

Vorlesung / Übungen

## **Multimedia Technologie II**

Prof. Dr. Michael Frank / Prof. Dr. Klaus Hering

Sommersemester 2004

**HTWK Leipzig, FB IMN**

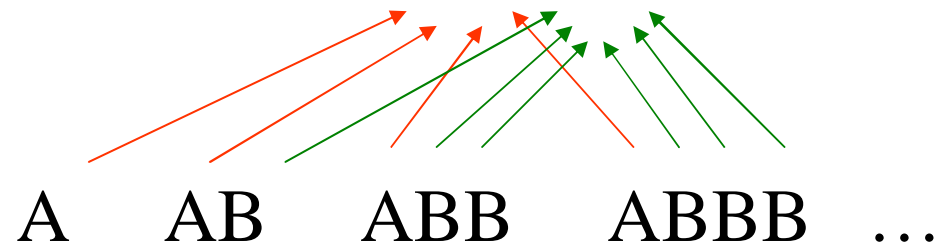
**Zusätzliche Bedingung** an Beschreibungen der Kategorie *element\_content* innerhalb von Elementtyp-Deklarationen (XML 1.0 Spezifikation):

Beim Parsen eines korrekt gebildeten Elementinhalts (von links nach rechts) muß der Parser zu jedem Vorkommen eines Elementtyps *e* (ohne Verwendung weiter rechts liegender Vorkommen) **eindeutig** ein Vorkommen von *e* innerhalb von *element\_content* zuordnen können, welches das erstere Vorkommen von *e* als korrekt ausweist.

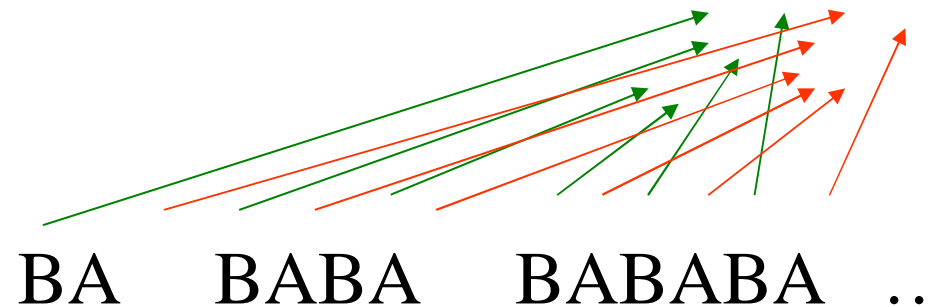
⇒ **deterministic content model**

Beispiel (deterministic content model):

<!ELEMENT Dok (( A , B\* ) | ( B , A)+ )>

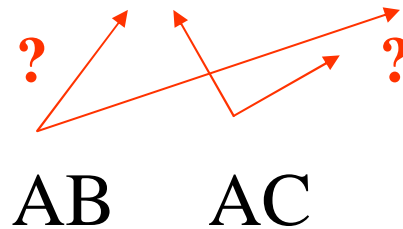


<!ELEMENT Dok (( A , B\* ) | ( B , A)+ )>



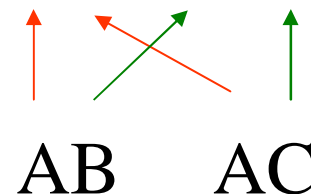
Beispiel (non-deterministic content model):

<!ELEMENT Dok (( A , B) | ( A , C))>



Überführbar in ein deterministisches Modell  
(nicht in jedem Fall möglich):

<!ELEMENT Dok ( A , ( B | C))>



## Attributlisten-Deklarationen

- ⇒ Definition der zu einem Elementtyp gehörenden Attributmenge
- ⇒ Festlegung von Typeinschränkungen für diese Attribute
- ⇒ Modus der Bereitstellung von Attribut-Werten

<!ATTLIST *el\_name att\_definition\** >

          |  
          Elementtyp

*att\_definition ::= att\_name att\_type value\_supply*

Attributtypen:

- String type (**CDATA**)  
beliebige Zeichenketten können als Wert angenommen werden
- Tokenized Type (**ID**, **IDREF**, **IDREFS**, **ENTITY**, **ENTITIES**, **NMTOKEN**, **NMTOKENS**)  
unterschiedliche syntaktische und semantische Einschränkungen für Werte
- Enumerated Type  
Attribut kann einen Wert aus einer Liste annehmen

Modi der Bereitstellung von Attributwerten (*value\_supply*):

- **#REQUIRED**

Der mit dem Attribut verbundene Elementtyp muß bei jedem Auftreten innerhalb eines Dokuments im Rahmen des *start tags* Attributname und Attributwert explizit enthalten.

- **#IMPLIED**

Die Angabe des Attributs innerhalb des *start tags* entsprechender Elemente ist nicht gefordert und es wird kein default-Wert für das Attribut festgelegt.

- Festlegung eines default-Wertes für das Attribut in der Form

**#FIXED?** “*value*“

Wird für ein Vorkommen des mit dem Attribut verbundenen Elementtyps das Attribut nicht explizit angegeben, wird es implizit als mit dem default-Wert angegeben angesehen. Bei Verwendung des Schlüsselworts **#FIXED** muß das Attribut in jedem Fall (auch bei expliziter Attributangabe) den default-Wert haben.

Demonstration von Varianten vor dem Hintergrund unseres **Katalog-**Dokumenttyps:

<!ATTLIST Preis Waehrung CDATA #REQUIRED>

⇒ Jedes Vorkommen eines **Preis**-Elements muß im *start tag* den Attributnamen **Waehrung**, gefolgt von einem Gleichheitszeichen und einer beliebigen in ““ eingeschlossenen Zeichenkette (interpretiert als Währungsbezeichnung) enthalten.

<Preis Waehrung=“USD“>52.25</Preis>

---

<!ATTLIST Preis Waehrung (Euro|USD) “Euro“>

⇒ Durch Verwendung eines *Enumerated Type* ist die Menge der möglichen Werte für das Attribut **Waehrung** (im Kontext des Elementtyps **Preis**) auf **Euro** und **USD** eingeschränkt. Als default-Wert wird **Euro** festgelegt.

<Preis Waehrung=“USD“>52.25</Preis> <Preis Waehrung=“Euro“>5</Preis>  
<Preis>10</Preis> ⇒ angenommener Attributwert von **Waehrung**: **Euro**

<!ATTLIST Preis Waehrung CDATA #IMPLIED>

⇒ Es wird kein default-Wert für das Attribut **Waehrung** festgelegt, das Attribut muß in einem **Preis**-Element nicht angegeben werden.

<Preis Waehrung="Rubel">50</Preis>

<Preis>111</Preis> ⇒ kein Attributwert für **Waehrung**

---

<!ATTLIST Preis Waehrung CDATA #FIXED "USD">

⇒ **USD** wird als nicht explizit „überlagerbarer“ default-Wert für das Attribut **Waehrung** festgelegt.

<Preis Waehrung="USD">12</Preis>

<Preis>12</Preis> ⇒ angenommener Attributwert: **USD**

<Preis Waehrung="Schilling">12</Preis>



führt zu ungültigem Dokument



Die Attributtypen **ID**, **IDREF** und **IDREFS** (*Tokenized Types*) erlauben die Markierung von Elementvorkommen innerhalb von Dokumenten und die Erzeugung von Links, die diese Marken als Ziel haben.

Attributwerte des Typs **ID** sind Namen, die innerhalb eines Dokuments nicht mehrfach in dieser Eigenschaft vorkommen dürfen. Ein Element-Typ darf höchstens mit einem Attribut vom Typ **ID** verbunden sein.

Wir erweitern unseren **Katalog**-Dokumenttyp um folgende Attributlisten-deklaration:

```
<!ATTLIST Buch Marke ID #REQUIRED>
```

Jedes Vorkommen eines Elements vom Typ **Buch** wird damit in einem (gültigen) Dokument über den Attributwert von **Marke** eindeutig lokalisierbar.

*Links* in Bezug auf Elemente mit einem bestimmten Attributwert vom Typ **ID** werden mittels Attributen der Typen **IDREF** und **IDREFS** realisiert.

Attributwerte des Typs **IDREF** / **IDREFS** verkörpern Namen / Folgen von Namen, welche innerhalb eines gültigen Dokuments mit dem Wert eines **ID** Attributs eines Elements übereinstimmen müssen.

Wir ergänzen die DTD unseres **Katalog**-Dokumenttyps noch um zwei Elementtyp-Deklarationen für **ref** und **refs**. Entsprechende Elemente sollen ausschließlich dem Referenzieren von Elementen bzw. Mengen von Elementen dienen. Desweiteren modifizieren wir die Elementtyp-deklaration in Bezug auf **Katalog**:

```
<!ELEMENT Katalog ((Buch)*, ref, refs)>
```

...

```
<!ELEMENT ref EMPTY>
```

```
<!ELEMENT refs EMPTY>
```

```
<!ATTLIST Buch Marke ID #REQUIRED>
```

```
<!ATTLIST ref Ziel IDREF #REQUIRED>
```

```
<!ATTLIST refs Ziele IDREFS #REQUIRED>
```

Mit dem folgenden Dokumentkörper (root-Element) entsteht ein gültiges Dokument:

```
<Katalog>
<Buch Marke="XP">
<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 Marke="LX">
<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><ref Ziel="XP"></ref><refs Ziele="XP LX"></refs>
</Katalog>
```

bisher betrachtete Deklarationen in DTDn:

- **Entity-Deklarationen**
- **Elementtyp-Deklarationen**
- **Attributlisten-Deklarationen**

weitere Form: **Notations-Deklarationen** (Festlegung von *Notationstypen*, welche von einem *XML processor* dazu benutzt werden, Daten dieser Typen einem entsprechenden Handler zu übergeben)

Beispiel: Einbeziehung einer Bilddatei in ein XML-Dokument

```
<!ENTITY mypic SYSTEM "photos/me.gif" NDATA gif>  
<!NOTATION gif SYSTEM file:///C:/Programs/IE/iexplore.exe>
```

```
...  
&mypic;
```

|  
Notationstyp

|  
Handler

## **Generelle Ziele bei der Entwicklung einer XML-Applikation**

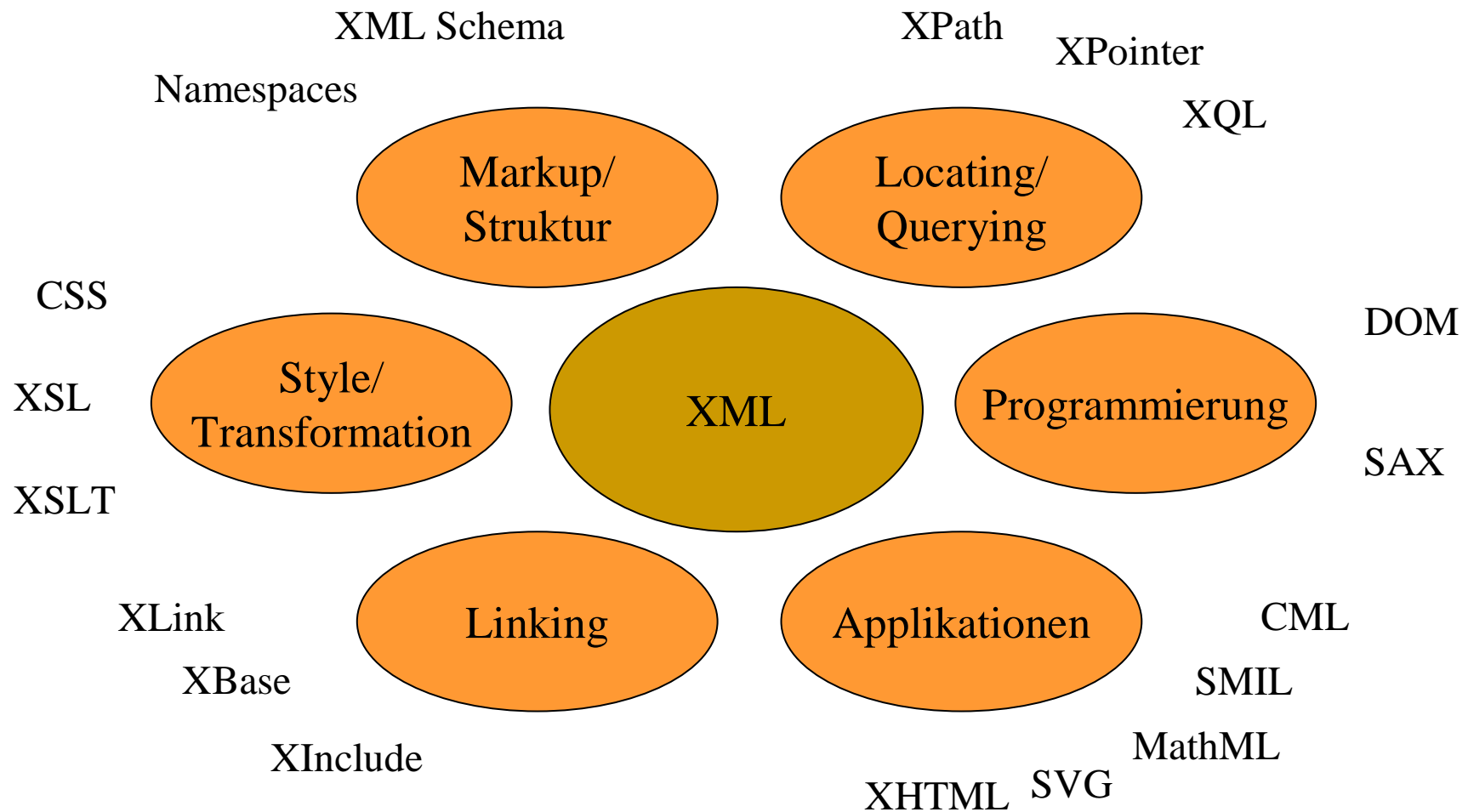
- ausgewogene Verwendung von Elementen und Attributen
- klare Bezeichnung syntaktischer Objekte
- logische hierarchische Strukturen

von **Elementen** ausgehende Informationen (nach E.T. RAY):

- Typ
  - Elementtyp-Name sollte Rolle des Elements innerhalb des Dokuments klar zum Ausdruck bringen
  - einfache Regelung in Bezug auf Groß- und Kleinschreibung
  - Vermeidung „allgemeiner“ Elementtypen zur Aufnahme vielfältiger Inhalte

- Inhalt
  - Auszeichnung von Textpassagen für Darstellungszwecke (*inline elements*)
  - Dokumentunterteilung auf einer Hierarchieebene (*components, blocks*)
- Position
  - Ordnung von Elementen innerhalb eines anderen Elements
  - Elemente können der Markierung von Textstellen dienen (etwa zur Einfügung von Grafik)
  - Auszeichnungsmöglichkeit von Textbereichen durch zwei verschiedene Elemente
- Hierarchie
  - das Enthaltensein eines Elements in einem anderen gibt Kontextinformation für seine Behandlung
- Namensräume (*Namespaces*)
  - Zusammenfassung logisch miteinander in Beziehung stehender Elemente
  - Notation über qualifizierende Namen (Präfix zur Bezeichnung eines Namensraums)

### 3.3 Standards im XML-Umfeld



## Markup/Struktur

- *Namespaces* : über URI referenzierte Sammlungen von Elementtyp- und Attributnamen zur Mehrfachverwendung in XML-Dokumenten
- *XML Schema* : Methode der Modellierung von XML-Dokumenten (alternativ zu DTD), *XML schemes* verkörpern selbst wohlgeformte XML-Dokumente

## Locating/Querying

- *XPath* : Sprache zur Lokalisierung von XML-Dokumentteilen
- *XPointer* : Standard zur Pfadspezifizierung innerhalb von URIs, Adressierung interner Dokumentstrukturen auf der Basis von Elementtypen, Attributwerten, Inhalten und Positionsinformationen
- *XQL* : Anfragesprache die XML als Datenmodell nutzt



## Style/Transformationen

- ***CSS (Cascading Style Sheets)*** : Sprache für Zuweisung von Formaten zu Dokumentelementen (Problem: keine Strukturtransformationen möglich und nur eingeschränkte Nutzung der vorliegenden Dokumentstruktur)
- ***XSL (Extensible Stylesheet Language)***: Sprache zur Beschreibung von Transformationen von XML-Dokumenten und Formatzuweisungen zu Dokumentteilen, Stylesheet ist selbst wohlgeformtes XML-Dokument; *XPath*, *XSLT* und *XML Formatting Objects* sind XSL-Bestandteile
- ***XSLT (XSL Transformations)***: Sprache zur Beschreibung von Transformationen von XML-Dokumenten in ebensolche

## Programmierung

- ***SAX (Simple API for XML)*** : ereignisgesteuertes API zur Manipulation von XML-Dokumenten mit Java, „flaches“ Dokumentmodell (keine Objekthierarchien)
- ***DOM (Document Object Model)*** : API zur Manipulation von XML-Dokumenten, Dokument als Sammlung von Objekten

## Linking

- ***XLink (XML Linking Language)*** : erlaubt die Spezifikation von Elementen zur Erzeugung und Beschreibung von Links in XML-Dokumenten, geht über die Möglichkeiten von HTML weit hinaus, Nutzung von *XPath*
- ***XBase (XML Base)***: Definition von URIs für Teile von XML-Dokumenten
- ***XInclude*** : Standard für die Einbettung von XML-Dokumenten

## Applikationen

- ***XHTML (Extensible Hypertext Markup Language)*** :  
Redefinition von HTML 4 auf der Basis von XML 1.0
- ***SVG (Scalable Vector Graphic)*** : Beschreibung zweidimensionaler Vektor- und gemischter Vektor-/Rastergrafik, 3 Grafik-Grundtypen (Figuren, Bilder, Text)
- ***MathML (Mathematical Markup Language)*** : Beschreibung mathematischer Notationen
- ***SMIL (Synchronized Multimedia Integration Language)*** : Integration unabhängiger Multimedia-Objekte in eine synchronisierte Präsentation
- ***CML (Chemical Markup Language)*** : Beschreibung chemischer Notationen (eine der ersten XML-Applikationen), Vokabular umfaßt *Molekül, Atom, Bindung, Kristall, Formel, Reaktion, ...*

### 3.4 Namensräume (Namespaces)

- **Motivation:** *Wiederverwendung* einer (möglicherweise größeren) *Sammlung von Elementtyp- und Attributtypnamen*, die in einem bestimmten Kontext bereits eingeführt ist (etwa für die Formulierung mathematischer Gleichungen)
- **Problem:** unterschiedliche DTD können nicht ohne weiteres für ein Dokument kombiniert werden
- **Was wird benötigt ?**  
Mechanismus, der Namen eindeutig mit einer Interpretation verbindet (gegeben durch die „Herkunft“ des Namens)
- **Idee:** Bindung einer Namensmenge an eine URI, die der Kontextunterscheidung bei gleichen Namen dienen kann

## Namespace-Deklaration:

- erfolgt in Form einer Attributangabe im *start tag* eines Elements
- Einführung einer Kurzbezeichnung für die den Namespace auszeichnende URI

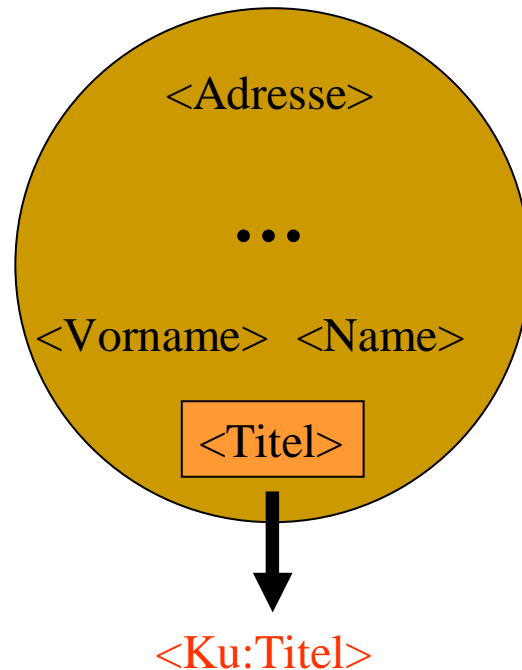
## Beispiel:

```
<Bestellung xmlns:Ku="http://www. ... /Kunde/"  
            xmlns:Bu="http://www. ... /Buch/"  
> ...  
</Bestellung>
```

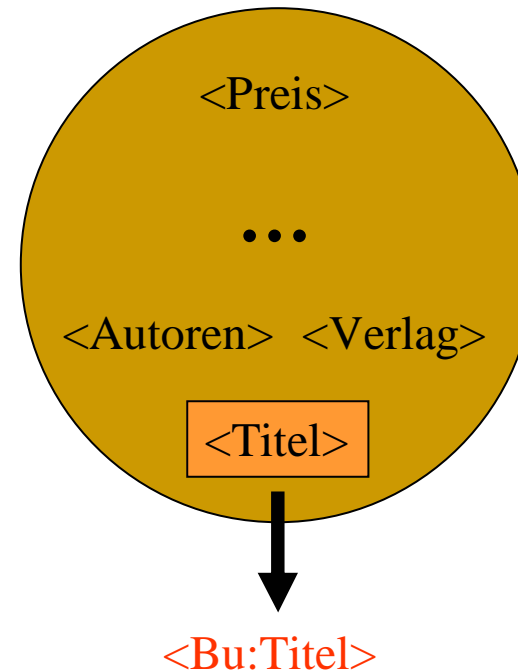
Schlüsselwort      Alias      URI

- Deklaration zweier Namensräume im Kontext eines Elements **Bestellung** (ein kundenspezifischer und ein buchspezifischer)
- mögliche Qualifizierung von Element- bzw. Attributnamen durch **Ku:** bzw. **Bu:**

Namensraum "Ku"



Namensraum "Bu"



- Bei Verwendung eines Namespace-Präfix vor einem Element-Namen wird dieser Präfix implizit vor allen Element-/Attributnamen innerhalb des betreffenden Elements angenommen, sofern nicht explizit ein anderer Präfix angegeben ist.
- Die Deklaration eines *default namespace* ist möglich.

## Dokumentfragment:

```
<Bestellung xmlns:Ku ...  
            xmlns:Bu ...  
...  
    <Ku:Kunde>  
        <Titel>Dr.</Titel>  
        ...  
    </Ku:Kunde>  
    ...  
    <Bu:Bücher>  
        <Buch>  
            <Titel>Learning XML</Titel>  
        </Buch>  
    </Bu:Bücher>  
</Bestellung>
```