

Vorlesung / Übungen

# **Multimedia Technologie II**

Prof. Dr. Klaus Hering

SS 2004

**HTWK Leipzig, FB IMN**

# Gültiges Dokument (Ü 1/A2)

```
<?xml version="1.0"?>  
<!DOCTYPE Dok [  
<!ELEMENT Dok (( A , B* ) | ( B , A)+ )>  
<!ELEMENT A EMPTY>  
<!ELEMENT B EMPTY>  
>  
<Dok>  
<B/><A/><B/><A/>  
</Dok>
```

# Gültiges Dokument (Ü 1/A3)

```
<?xml version="1.0"?>  
<!DOCTYPE Dok [  
<!ELEMENT Dok (A,B,A,(B,A)+)>  
<!ELEMENT A (#PCDATA)>  
<!ELEMENT B (#PCDATA)>  
>  
<Dok>  
<A>a</A><B>b</B><A>c</A><B>d</B><A>e</A>  
</Dok>
```

Nicht-deterministische Inhaltsmodelle:

$(A, B, (A, B)^+, A)$

$(A, (B, A)^+, B, A)$

# Gültiges Dokument (Ü 1/A4)

```
<?xml version="1.0"?>  
<!DOCTYPE Dok [  
  <!ELEMENT Dok (A)>  
  <!ELEMENT A (#PCDATA|B)*>  
  <!ELEMENT B (#PCDATA|A)*>  
<Dok>  
<A>aa<B>b<A>a</A>b</B></A>  
</Dok>
```

## 3.5 Auswahl von Dokumentteilen (XPath)

Was ist XPath ?

- Syntax zur Beschreibung von XML-Dokumentteilen
- XPath modelliert XML-Dokumente als Bäume
- Verwendung von Pfadausdrücken
- Bibliothek von Standardfunktionen
- Verwendung innerhalb von **XSLT** und **XPointer**
- W3C Standard

Wir betrachten noch einmal das Dokument von Folie 0470 und überführen es in einen Baum, bestehend aus Element- und Attributknoten. Die Beziehungen zwischen Vater- und Kindknoten ergeben sich unmittelbar aus der Verschachtelungsstruktur der Elemente und der Attributbindung an Elemente.

...

<Katalog>

<Buch>

<Titel>XML in der Praxis</Titel>

<Autoren>

<Autor>Henning Behme</Autor><Autor>Stefan Mintert</Autor>

</Autoren>

<Verlag>Addison-Wesley</Verlag>

<ISBN >3-8273-1636-7</ISBN>

<Preis Waehrung="USD">52.25</Preis><Preis Waehrung=„ATS">700</Preis>

</Buch>

<Buch>

<Titel>Learning XML</Titel>

<Autoren><Autor>Erik T. Ray</Autor></Autoren>

<Verlag>O&apos;Reilly</Verlag>

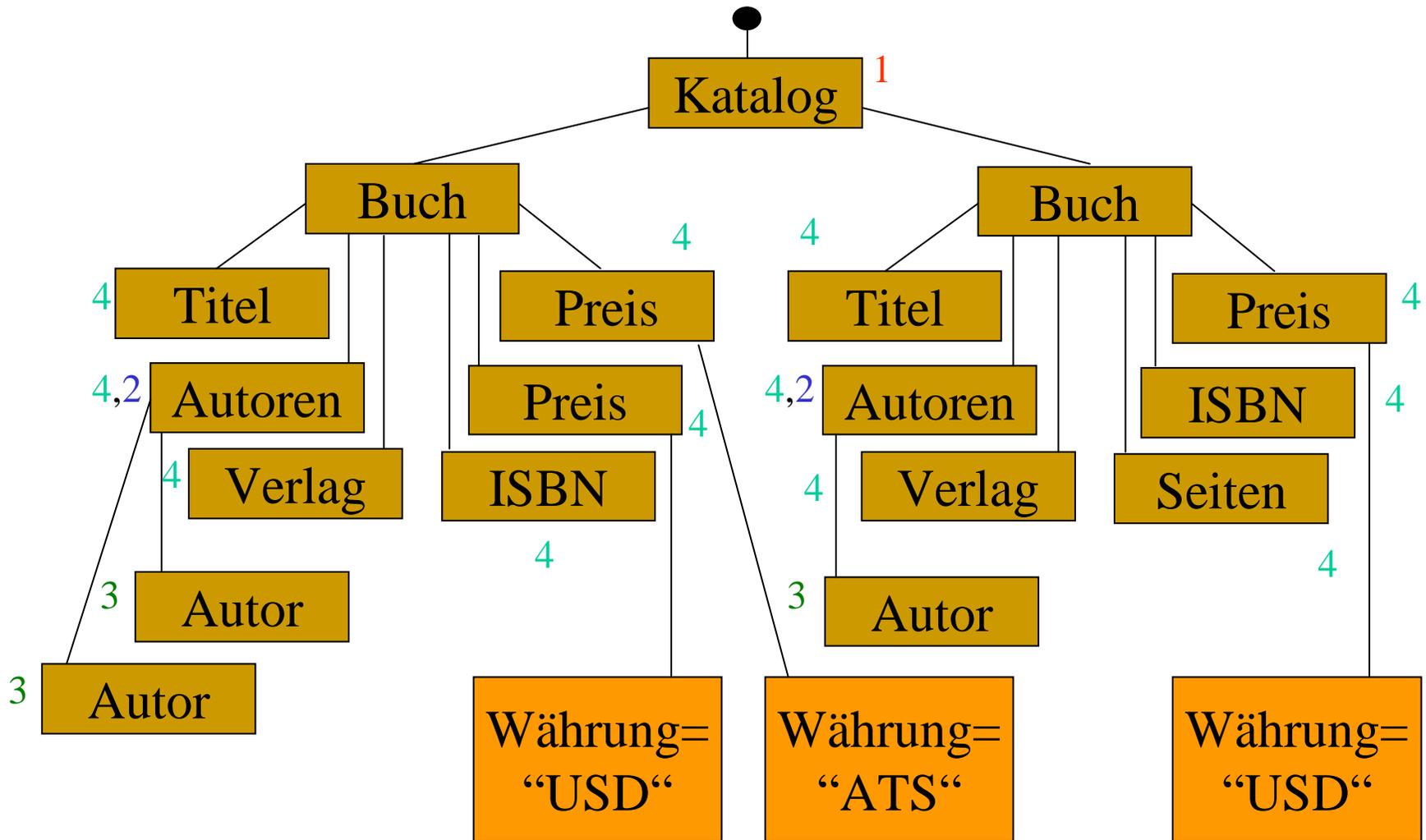
<Seiten>356</Seiten>

<ISBN>0-596-00046-4</ISBN>

<Preis Waehrung="Euro">47</Preis>

</Buch>

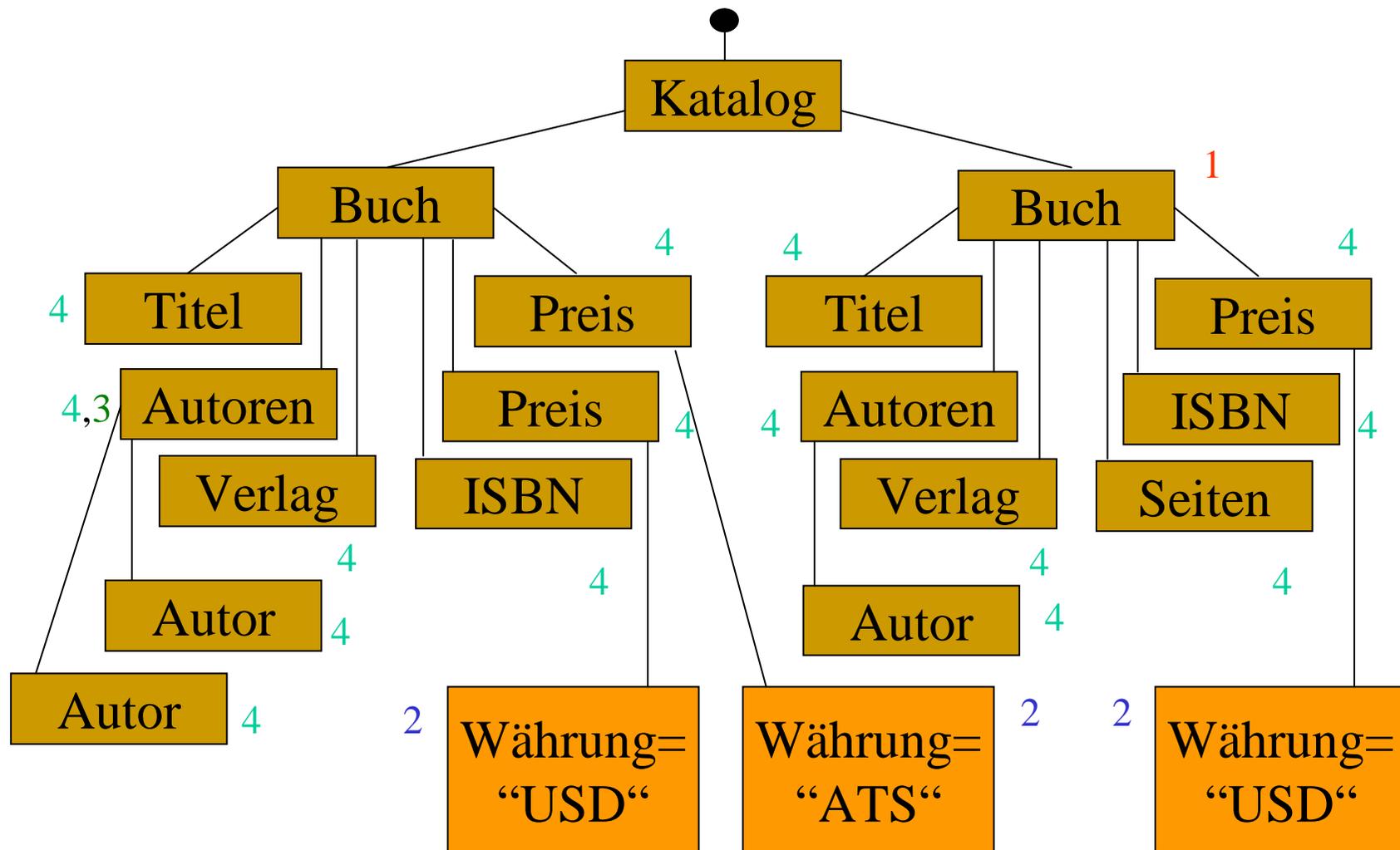
</Katalog>



Verschiedene Pfadausdrücke und die durch Sie ausgewählte Knotenmenge:

/Katalog 1  
 /Katalog/Buch/Autoren 2

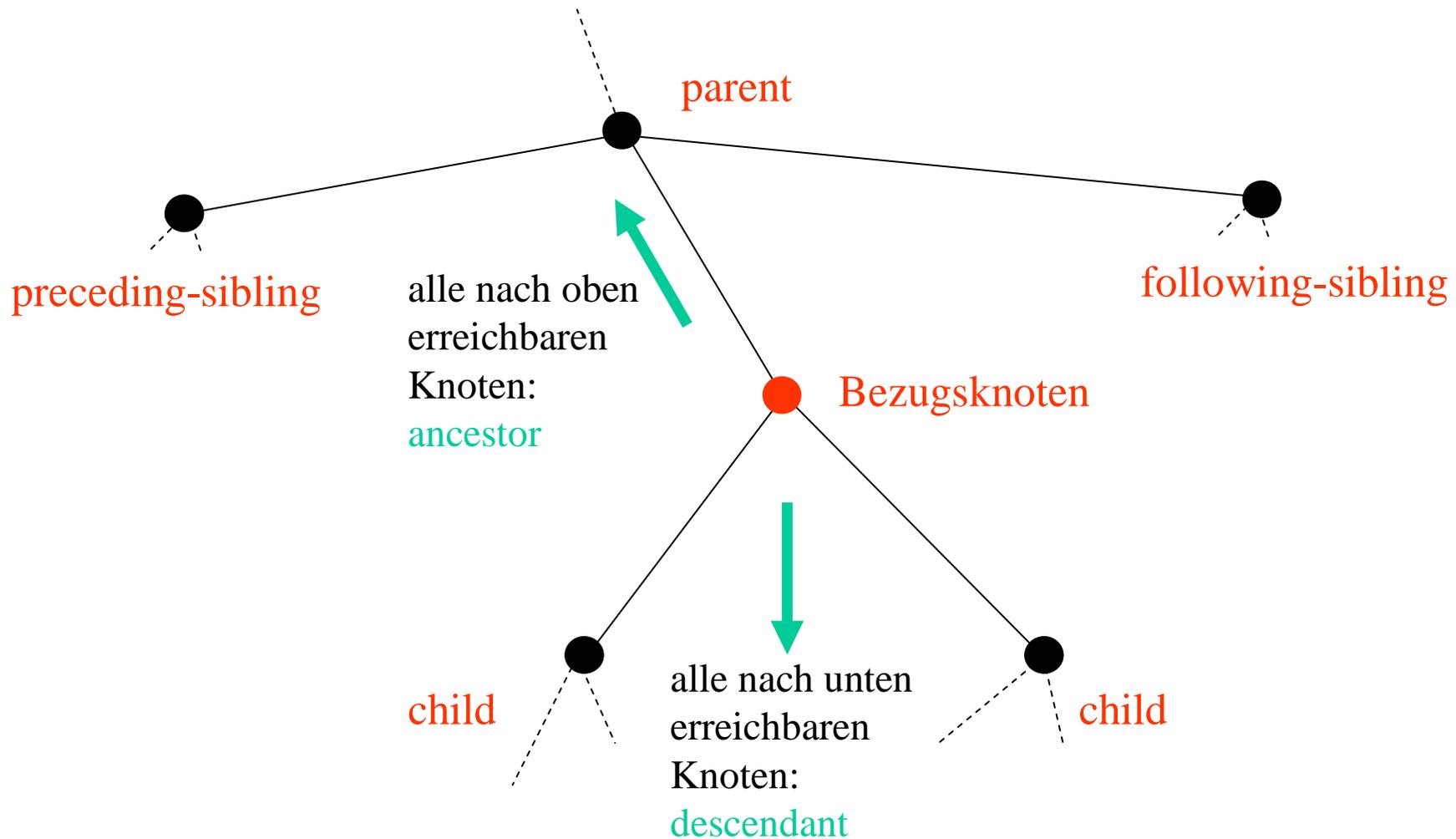
//Autor 3  
 /Katalog/Buch/\* 4



Verschiedene **Pfadausdrücke** und die durch Sie ausgewählte Knotenmenge:

<code>/Katalog/Buch[2]</code>	1	<code>//*[count(Autor)=2]</code>	3
<code>//@Währung</code>	2	<code>//Buch/descendant::*</code>	4

# Knoten im Baum relativ zu einem Bezugsknoten



Neben dem in den Übungen verwendeten XPath-Tutorial ist ein weiteres unter

<http://www.w3schools.com/xpath/default.asp>

zu finden.

### **3.6 Links (XLink)**

3 Aspekte der Bildung von Linkstrukturen (Ausgangspunkt der Betrachtung bei XLink-Entwicklern):

#### **Beteiligte Ressourcen**

zu verbindende Dokumentbestandteile (lokal oder außerhalb)

#### **Verbindungsdefinition**

Beziehung zwischen den betrachteten Dokumentbestandteilen

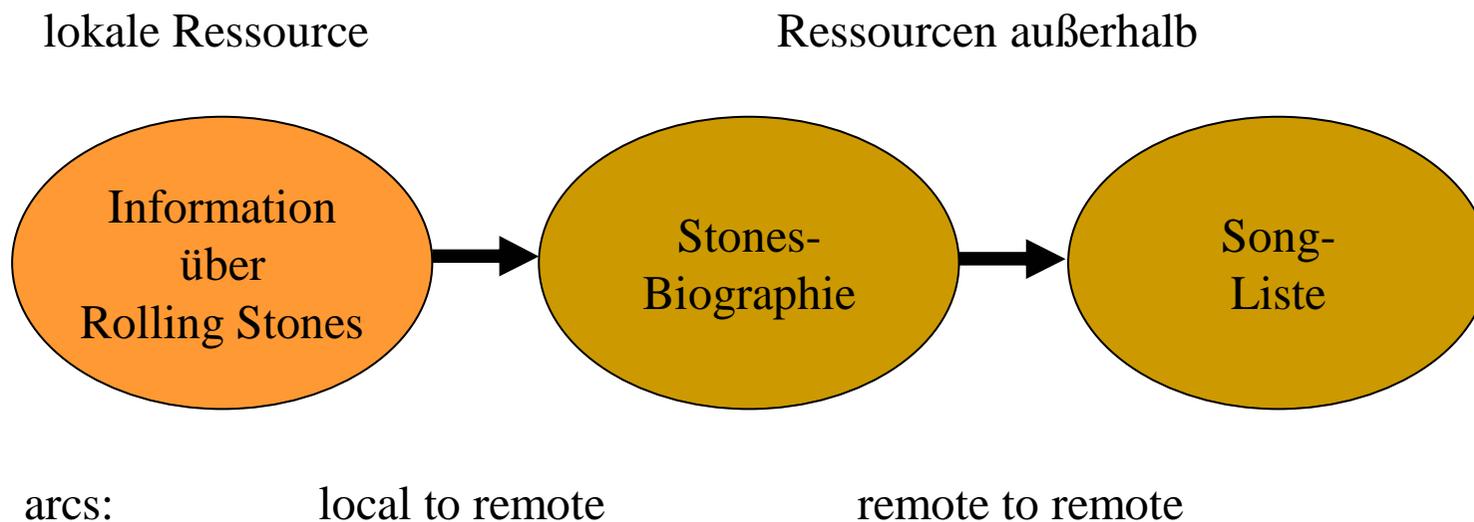
#### **Traversierungsregeln**

*Arcs*, welche die Richtung der Linkverfolgung angeben

⇒ Begriff des *extended link* (weit über Linkbegriff im Rahmen von HTML hinausgehend)

Ein *extended link* erscheint als Attribut eines XML-Elements, welches selbst als Hülle für weitere XLink-Definitionen in bezug auf beteiligte Ressourcen und *arcs* dient.

Beispiel (zunächst schematisch):



```

<performers xmlns:xlink="http://www.w3.org/1999/xlink">;
  <item xlink:type="extended">
    <!-- link definition (local) -->
    <link xlink:type="resource" xlink:label="overview"
      xlink:title="Information on Stones">Rolling Stones</link>

    <!-- link definitions (remote) – Stones‘ biography, songs -->
    <link xlink:type="locator" xlink:href="bio.xml"
      xlink:label="bio" xlink:title="Biography" />
    <link xlink:type="locator" xlink:href="songs.xml"
      xlink:label="songs" xlink:title="Songs"/>

    <!-- local to remote arc - from name to biography -->
    <arc xlink:type="arc" xlink:from="overview" xlink:to="bio"
      xlink:show="replace" xlink:actuate="onRequest" />

    <!-- remote to remote arc - from biography to song list -->
    <arc xlink:type="arc" xlink:from="bio" xlink:to="songs"
      xlink:show="replace" xlink:actuate="onRequest" />

  </item>
</performers>

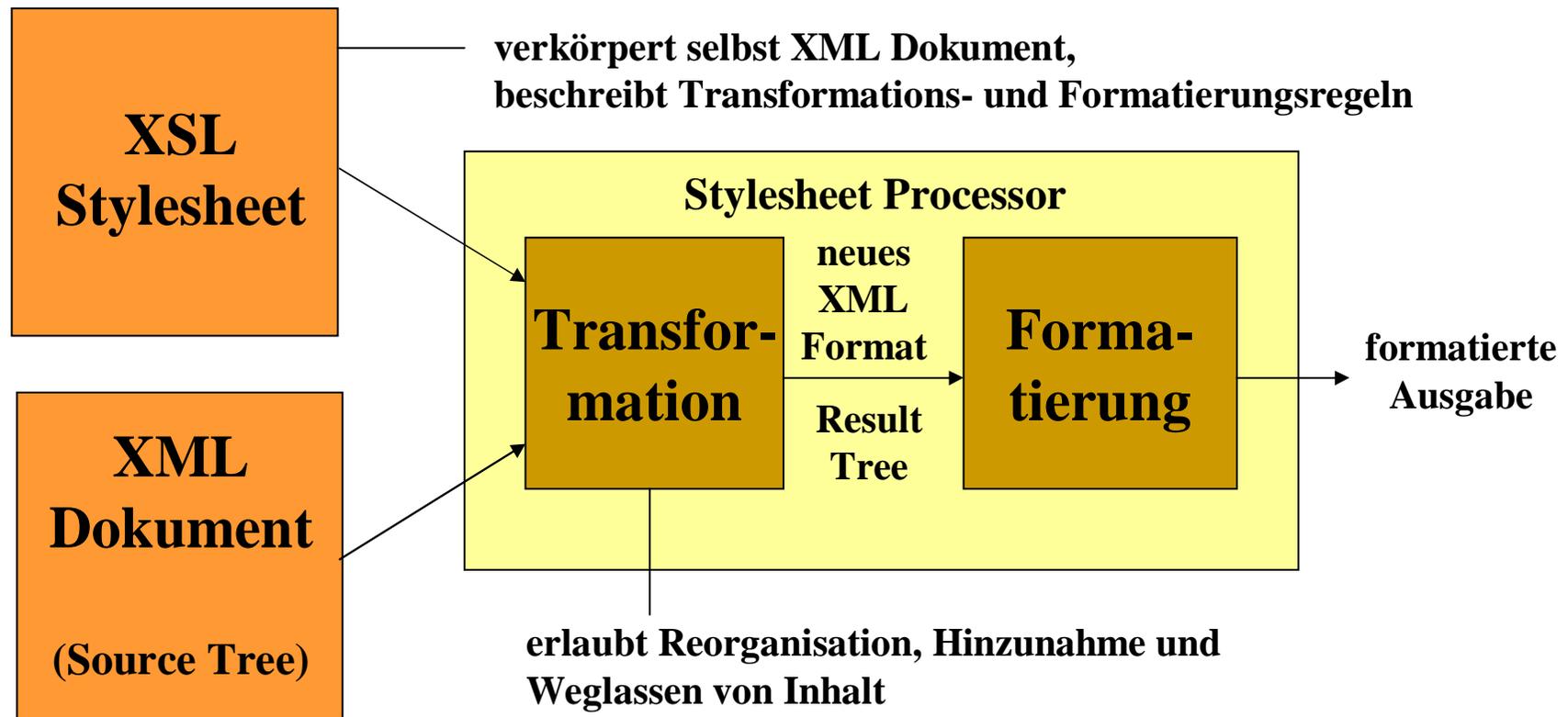
```

modifiziert ausgehend von  
<http://www.melonfire.com/community/columns/trog/print.php?id=90>

- prinzipiell jedes XML-Element zu einem Link erweiterbar
- Anzahl der an einem Link beteiligten Ressourcen nicht auf 2 beschränkt, Bidirektionalität durch 2 *arcs* ausdrückbar
- herkömmliche *simple links* sind als spezieller Typ verfügbar
- Nutzung der Ausdruckstärke von *XPointer (XPath)* zur Adressierung von Zieldokumentbestandteilen
- in der Regel Speicherung von *extended links* separat von den Ressourcen, die sie „verbinden“; Link-updates nicht an Schreibrechte bezüglich Quell- und Zielressourcen gebunden

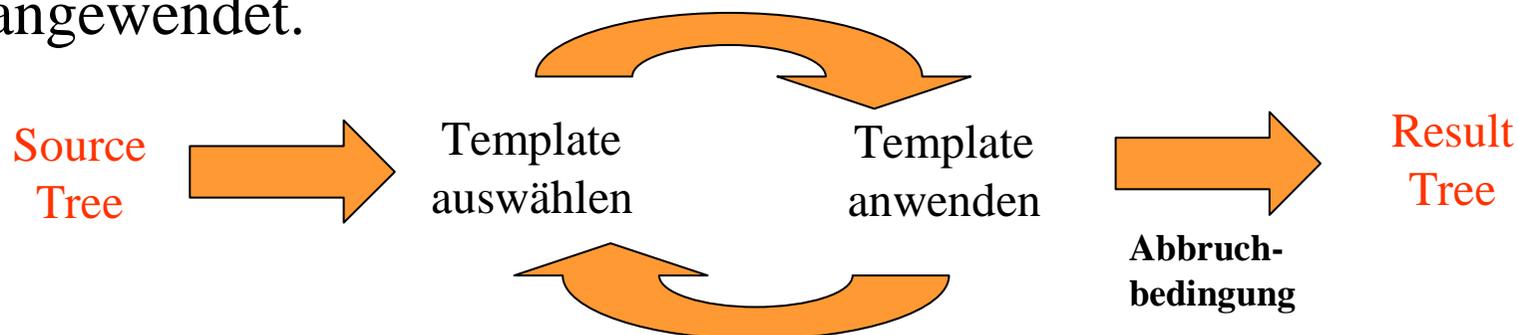
## 3.7 Extensible Stylesheet Language (XSL)

XSL = Transformations- + Formatierungssprache  
(XSLT) (Formatting Objects)



Ein XSL Stylesheet enthält eine Folge von *Templates*. Ein Template kann als Folge von Transformationsregeln aufgefaßt werden, welche die Erzeugung eines Teils des Resultatbaums ausgehend von einem Teil des Quellbaums beschreiben. Jedes Template (als XML-Element) hat ein Attribut *match*, dessen Wert Teile des Quellbaums beschreibt (XPath-Ausdrücke!).

Im Laufe eines Transformationsprozesses werden, beginnend mit dem Wurzelknoten des Quellbaums, Knoten des Quellbaums zu aktiven Knoten. Für diese Knoten wird dann jeweils ein Template mit passendem *match*-Attributwert gesucht und angewendet.



## Beispiel-Dokument

```
<?xml version="1.0"?>
<greeting>
  Hello, World!
</greeting>
```

## Transformations- Resultat

```
<html>
<body>
<h1>
  Hello, World!
</h1>
</body>
</html>
```

**D. Tidwell: XSLT**

0900

## XSL Stylesheet

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="html"/>

  <xsl:template match="/">
    <xsl:apply-templates select="greeting"/>
  </xsl:template>

  <xsl:template match="greeting">
    <html>
      <body>
        <h1>
          <xsl:value-of select="."/>
        </h1>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Multimedia Technologie II, Prof. Klaus Hering, SS 2004

XSL-Tutorial:

<http://www.w3schools.com/xsl/default.asp>

weiterführende Literatur:

- D. Tidwell:  
XSLT.  
O'Reilly, 2001
- P. Spencer:  
Professional XML Design and Implementation.  
Wrox Press, 1999

## 4. Synchronized Multimedia Integration Language (SMIL)

### 4.1 Grundlagen

⇒ Verbindung von *Clips* (Medienobjekte wie Text-, Bild-, Audio-, Video-clips, ...) zu einer Gesamtpräsentation (Zusammenfügen vorhandener Bausteine)

⇒ Was kann spezifiziert werden ?

#### **Inhalt**

(Teile von) Medienobjekte(n)

#### **Räumliches Layout**

Objektpositionierung auf einer Präsentationsfläche

#### **Zeitliches Layout**

Einsatz und Dauer einer Objektpräsentation

#### **Adaptivität**

Anpassung an Kontextbedingungen

#### **Objektbeziehungen**

Links, Synchronisation

⇒ Enge Verbindung mit *Streaming Technologie* (Übertragung von Daten derart, dass sie am Ziel als kontinuierlicher Strom bearbeitet werden können)

## Vorzüge von SMIL

- Streaming von Clips, die auf verschiedenen Servern liegen (Verpacken in gemeinsames File nicht erforderlich, SMIL ist eine *Integrationssprache*)
- Erzeugung interaktiver Multimedia-Präsentationen
- Verbindung einer Präsentation mit anderen Präsentationen oder Webseiten
- Kontextsensitive Präsentationen (etwa abhängig von bevorzugter Sprache beim Betrachter oder verfügbarer Bandbreite)
- Animationen und Übergangseffekte
- Automatische Generierbarkeit speziell zugeschnittener Präsentationen

SMIL 1.0: 1998, Spezifikation umfaßte 30 Seiten

SMIL 2.0: 2001, Spezifikation umfaßt 540 Seiten

SMIL 2.0 definiert 10 *funktionelle Bereiche*:

Timing	Time Manipulations
Animation	Content Control
Layout	Linking
Media Objects	Metainformation
Structure	Transitions

Diese Bereiche werden wiederum in insgesamt 45 *Module* untergliedert. Module enthalten semantisch miteinander eng verbundene (*XML-*)*Elemente*, *Attribute* und *Attributwerte*. Ein *Language Profile* wird als Kombination von Modulen gebildet.

- *SMIL 2.0 Language Profile* (enthält fast alle SMIL 2.0 Module)
- *SMIL 2.0 Basic Language Profile* (reduzierte Funktionalität zum Einsatz in Mobiltelefonen und anderen portablen Systemen)
- *XHTML+SMIL* (Mischung von Modulen unterschiedlicher Namensräume)

## Literatur zu SMIL

- W3C:  
*Synchronized Multimedia Integration Language (SMIL 2.0)*  
<http://www.w3.org/TR/smil20/> [11/21/2002]  
⇒ W3C Recommendation (7. August 2001)
- RealNetworks Inc.:  
*RealNetworks Production Guide with RealOne Player*  
<http://service.real.com/help/library/encoders.html> [11/21/2002]  
⇒ SMIL 2.0 ⇒ RealText, RealPix (proprietäre Formate)
- Wolfgang von Keitz  
*Das SMIL-Textbuch*  
<http://v.hdm-stuttgart.de/~keitz/skripte/SMILStart.htm> [11/21/2002]  
⇒ SMIL 1.0 (Nachfolger für Version 2.0 in Vorbereitung)
- Frank Goetz  
*SMIL - Multimedia im Internet mit Realsystem G2*  
Addison-Wesley, 2000  
⇒ SMIL 1.0 ⇒ RealServer

## SMIL – Implementatoren

- *RealNetworks*
  - führend in Bezug auf SMIL-Player
  - RealOne Player für SMIL 2.0
- *Oratrix*
  - GRiNS-Player und –Editor
  - Unterstützung für alle Profiles
- *Microsoft*
  - ab IE 5.5 : XHTML+SMIL
- *Apple*
  - ab QuickTime 4.1 Unterstützung von SMIL 1.0
- *Adobe*
  - SVG Viewer unterstützt SMIL-Animationen in SVG

# Dokument-Grundstruktur

SMIL 2.0



```
<smil  
xmlns="http://www.w3.org/2001/SMIL20/Language">
```

```
<head>
```

...optional section with all header markup...

Metainformation  
Layout

```
</head>
```

```
<body>
```

...required section with all body markup...

Inhalt  
Timing

```
</body>
```

```
</smil>
```

## SMIL 2.0 – Clip Source Tags

<i>&lt;animation/&gt;</i>	z.B. Macromedia Flash
<i>&lt;audio/&gt;</i>	z.B. RealAudio
<i>&lt;brush/&gt;</i>	farbiges Rechteck
<i>&lt;img/&gt;</i>	z.B. GIF, JPEG, PNG
<i>&lt;ref/&gt;</i>	Clip-Typ, der nicht den anderen Tags zugeordnet werden kann
<i>&lt;text/&gt;</i>	einfacher (statischer) Text
<i>&lt;textstream/&gt;</i>	Text in streaming-Format, z.B. RealText
<i>&lt;video/&gt;</i>	z.B. RealVideo