



HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

XML

Teil 3: Namensräume

Abteilung Informatik – WS 02/03

Namensräume: Motivation

Namensräume erlauben:

- Gleichnamige Elemente und Attribute verschiedener XML-Applikationen zu unterscheiden
 - ▶ Beispiel: Mit dem `set`-Element werden Attributswerte in SVG (Scalable Vector Graphics) gesetzt. Das `set`-Element definiert eine Menge im MathML
- Verwandte Elemente und Attribute einer XML-Applikation zu gruppieren

Namensräume und XML 1.0

- Namensräume sind nicht Bestandteil der XML-Spezifikationen 1.0
- Rückwärtskompatibilität wurde bei der Spezifikation der Namensräume sichergestellt: Ein XML-Parser, der Namensräume nicht unterstützt, wird (sollte) kein Problem haben, ein XML-Dokument mit Namensräumen zu bearbeiten

Beispiel: Gemäldeliste (1/4)

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<catalog>
  <painting>
    <title>Memory of the Garden at Etten</title>
    <artist>Vincent Van Gogh</artist>
    <date>November, <year>1888</year></date>
    <description>
      Two women look to the left. A third works in her garden.
    </description>
  </painting>
  <!-- Many more paintings... -->
</catalog>
```

Beispiel: Gemäldeliste (2/4)

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<catalog>
  <RDF>
    <description
      about="http://ibiblio.org/examples/impressionists.xml">
      <title> Impressionist Paintings </title>
      <creator> Elliotte Rusty Harold </creator>
      <description>
        A list of famous impressionist paintings </description>
      <date>2000-08-22</date>
    </description>
  </RDF>
  <painting>
    <title>Memory of the Garden at Etten</title>
    <artist>Vincent Van Gogh</artist>
    <date>November, <year>1888</year></date>
    <description>
      Two women look to the left. A third works in her garden.
    </description>
  </painting>
  <!-- Many more paintings... -->
</catalog>
```

1. XML-Applikation
(Research Description Framework)

2. XML-Applikation

Beispiel: Gemäldeliste (3/4)

- Elemente wie `title` und `date` haben unterschiedliche Inhaltsmodelle je nach ihrer Position im Dokument ⇒ Validierungsproblem
- Lösungsansätze:
 - ▶ Name der betroffenen Elemente ändern (`painting_title` statt `title`)
 - ▶ Jedem Element die URI der betroffenen XML-Applikationen zuordnen
 - ☺ Meistens kann eine XML-Applikation eindeutig durch eine URI identifiziert werden (Ausnahme: XSL mit XSLT und XSL-FO)
 - ☹ URIs enthalten meistens illegale Charaktere (`%`, `/`, `~`)

Namensräume: Syntax

- Syntax

Präfix : lokaler Teil

- Das Präfix und der lokale Teil bestehen aus legalen XML-Zeichen ausser dem Doppelpunkt. Präfixe, welche mit xml beginnen, sind reserviert (z.B. `xml:lang`-Attribut)

- Beispiel

`rdf : description`
└───┬───┘
Präfix lokaler Teil
 (engl. *local part*)

└────────────────────────────────┘
qualifizierte Name
(engl.: qualified name, QName, raw name)

URI eines Namensraums

- Jedem Präfix muss ein URI zugeordnet werden
- Deklaration des URIs mit dem Attribut `xmlns` im betroffenen Element oder in einem seiner „Ancestor“-Elemente

- Beispiel:

```
<rdf:RDF xmlns:rdf="http://www.w3c.org/TR/REC-rdf-syntax#">
  <rdf:Description
    ...
  </rdf:Description>
</rdf:RDF>
```

- Elemente (oder Attribute) ohne Präfix sind in keinem Namensraum

Beispiel: Gemäldeliste (4/4)

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<catalog>
  <rdf:RDF xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax#"
    <rdf:Description xmlns:dc="http://purl.org/dc/"
      about="http://ibiblio.org/examples/impressionists.xml">
        <dc:title> Impressionist Paintings </dc:title>
        <dc:creator> Elliotte Rusty Harold </dc:creator>
        <dc:description>
          A list of famous impressionist paintings organized
          by painter and date
        </dc:description>
        <dc:date>2000-08-22</dc:date>
      </rdf:Description>
    </rdf:RDF>
    <painting>
      <title>Memory of the Garden at Etten</title>
      <artist>Vincent Van Gogh</artist>
      <date>November, <year>1888</year></date>
      <description>
        Two women look to the left. A third works in her garden.
      </description>
    </painting>
    <!-- Many more paintings... -->
  </catalog>
```

Default-Namensräume

- Durch die Deklaration eines `xmlns`-Attributs ohne Präfix wird der Namensraum dieses Elements und jedes Nachfolger-Elements „per Default“ definiert
- Beispiel

```
<svg xmlns="http://www.w3.org/2000/svg/">
  <ellipse rx="110" ry="130"/>
</svg>
```
- Default-Namensräume betreffen nur Elemente, nie Attribute
- Default-Namensräume betreffen nur Nachfolger-Elemente ohne Präfix

Default-Namensräume: Beispiele

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:xlink="http://www.w3.org/1999/xlink">
  <head><title>Three Namespaces</title></head>
  <body>
    <h1 align="center">An Ellipse and a Rectangle</h1>
    <svg xmlns="http://www.w3.org/2000/svg"
         width="12cm" height="10cm">
      <ellipse rx="110" ry="130" />
      <rect x="4cm" y="1cm" width="3cm" height="6cm" />
    </svg>
    <p xlink:type="simple" xlink:href="ellipses.html">
      More about ellipses
    </p>
    <p xlink:type="simple" xlink:href="rectangles.html">
      More about rectangles
    </p>
    <hr/>
    <p>Last Modified May 13, 2000</p>
  </body>
</html>
```

Namensräume und DTD

- Namensräume haben keinen Impact auf DTD.
- Die Namen eines Elements oder eines Attributs im XML-Dokument und in der DTD müssen genau dieselben sein
- Insbesondere darf eine DTD das Präfix eines Elements oder eines Attributs nicht weglassen
- Umgekehrt: Die Namen eines Elements mit Default-Namensraum darf nicht mit Präfix in der DTD deklariert werden

Präfixänderung in der DTD

- Problemstellung: Präfixe müssen in der DTD deklariert werden ⇒ Eine Änderung des Präfixes muss in jeder Deklaration, die dieses Präfix verwendet, berücksichtigt werden
- Ansatz, um dieses Problem zu vermeiden:
 - ▶ Namensraum und Doppelpunkt als Parameterentität definieren

```
<!ENTITY %dc-prefix "dc">
<!ENTITY %dc-colon ":">
```
 - ▶ Parameterentitätsreferenz für die qualifizierten Namen definieren

```
<!ENTITY % dc-title "%dc-prefix;%dc-colon;title">
```
 - ▶ Parameterentitätsreferenzen in der Deklaration der qualifizierten Namen verwenden

```
<!ELEMENT %dc-title; (#PCDATA)>
```

Referenz

- E. Harold, W. Means: *XML in a Nutshell*, Kapitel 4