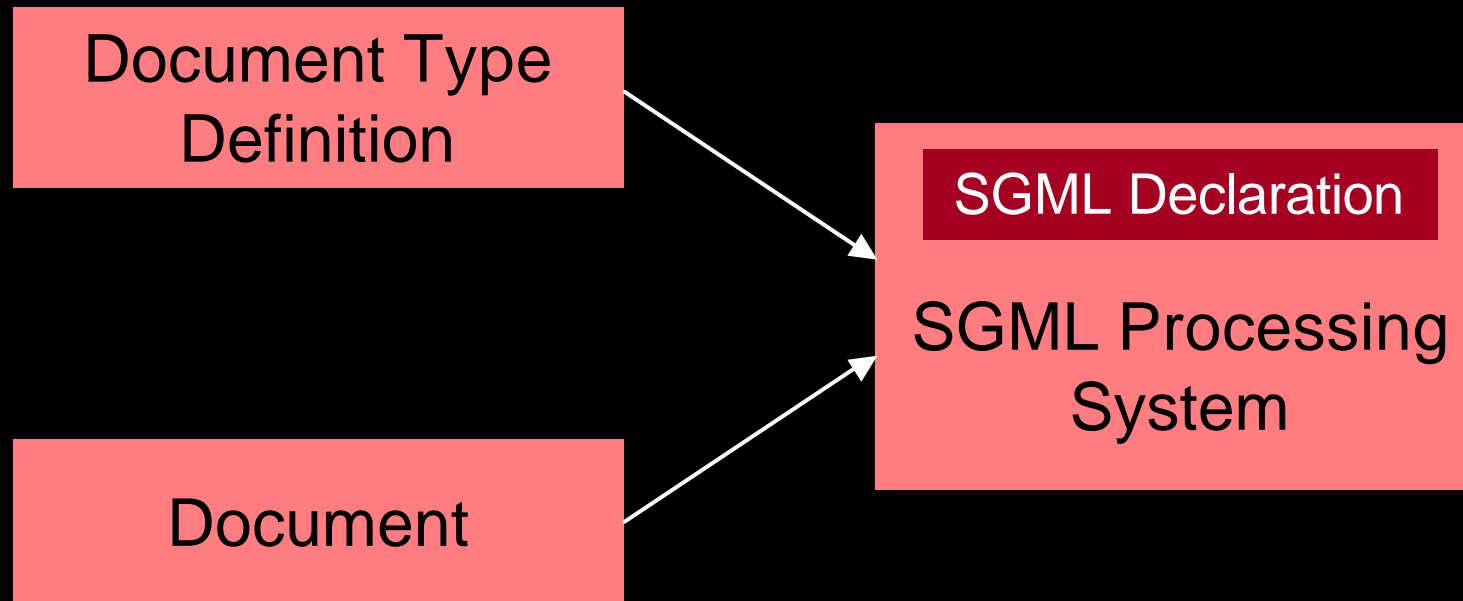
Generelles SGML Processing



Spezielles SGML Processing



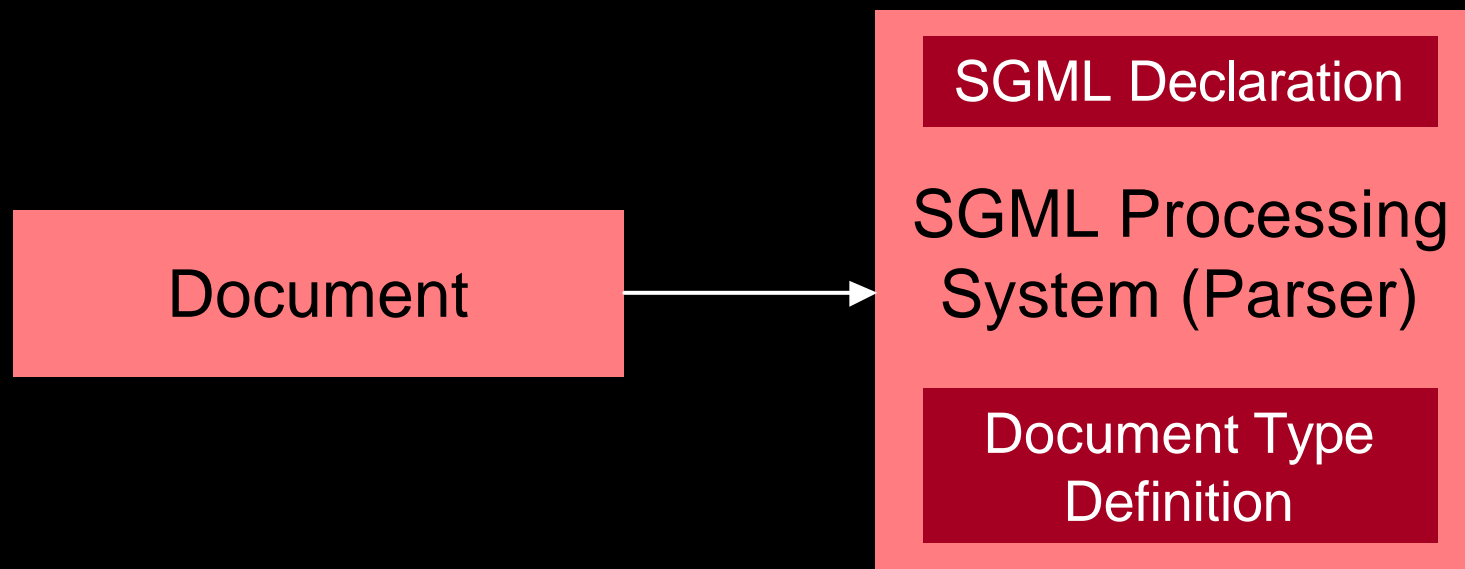
- SGML Syntax festgelegt
 - z.B. bei XML (Extensible Markup Language)



Spezielles SGML Processing



- Document Type Definition und SGML Syntax festgelegt
 - z.B. bei HTML ab Version 4
 - z.B. bei SMIL



SGML Konzepte



- deskriptives statt prozedurales Markup
 - "this is an X" statt "do X here"
- Entity
 - Sammlung von Zeichen, die als Einheit referenziert werden können
- Element
 - Komponente in der hierarchischen Struktur die durch die DTD definiert ist
 - SGML verwendet Tags zur Beschreibung der Element Structure
- Entities werden als Bezugspunkte definiert



Document Type Definition - Beispiel



```
<!ELEMENT aushang (kopf,rumpf)>
<!ELEMENT kopf (titel,datum?)>
<!ELEMENT titel (#PCDATA)>
<!ELEMENT datum (#PCDATA)>
<!ELEMENT rumpf (kurzfassung?, absatz+)>
<!ELEMENT absatz
  ((#PCDATA|vorlesung|dozent)*)>
<!ELEMENT (vorlesung|dozent) (#PCDATA)>
```



Document Type Definition (SGML)



- Eine DTD definiert
 - Elemente einer Dokumentenklasse
 - Regeln für die Kombination dieser Elemente
- Aufbau
 - ***MDO ELEMENT name model MDC***
 - ***<!ELEMENT name model>***
- ***name***
 - ist ein Elementname, z.B. (**buch**) oder
 - eine ODER-Verknüpfung von Elementnamen z.B. (**datum|zeit|ort**)
- ***model***
 - ist der erlaubte Inhalt für das Element
 - Inhalte können kombiniert werden



DTD Modell, Modellgruppe



- **(#PCDATA)**
 - Hier sind Buchstaben erlaubt (character data)
- Verbindungen zwischen Elementen
 - , in einer Sequenz in der vorgegebenen Ordnung
 - & alle müssen vorhanden sein, in beliebiger Ordnung
 - | eines, oder eine beliebige Kombination muss vorhanden sein
- Beispiele
 - `<!ELEMENT aushang (kopf,rumpf)>`
 - `<!ELEMENT kopf (name & autor & datum)>`
 - `<!ELEMENT rumpf (bild | absatz | tabelle)>`



DTD Modell, Modellgruppe



- Häufigkeit der Elemente
 - + mindestens ein Vorkommen des Elements
 - * optional, 0-mal bis n-mal
 - ? optional
- Beispiel
 - `<!ELEMENT dokument (autor, kapitel+, anhang*, zusammenfassung?)>`



DTD Modell, Modellgruppe



- Ausnahmen

- Inclusion

- das angeführte Element kann in der Modellgruppe an beliebiger Stelle verwendet werden

- Exclusion

- das angeführte Element darf in der Modellgruppe nicht verwendet werden

- Beispiele

- `<!ELEMENT dokument(autor,kapitel+)(kursiv)>`

- `<!ELEMENT kursiv (#PCDATA) -(kursiv)>`



DTD Attribute



- Aufbau - Attribute definition list declaration
 - *MDO ATTLIST name attribute-definition MDC*
 - *<!ATTLIST name attribute-definition >*
- Aufbau - Attribute definition
 - *attribute-name declared-value default-value*
 - *declared-value* - alphanumerisch,
wie Variablennamen,
oder reserviertes Wort
 - *default-value* - alphanumerisch
oder reserviertes Wort
 - **#REQUIRED** – muss angegeben sein
 - **#IMPLIED** – wenn nicht vorhanden nimmt der Interpreter einen an
 - **#FIXED** – festes Attribut



DTD Attribute



- Beispiel

- `<!ATTLIST bild>`

<code>name</code>	<code>CDATA</code>	<code>#REQUIRED</code>
<code>type</code>	<code>(bw farbe)</code>	<code>bw</code>
<code>datum</code>	<code>CDATA</code>	<code>#IMPLIED></code>



Fully-Tagged SGML



- Fully-tagged SGML document
 - kann ohne DTD geparsed werden
 - notwendige Information steckt in den Tags
 - Voraussetzung: Typ-Validierung bei Erzeugung
- Extensible Markup Language (XML)
 - Untermenge von SGML für vereinfachtes Parsen
 - insbesondere keine Markup Minimalisierung
 - mit XML können fully-tagged Dokumente erstellt werden



XML



- Beschreibung allgemeiner strukturierter Information
- Well-formed-Documents
 - End-Tags sind verpflichtend, oder `< ./>`
 - Minimierung ist nicht erlaubt
 - Unterscheidet Groß- und Kleinschreibung
- Definition eigener DTDs
- XML-Namensraum
 - Festlegung des Tag-Vokabulars
 - <http://www.w3.org/TR/REC-xml-names/>
- Familie von Sprachen für den Umgang mit XML-Dokumenten
 - XLink, XPointer, XSL, ...



XML Namensraum – Beispiel I



```
<?xml version="1.0"?>
<!--
  all elements here are explicitly in the HTML
  namespace -->
<html:html
  xmlns:html='http://www.w3.org/TR/REC-html40'>
  <html:head>
    <html:title>Vorlesung</html:title>
  </html:head>
  <html:body>
    <html:p>Moved to
      <html:a href='http://www.uni-ulm.de'>here
    </html:a>
    </html:p>
  </html:body>
</html:html>
```



XML Namensraum – Beispiel II



```
<?xml version="1.0"?>
<!--
  initially, the default namespace is "books" -->
<book xmlns='urn:loc.gov:books'
      xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <title>Cheaper by the Dozen</title>
  <isbn:number>1-56849-1379</isbn:number>
  <notes>
    <!-- make HTML the default namespace
          for some commentary -->
    <p xmlns='urn:w3-org-ns:HTML'>
      This is a <i>funny</i> book!
    </p>
  </notes>
</book>
```



DTD - Beispiel



```
<?xml version="1.0"?>
```

```
<!DOCTYPE nameregister [
```

```
  <!ELEMENT nameregister (entry*) >
```

```
  <!ELEMENT entry (name, phone?, email?) + >
```

```
  <!ELEMENT name (#PCDATA) >
```

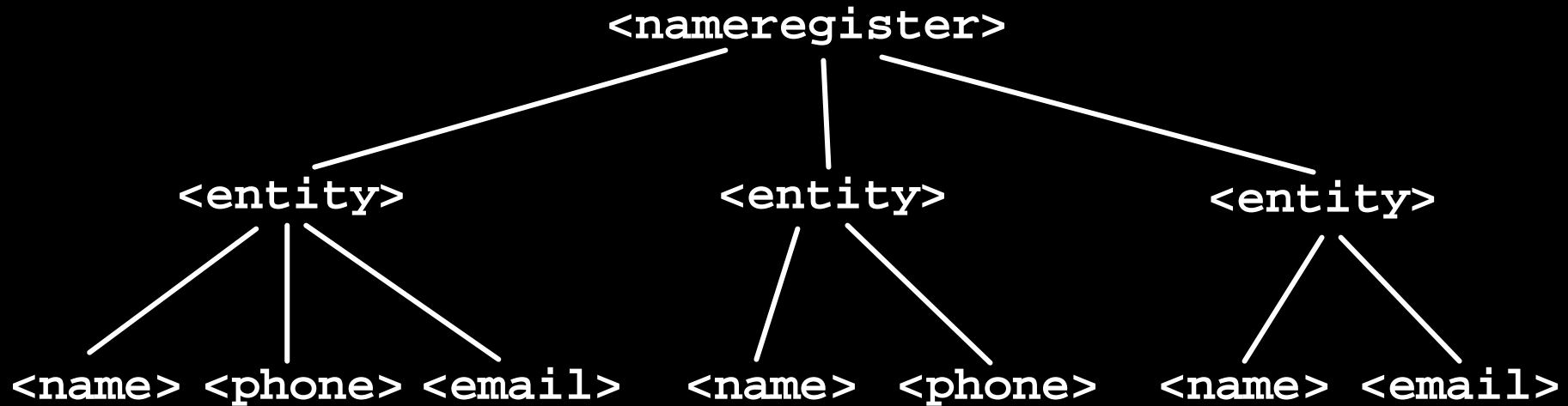
```
  <!ELEMENT phone (#PCDATA) >
```

```
  <!ELEMENT email (#PCDATA) >
```

```
]>
```



XML Baumstruktur



- Jedes XML-Dokument kann hierarchisch – z.B. in Form einer Baumstruktur - dargestellt werden



XML Beispiel



- Namensregister

```
<nameregister>
```

```
  <entry><name>Michael Weber</name>
```

```
    <phone>31300</phone>
```

```
  </entry>
```

```
  <entry><name>Schröder</name>
```

```
    <email>gerd@kanzleramt.de</email>
```

```
  </entry>
```

```
  <entry><name>James Bond</name>
```

```
    <phone>+44 007</phone>
```

```
    <email>007@mi5.gov.uk</email>
```

```
  </entry>
```

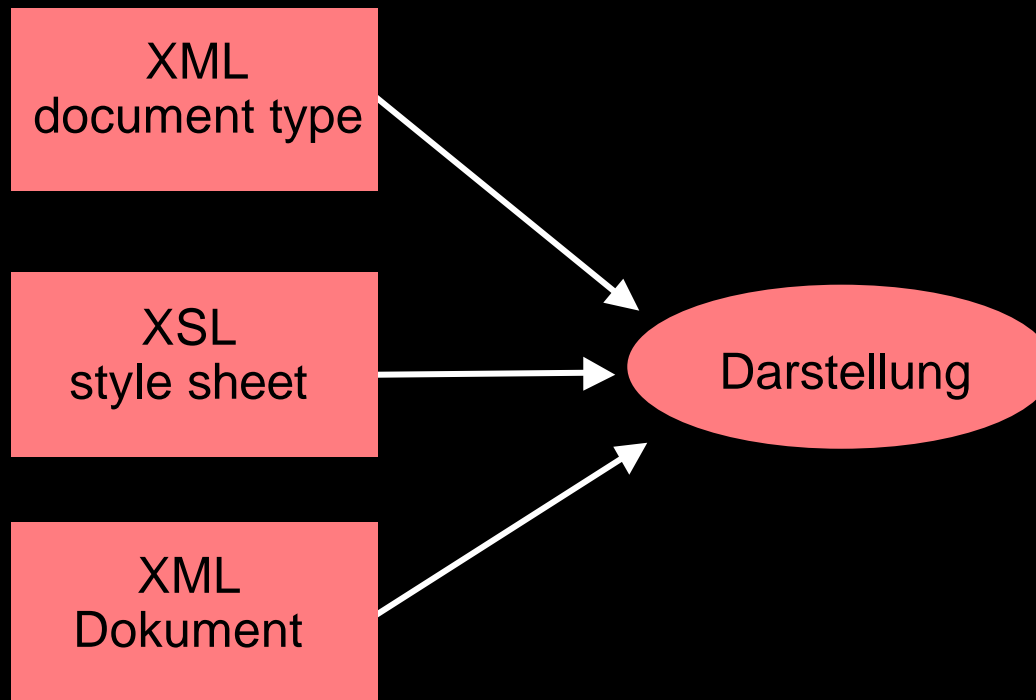
```
</nameregister>
```



XML Darstellung



- XML Browser



SMIL



- Synchronized Multimedia Integration Language
 - Entwicklung des W3C
 - <http://www.w3c.org/AudioVideo/>
 - Integrationsformat für präsentierbare Monomedien-Formate
 - Multimediabeschreibungen, die im Web ablauffähig sind
 - "SMIL = HTML für Multimedia"
- Historie
 - SMIL 1.0 W3C Recommendation, Juni 1998
 - SMIL 2.0 "meta-language" W3C Recommendation, August 2001
 - Derzeit SMIL 2.0 Family Formats: SMIL Profile and SMIL Basic
 - Zukünftiges Family Format: XHTML + SMIL



SMIL



- Eigenschaften
 - Integration der Monomedien
 - Beschreibung von Timing und Synchronisation
 - Adaptives Verhalten bzgl. Nutzer und System
 - Modellierung einer flexiblen aber konsistenten Präsentation



SMIL Profile



- SMIL 2.0 Profile
 - Eine Sprache für die ein Browser entwickelt werden kann
 - Eine Kombination von Modulen der SMIL 2.0 "Meta-Language"
 - Potentiell angereichert durch non-SMIL Konstrukte
- SMIL 2.0 Language Profile (SMIL Profile)
 - Eigentlicher Umfang von SMIL 2.0
- SMIL 2.0 Basic Language Profile (SMIL Basic)
 - Abgespeckte Fassung für mobile Geräte
 - Nimmt eingeschränkte Processor-Faähigkeiten an
- XHTML + SMIL
 - Erweiterung von hypertext-basierter Darstellung um Multimedia-Timing
 - XHTML = HTML mit striktem XML-DTD
- SMIL 1.0
 - Rückwärts-kompatibel, d.h. kann mit SMIL 2.0 Language Profile Browsern dargestellt werden



SMIL Implementierungen (Auszug)



- RealNetworks
 - RealOne Player for SMIL 2.0
 - Marktführer bei SMIL Playern
- Oratrix
 - GRiNS Player und Editor
 - Unterstützt beide Profile
- Microsoft
 - Ab Internet Explorer 5.5 ein Player für XHTML+ SMIL
- Apple
 - Ab QuickTime 4.1 Unterstützung für SMIL 1.0
- Adobe
 - Adobe's SVG Viewer unterstützt SMIL Animationen in SVG
- HELIO
 - SOJA, Java-basierter Player



SMIL Dokumentklassen



■ SMIL

- ```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL
2.0//EN"
"http://www.w3.org/TR/REC-smil/SMIL20.dtd">
<smil
xmlns="http://www.w3.org/2001/SMIL20/Language">
```

## ■ SVG

- ```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
http://www.w3.org/TR/2001/REC-SVG-
20010904/DTD/svg10.dtd
<svg xmlns="http://www.w3.org/2000/svg">
```

■ XHTML+SMIL

- ```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+SMIL
//EN"
http://www.w3.org/2001/SMIL20/WD/xhtmllplusmil.
dtd
<html xmlns=http://www.w3.org/1999/xhtml
xmlns:smil="http://www.w3.org/2001/SMIL20">
```



# SMIL Dokumentstruktur



- **<smil>** Element
  - Container für das SMIL Dokument
- **<head>** Element
  - Layout und Metainformation
  - Ähnlich wie bei HTML
- **<body>** Element
  - Inhalt und Links (wie bei HTML)
  - Timing



# SMIL Dokumentstruktur



- Beispiel

```
<smil>
 <head>
 <meta name="copyright" content="Weber" />
 <layout>
 <!-- layout tags -->
 </layout>
 </head>
 <body>
 <!-- media and synchronization tags -->
 </body>
</smil>
```



# SMIL Medienobjekte und Formate



- Typisierung durch Mimetype
  - z.B. image/jpeg, audio/wav, video/mpeg
- Medienobjekte und gängige Formate
  - **<text>**
    - ASCII, Unicode, HTML
  - **<image>**
    - JPEG, GIF, RealPix, PNG
  - **<video>**
    - MPEG, RealVideo
  - **<audio>**
    - WAV, RealAudio
  - **<animation>**
    - GIF, SVG
  - **<textstream>**
    - RealText (Laufschrift)
  - **<brush>**
    - (eigenes) Zeichnen von Rechtecken

