
XML und XML Schema

Universität Leipzig

Problemseminar Semantic Web
2003 Wintersemester

Xuân Baldauf

Themen

- Warum XML?
 - Syntax
 - Semantik
- Warum Namensräume?
- Wozu XML Schema?
- Was hat XML mit Semantic Web zu tun?

Information mit **Steuerzeichen**

Es war einmal ein Text mit verschiedenen Schrift-Auszeichnungen:

„Warum *bebrüten* **Vögel Eier**?“

Drucker musste wissen, wann er anders drucken soll:

„Warum 4bebrüten EVögel 5 Eier F?“

Kommandos mit („Escape“) sind

- unhandlich, nur computer-lesbar
- nur inkompatibel erweiterbar
- prozedural

Markup Language

„Warum `<i>`bebrüten ``Vögel`</i>` Eier``?“

- solche Markierungen sind
 - beschreibend statt prozedural
 - lesbar für Computer und Mensch
 - erweiterbar
 - Das Ende einer Markierung ist immer gleich: '>'

Attribute

„Warum <i>bebrüten
Vögel</i> Eier?“

- Markierungen sollen parametrisch sein
- deswegen: **Attribute**
 - Theoretisch käme man ohne Attribute aus:
„<color>red</color><content>Warum
<i>bebrüten Vögel</i> Eier?
</content>“
 - Ist aber nicht aufwärts-kompatibel: Gehört „red“ zum Inhalt oder zur Markierung?

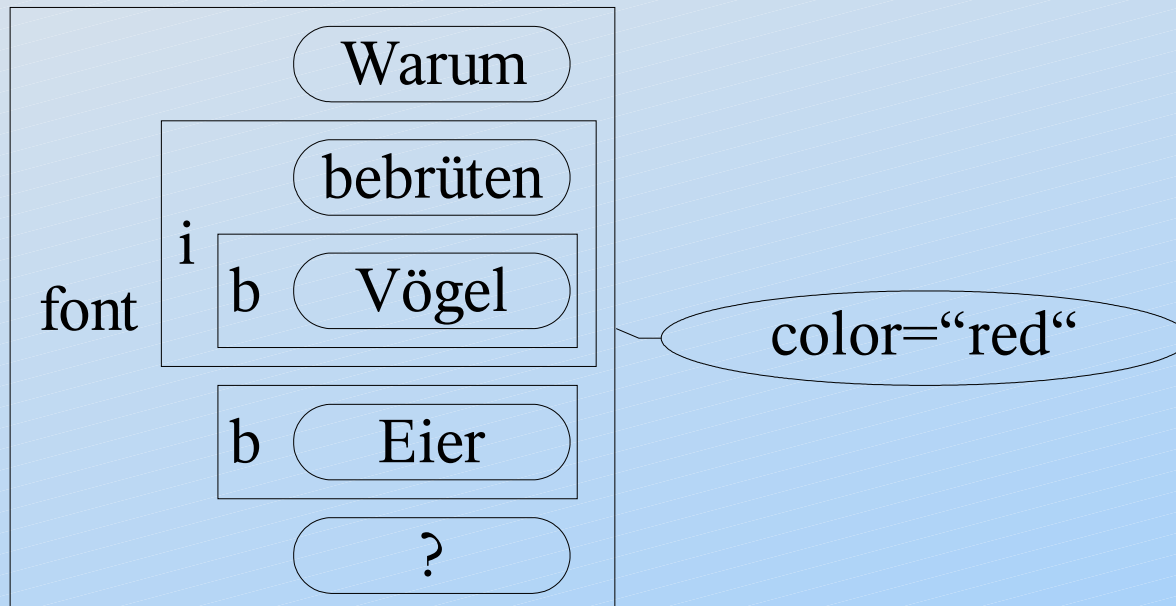
eXtensible Markup Language

- Standard für Markierungen
- mit vielen Restriktionen
 - Attribut-Werte müssen in Anführungszeichen angegeben sein
 - Markierungen müssen verschachtelt sein
 - ...

„Warum <i>bebrüten
Vögel </i> Eier?“

Madroschka

- Verschachtelungs-Restriktion führt zur Eigenschaft:
- Jedes XML-Dokument ist wie eine Madroschka
- Jede solche Verpackung heißt **Element**



Element

- Jedes Element hat einen Namen
 - z.B.: „font“
- Jedes Element hat eine Menge von Attributen
 - z.B.: {(„color“, „red“)}
- Jedes Element enthält eine Liste von **Kind-Knoten**
 - z.B.: („Warum “, i, b, “?“)
- Jedes Element ist ein Knoten.
- Jeder Text ist ein Knoten.

nomen est omen

(„Der Name ist ein Vorzeichen“)

- Definiert man, dass Elemente und Attribute mit dem selben **Namen** die selbe **Bedeutung** haben
 - „“ bedeute „Festlegung der Schrift“
 - „“ bedeute „Festlegung der Schriftfarbe“
 - „<i>“ bedeute „Kursivschrift“
 - „“ bedeute „Fettschrift“
- dann erlaubt ein XML-Dokument **formalisierte Beschreibung**

Objekt-orientierte Beschreibung

- Element-Name entspricht Objekt-Klasse
- Element-Vorkommen entspricht Objekt-Vorkommen
- Element-Lokalität (in der Madroschka) entspricht Objekt-Lokalität (in seiner Modell-Umwelt)

```
<einHaus>
```

```
  <raum name="Küche">
```

```
    <kühlschrank />
```

```
    <herd />
```

```
  </raum>
```

```
</einHaus>
```

Objekt-orientierte Beschreibung

- XML: Eignung für die Beschreibung von Instanzen.?
- Viele Objekte haben „lokale“ Beziehungen
 - Jeder Tisch hat Beine.
 - Jeder Tisch hat eine Tischplatte.
 - Jede Tischplatte hat eine Farbe.
- Unterstützung von Lokalität durch Schachtelung
- Angabe von Objekteigenschaften durch Schachtelung oder Attribute

Soll der Computer das Publikum befragen?

- Was beschreibt das Element `<matrix>`?
 - (a) ein lineares Gleichungssystem?
 - (b) interzelluläres Stützgewebe?
 - (c) einen Teil einer speziellen prädikatenlogischen Formel?
 - (d) ein Multi-User-Realitäts-Simulations-Programm?
- Was beschreibt das Element
 - `<math:matrix>`? `<bio:matrix>`?
 - `<logic:matrix>`? `<cinema:matrix>`?

Namensraum

- Ein Namensraum ist eine eindeutige Zuordnung von Element- und Attribut-Namen zu ihrer Bedeutung.
- Jeder Namensraum hat eine eindeutige Adresse (URI)
- Angabe von Bindung von Prefixen an Namensräume in jedem XML-Dokument und -Element möglich
 - `<matrix xmlns:mathml='http://www.w3.org/1998/Math/MathML'>`
 `<mathml:matrixrow>`
 `<mathml:cn>1.5</mathml:cn>`
 `</mathml:matrixrow>`
 `</matrix>`

Gültigkeit von Objekt-Beschreibungen

- Die Bedeutung der Element-Namen schreibt vor, dass bestimmte Dinge unmöglich sind.
 - möglich: `<telefongespräch dauer="0h13m48s">`
 - möglich: `<container masse="1863.3kg">`
 - unmöglich: `<telefongespräch masse="2.3kg">`
 - unmöglich: `<container masse="0h5m29s">`
- Computer: kein intuitives Verständnis für Gültigkeit von Informationen
- daher: ausdrückliche Gültigkeitsangabe

Einschränkungen von Text

- möglich: `<container masse="1863.3kg">`
- unmöglich: `<container masse="0h5m29s">`
- `<xsd:complexType name="container">`
 - `<xsd:attribute name="masse">`
 - `<xsd:restriction base="xsd:string">`
 - `<xsd:pattern value="[0-9]*.[0-9]*kg"/>`
 - `</xsd:restriction>`
 - `</xsd:attribute>`
- `</xsd:complexType>`

Einschränkungen von Attributen

- möglich: `<container masse="1863.3kg">`
- unmöglich: `<container dauer="0h5m29s">`
- `<xsd:complexType name="container">`
 `<xsd:attribute name="masse">`
 `<xsd:restriction base="xsd:string">`
 `<xsd:pattern value="[0-9]*.[0-9]*kg"/>`
 `</xsd:restriction>`
 `</xsd:attribute>`
 `<!-- Das Attribut „dauer“ kommt hier nicht vor. -->`
`</xsd:complexType>`

Einschränkung von Kind-Elementen

```
<xsd:complexType name="Haus">
  <xsd:element name="raum" maxOccurs="unbounded">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="kühlschrank" type="Kühlschrank" minOccurs="0"/>
        <xsd:element name="herd" type="herd" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string"/>
    </xsd:complexType>
  </xsd:element>
</xsd:complexType>

<einHaus>
  <raum name="Küche">
    <kühlschrank />
    <herd />
  </raum>
</einHaus>
```

XML Schema

- [eks em el ski:m] ist eine Sprache zur Beschreibung von Restriktionen in XML-Dokumenten
- kennt Daten-Typen (Klassen)
- kennt Ableitung (Vererbung)
 - durch Erweiterung um Elemente
 - durch zusätzliche Einschränkung
- kennt Polymorphie
- keine keine Kapselung

einfache Datentypen

- sind alle durch Text darstellbaren Datentypen
- Instanzen können dort stehen, wo Text steht (Attribut-Werte und Text-Knoten)
 - string
 - integer
 - positiveInteger
 - long
 - float
 - double
 - boolean
 - dateTime
 - duration
 - anyURI
 - language
 - hexBinary
 - base64Binary
 - ...

Konstruktion einfacher Datentypen

- Jeder einfache Datentyp hat eine Sprache zulässiger Instanzen
- Schnitt (Vererbung durch Restriktion)
 - mit Sprachen regulärer Ausdrücke „pattern“
 - mit Aufzählung aller möglichen Instanzen „enumeration“
 - mit Intervallen („alle Integer von 2 bis inclusive 12“)
- Vereinigung
- Liste von Instanzen
 - z.B.: „2 3 5 8“ wäre gültige Instanz einer Liste von Integer

komplexe Datentypen

- beschreiben die Struktur eines Element-Inhalts
 - an welcher Stelle darf welcher Knoten stehen?
- Struktur-Operatoren
 - „sequence“: angegebene Knoten müssen vorgegebene Reihenfolge haben
 - „choice“: genau eines der angegebenen Knoten muss vorkommen
 - „all“: angegebene Knoten dürfen beliebige Reihenfolge haben, dafür nur maximal einmal vorkommen

Struktur-Operationen

- beschreiben eine reguläre Sprache, deren Zeichen XML-Elemente sind
 - „sequence“ bezeichnet den Verkettungs-Operator
 - „choice“ bezeichnet die (exklusive) Alternative
- Struktur-Operationen können benannt werden:
 - ```
<xsd:group name="frischGepresst">
 <xsd:sequence>
 <xsd:element name="früchte" type="Orangen">
 <xsd:element name="presse" type="Orangenpresse">
 </xsd:sequence>
</xsd:group>
```

# Verwendung von Struktur-Operationen

- `<xsd:complexType name="DittmeyerWerbeziel">`
  - `<xsd:sequence>`
    - `<xsd:element name="kundenwunsch">`
      - `<xsd:complexType>`
        - `<xsd:choice>`
          - `<xsd:group name="frischGepresst"/>`
          - `<xsd:element ref="valensina"/>`

# Vererbung durch Erweiterung

- Eine Münze sei eine Metallscheibe mit Prägung:
- ```
<xsd:complexType name="münze">  
  <xsd:complexContent>  
    <xsd:extension base="metallscheibe">  
      <xsd:sequence>  
        <xsd:element name="prägung" type="münzprägung"/>  
      </xsd:sequence>  
    </xsd:extension>  
  </xsd:complexContent>  
</xsd:complexType>
```


Vererbung durch Einschränkung

- Eine Kupfermünze sei eine Münze aus Kupfer
- ```
<xsd:complexType name="kupfermünze">
 <xsd:complexContent>
 <xsd:restriction base="münze">
 <xsd:sequence>
 <xsd:element name="stoff" type="kupfer"/>
 <xsd:element name="prägung" type="münzprägung"/>
 </xsd:sequence>
 </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

# Unzulänglichkeiten der Vererbung

- Es muss oft die komplette Typ-Definition wiederholt werden.
  - Nur bei der Erweiterung im Sinne von Verlängerung einer `<xsd:sequence/>` nicht.
- Keine Restriktion über Invarianten
  - Dies ist nicht ausdrückbar:
    - Ein Rechteck besteht aus Breite und Höhe
    - Ein Quadrat ist ein Rechteck, wobei Breite und Höhe gleich sind.

# Was nützt's?

- XML liefert einen Standard für die Strukturierung von Daten.
  - ermöglicht Lesbarkeit für Mensch und Rechner
- XML Schema liefert einen Standard für die Einschränkung von erlaubten Datenstrukturen
  - ermöglicht Interoperabilität zwischen Applikationen
- XML Namensräume liefern einen Standard für die Einbettung nicht vorhergesehenen Datentypen
  - ermöglicht die Erweiterung bestehender XML-Sprachen

# Wo fehlt's?

- Beschreibung über Invarianten nicht möglich
- Vererbung zu umständlich und nicht vollständig
  - Typ muss explizit angegeben werden:
  - `<geldbeutel>`
    - `<inhalt>`
      - `<münze type="kupfermünze">`
        - `<!-- ... -->`
      - `</münze>`
    - `</inhalt>`
  - `</geldbeutel>`

# Fazit

- XML Schema reicht aus zur Beschreibung der **Datenstruktur** von Objekten.
- XML Schema reicht nicht aus zur Beschreibung der **Struktur** von Objekten.
- XML Schema reicht damit erst recht nicht aus zur Beschreibung der **Bedeutung** von Objekten,
  - selbst wenn sich die Bedeutung von Objekten allein aus ihrer Struktur ergeben würde.

---

# XML Schema für Semantic Web

---

- automatischer Datenaustausch: ja
- automatisches Schlussfolgern: nein

# Literatur (URIs)

- <urn:isbn:3-827-31844-0>
- <http://www.w3.org/TR/xmlschema-0/>
- <urn:isbn:0-262-06232-1>, chapter 4
- <http://www.w3.org/XML/Schema>
- [http://ais.informatik.uni-leipzig.de/studium/vorlesungen/2003\\_ws/xml](http://ais.informatik.uni-leipzig.de/studium/vorlesungen/2003_ws/xml)
- Dieses Dokument:
  - [http://studium.baldauf.org/informatik/2003.WS/schwerpunkt/Problemseminar Semantic Web/Vortrag/Vortrag.pdf](http://studium.baldauf.org/informatik/2003.WS/schwerpunkt/Problemseminar%20Semantic%20Web/Vortrag/Vortrag.pdf)

# Literatur (traditionell)

- Stefan Mintert (Herausgeber): „XML & Co. Die W3C-Spezifikationen für Dokumenten- und Datenarchitektur“, Addison-Wesley, 2003
- David C. Fallside (Herausgeber): „XML Schema Part 0: Primer - W3C Recommendation, 2 May 2001“, World Wide Web Consortium, 2001
- Michel Klein, Jeen Broekstra, Dieter Fensel, Frank van Harmelen, Ian Horrocks (Autoren): „Ontologies and Schema Languages on the Web“ In: Dieter Fensel, James A. Hendler, Henry Lieberman, Wolfgang Wahlster (Herausgeber): „Spinning the Semantic Web“, MIT Press, 2003, Seite 95..140