

Kodierung von Dokumenten mit XML

XML Namespaces, DocBook und SVG

Anne Brüggemann-Klein

TU München

Einsatz von XML-Kodierung im Praktikum

Kodierung von **Daten**, z.B. Spielstand

→ eigenes Format

- oft als Datenanteil von Objekten
- auszuwerten bzw. zu interpretieren mit eigenen Programmen (XQuery, XSLT), oft als Methodenanteil von Objekten

Kodierung von **Text** zur Dokumentation

→ DocBook

- zu editieren und zu formatieren mit existierenden Werkzeugen (syntaxgesteuerter Editor, XSLT-Programm)

Kodierung von **Vektorgraphik**

→ SVG

- zu editieren und zu visualisieren mit existierenden Werkzeugen (evtl. Graphik-Editor, Web-Browser)

XML-Kodierung und Baummodell am Beispiel

Konzepte: Element, Attribut, Text, Tag, (eingebettetes) Markup, Markupsprache, Metasprache für Markupsprachen, Datenformat vs. Präsentationsformat, Unicode, UTF-8, Dokument vs Daten

- XML zur **Kodierung von Daten**: Spielstand, Mediathek von iTunes, ...
- **Extreme XML zur Kodierung von Dokumenten** [[..\..\xsltBeispieleBook\compBookFilters.xpr](#)]
 - hierarchische Struktur mit Querbezügen über Attribute vom Typ ID und IDREF
 - semantisch reich, normalisiert, bearbeitbar
- **Formatiertes Dokument (generiert mit XSLT)**
 - [[..\..\xsltBeispieleBook\compBook.pdf](#)]
 - [[..\..\xsltBeispieleBook\compBookHTML.htm](#)]
- Visualisierung als **Baummodell** [[..\..\xsltBeispieleBook\compBookModel.pdf](#)]

Der rote Faden

Präsentation Aufgabenblatt 1

Review Kodierung von Dokumenten und Daten: XML

Weiterer Baustein zur Kodierung von Dokumenten: XML-Namespaces

Spezifische XML-Anwendung: DocBook als LaTeX im Gewand von XML

Spezifische XML-Anwendung: SVG zur Kodierung von Vektorgraphik mit XML

Literatur



Weiterer Baustein Kodierung XML-Namespaces

XML Namespaces

Bisher: einfache Namen für Elemente und Attribute

Konfliktpotential bei Kombination von Anwendungen

- z.B. SVG-Graphik oder MathML oder XForms in HTML
- "nativ mehrsprachig": XSLT, XML Schema

Lösung um Namen eindeutig zu machen

- Kombiniere einfachen Namen mit URL zur eindeutigen Identifizierung (vergleichbar mit Paketnamen)
- Syntax von XML Namespaces: Indirektion über Präfixe

Achtung: Namensraum-Namen / URLs sind nur Symbole !

- die URLs müssen nicht auf eine echte Ressource zeigen
- selbst die Domänen müssen nicht registriert sein (sind es aber natürlich bei "offiziellen" Namensräumen)

Konstruiertes Beispiel: konzeptuell

Namen sollen den Namensräumen entsprechender Farbe zugeordnet werden

Namenraum für Kataloge: <http://www.catalogues.com>

Namensraum für Bücher: <http://www.books.com>

Namensraum für Personennamen: <http://www.names.com>

```
<Katalog AutorIn="ABK">  
  <Titel>Neuanschaffungen Mai 2000</Titel>  
  <Buch>  
    <Titel>Green Eggs and Ham</Titel>  
    <AutorIn>  
      <Titel>Dr.</Titel><NName>Seuss</NName>  
    </AutorIn>  
  </Buch>  
</Katalog>
```

Konstruiertes Beispiel: Syntax

XML Namespaces: Markierung von Element- und Attributnamen mit Namensraumnamen (URIs) **indirekt** über Präfixe, die passend deklariert werden

```
<cat:Katalog nm:AutorIn='ABK'  
  xmlns:cat='http://www.catalogues.com'  
  xmlns:bk='http://www.books.com'  
  xmlns:nm='http://www.names.com'>  
  <cat:Titel>Neuanschaffungen Mai 2000</cat:Titel>  
  <bk:Buch>  
    <bk:Titel>Green Eggs and Ham</bk:Titel>  
    <bk:AutorIn>  
      <nm:Titel>Dr.</nm:Titel><nm:NName>Seuss</nm:NName>  
    </bk:AutorIn>  
  </bk:Buch>  
</cat:Katalog>
```


Konstruiertes Beispiel: Syntax

XML Namespaces: Markierung von Element- und Attributnamen mit Namensraumnamen (URIs) **indirekt** über Präfixe, die passend deklariert werden

```
<cat:Katalog nm:AutorIn='ABK'  
  xmlns:cat='http://www.catalogues.com'  
  xmlns:nm='http://www.names.com'  
  xmlns:bk='http://www.books.com'>  
  <cat:Titel>Neuanschaffungen Mai 2000</cat:Titel>  
  <bk:Buch xmlns:bk='http://www.books.com/neu'>  
    <bk:Titel>Green Eggs and Ham</bk:Titel>  
    <bk:AutorIn>  
      <nm:Titel typ="honorary">Dr.</nm:Titel><nm:NName>Seuss</nm:NName>  
    </bk:AutorIn>  
  </bk:Buch>  
</cat:Katalog>
```

Konstruiertes Beispiel: Syntax

XML Namespaces: Markierung von Element- und Attributnamen mit Namensraumnamen (URIs) **indirekt** über Präfixe, die passend deklariert werden

```
<Katalog nm:AutorIn='ABK'  
  xmlns:bk='http://www.books.com'  
  xmlns:nm='http://www.names.com'>  
  <Titel>Neuanschaffungen Mai 2000</Titel>  
  <bk:Buch>  
    <bk:Titel>Green Eggs and Ham</bk:Titel>  
    <bk:AutorIn>  
      <nm:Titel>Dr.</nm:Titel><nm:NName>Seuss</nm:NName>  
    </bk:AutorIn>  
  </bk:Buch>  
</Katalog>
```

Standardisierte Namensräume (URLs)

DocBook	→ http://docbook.org/ns/docbook
SVG	→ http://www.w3.org/2000/svg
XLink	→ http://www.w3.org/1999/xlink
XHTML	→
XSLT	→ http://www.w3.org/1999/XSL/Transform

Spezifische XML-Anwendung

DocBook als LaTeX im Gewand von XML

DocBook: LaTeX in XML

Eine XML-Anwendung für technische Dokumentation und wissenschaftliche Arbeiten

Formal definiert durch Schema (DTD, Relax NG)

➤ Editierunterstützung

Standardisiert durch OASIS

Tools: Transformierbar nach HTML, PDF durch XSL(T)

Konfigurierbar (erweiterbar und einschränkbar)

Beispiel [CalendarXSystem](http://docbook.org/ns/docbook) (→ Namensraum <http://docbook.org/ns/docbook>)

Alternative vergleichbarer Größenordnung: TEI (Text Encoding Initiative, M. Sperberg-McQueen)

- Transkription literarischer Texte
- mit eigener Struktur (DTD) und eigenen XML-Tools

Spezifische XML-Anwendung

SVG zur Kodierung von Vektorgraphik mit XML

Graphik mit XML: SVG

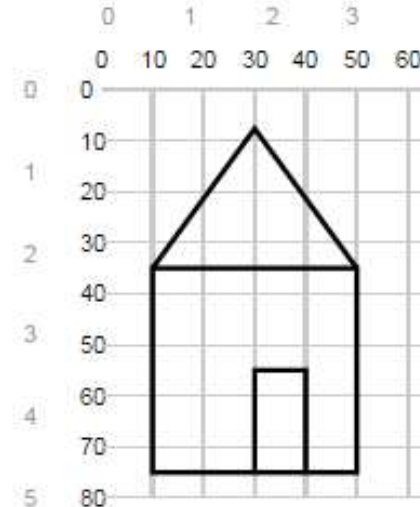
Scalable **Vector** Graphics (SVG),
W3C Recommendation
Version 1.1 (Second Edition): 16. August 2011

XML-Anwendung zur Kodierung von
Vektorgraphiken, Bestandteil von HTML5

- Beschreibung von graphischen Objekten mit XML-Vokabular
- Erfordert Viewer zum Anzeigen (skalierbare Anzeige)
 - spezielle Software oder Web-Browser
- Editieren (mit XML-Editor, Text-Editor oder Graphik-Editor möglich) oder Generieren

```
<svg width="4cm" height="5cm" viewBox="0 0 64 80">
  <!-- 16 user units per cm horizontally and vertically -->
  <title>House</title>
  <rect x="10" y="35" width="40" height="40"
    style="stroke: black; fill: none"/>
  <!-- roof -->
  <polyline points="10 35, 30 7.68, 50 35"
    style="stroke: black; fill: none"/>
  <!-- door -->
  <polyline points="30 75, 30 55, 40 55, 40 75"
    style="stroke: black; fill: none"/>
</svg>
```

Refresh Reset ☒ Show grid



Graphik mit XML: SVG

Funktionalität (Übersicht unter <http://tutorials.jenkov.com/svg/svg-examples.html>)

- mächtige Primitive (Bezier-Kurven, Farbverläufe, Koordinatensysteme)
- mächtige Operatoren (Gruppen, Transformationen, Filter)
- Animation, Interaktionselemente

Anwendungen

- genau vermaßte Graphiken (direkt definierbar)
- Graphiken aus "semantischen" Daten generierbar,
z.B. Geschäftsgraphiken aus statistischen Daten, Kalender, [Mancala-Spielbretter](#), Blackjack-Hand von Karten
- GUI-Elemente (Buttons, Logos)

Die Sprache SVG

Beispiele von [→](#) W3Schools

- Koordinatensystem und Viewport
 - Ursprung links oben, x nach rechts, y nach unten
 - Einheiten in User-Einheiten oder explizit (mm, in, pt, em, ...)
- Shapes
- Farbverläufe, Filter
- Animation

Beispiele von [→](#) SVG Essentials (<http://oreillymedia.github.io/svg-essentials-examples/>)

- Struktur, Namensraum
- Maßeinheiten, Koordinatensysteme
- Transformationen
- Links

Management von SVG-Graphiken

Systematischer Aufbau, Dokumentation durch Skizzen des graphischen Layouts mit Maßangaben

Benennung von Einheiten und Berechnungen
(z.B. mit Hilfe von Entities)

Wiederverwendung von Teilen einer Graphik mit `<use>`

Einbetten von SVG in SVG

Dokumentation von zusammengesetzten Objekten der Graphik durch Kommentare

Separierung von Style: Verwendung von CSS Stylesheets

Generierung aus semantischen Daten, unter Beibehaltung der oben angegebenen Techniken

Werkzeuge für SVG

Editieren mit Text-Editoren oder XML-Editoren

Native Unterstützung in Web-Browsern

Batik aus Apache XML (Viewer, SVG-Prozessor mit API)

Graphische Editoren mit optionaler Quelltext-Sicht

- Inkscape (freie Software)
- Sketsa (freie, unbegrenzte Evaluierungsversion)

Graphiksysteme mit Speichermöglichkeit in SVG

- Visio

Literatur

Literatur

T. Bray: XML Namespaces by Example (<http://www.xml.com/lpt/a/569>)

DocBook Wiki (<http://wiki.docbook.org/>)

Eric Raymond: DocBook Intro. 2010

(<http://en.tldp.org/HOWTO/DocBook-Demystification-HOWTO/>)

D. Rugge, M. Galassi, É. Bischoff: DocBook Crashcourse. 2006

(<http://opensource.bureau-cornavin.com/crash-course/>)

W3Schools Tutorial zu SVG (<http://www.w3schools.com/svg/>)

J.D. Eisenberg, A. Bellamy-Royds: *SVG Essentials*. O'Reilly 2014

(<http://oreillymedia.github.io/svg-essentials-examples/>)

J. Jenkov: Tutorial zu SVG (<http://tutorials.jenkov.com/svg/index.html>)

D. Dailey: An SVG Primer for Today's Browsers

(<https://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html>)