

Vorlesung Datenbanksysteme II

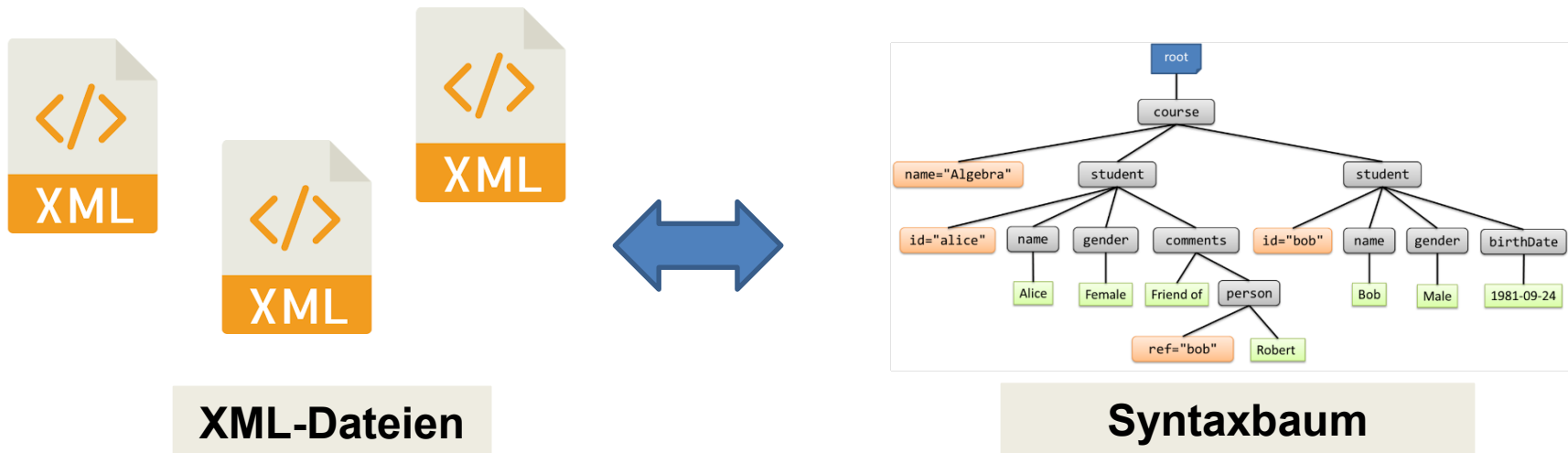
XML-Namensräume

Inhalt

- Motivation und Grundlagen
- Namensraumbezeichner
- Vorgegebene Namensräume
- Namenslose Namensräume

Motivation und Grundlagen

Motivation



- Bisher: 1 XML-Dokument \Leftrightarrow 1 Syntaxbaum.
- Jetzt: Mischen mehrerer Eingabe-Dateien aus verschiedenen Quellen \Rightarrow Namenskollisionen.

Beispiel: Namenskonflikte in Transportdateien

```

<Kapitel>
  <Titel>Vorträge im Mai 2001</Titel>
  <Vortragstermin Datum="5.5.2001">
    <Vortragender Titel="Dr."
      Name="Klein" Vorname="Karl" />
    <Vortrag Dauer="60">
      <Titel>Einführung in XML</Titel>
    </Vortrag>
    <Videoaufzeichnung>
      <Titel>Vortrag von Herrn Klein am 5.5.2001</Titel>
      <Dauer Einheit="sek">2200</Dauer>
    </Videoaufzeichnung>
  </Vortragstermin>
</Kapitel>

```

Unkritisch:
Kollision von
Attributnamen

Problematisch:
Kollision von
Elementnamen

Bisher: ein (impliziter) Namensraum (pro Datei).

XML-Namensräume

- XML-Namensräume werden in einer ergänzenden Spezifikation der XML beschrieben (1999-01-14)
<http://www.w3.org/TR/REC-xml-names>
- Konservative Erweiterung: „Anbau“ an die bisherigen Grammatikregeln.
- Achtung: Zum Teil große Unterschiede zu Namensraum-Konzepten von gängigen Programmiersprachen.

Exkurs: Namensräume in Programmen

- Innerhalb eines **Namensraums**: Verbot, den gleichen Bezeichner lokal mehrfach zu deklarieren.
- Definiert zugehörigen **Sichtbarkeitsbereich** eines Namensraums im Programm (z.B. anhand der Blockstruktur).
- Überlappende Sichtbarkeitsbereiche: **Sichtbarkeitsregeln** entscheiden bei Namenskollisionen (Beispiel: Overriding in Java)

XML-Namensräume

- XML-Namensräume werden **explizit deklariert** und **explizit benannt** (in Programmen zumeist implizit und ohne Namen).
- Deklarationen werden manuell zu XML-Namensräumen zugeordnet.
- Namen für XML-Namensräume sind **universell eindeutig**; somit auch universell eindeutige Typnamen durch **expliziten Bezug** auf einen Namensraum (“qualifizierte” Typnamen).

XML-Namensräume

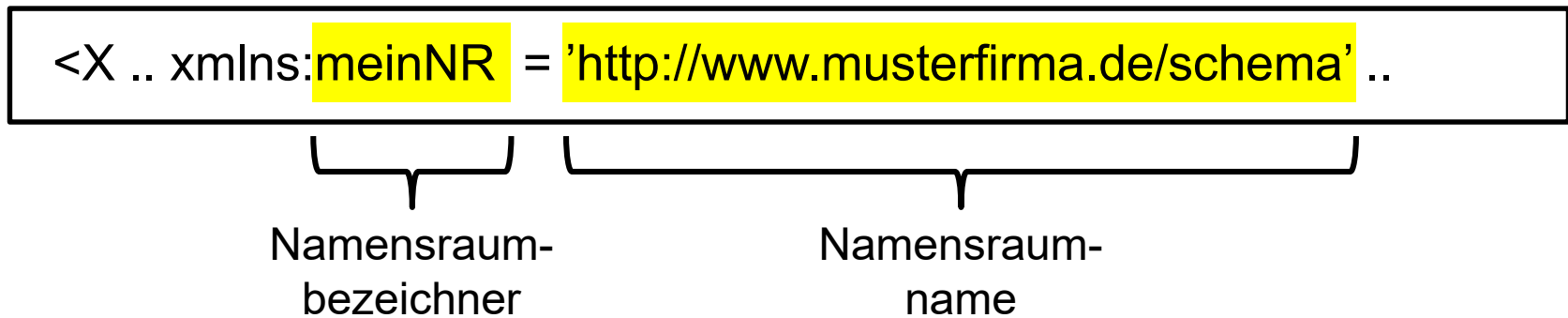
- Auch in Deklarationen müssen die qualifizierten Typnamen verwendet werden.
- Somit keine echten Namenskonflikte.
- Unqualifizierte Namen sind ein Sonderfall.

Namensraumbezeichner

Namensraumbezeichner

- Namen für Namensräume sollen universell eindeutig sein.
- Naheliegende Lösung: URI (Uniform Resource Identifiers), aber: sehr lang und unzulässige Zeichen (z.B. „/“).
- Stattdessen in der Transportdatei nur (kompakte) Namensraumbezeichner verwenden.

Deklaration von Namensraumbezeichnern



- Deklaration im öffnenden tag eines Elements als Attribut mit Namenspräfix „xmlns:“.
- Sichtbarkeitsbereich: das Element selbst und dessen innere Elemente und Attribute (also der ganze Teilbaum im Syntaxbaum).

Namensraumnamen

```
<X xmlns:meinNR='http://www.musterfirma.de/schema' .. >  
...  
  <!-- innerhalb dieses Elements ist der  
        Bezeichner meinNR an die Zeichenkette  
        'http://www.musterfirma.de/schema' gebunden -->  
...  
</X>
```

Namensraumname muss Syntax von URIs haben,
wird aber nicht als URI interpretiert.

Verwendung von Namensraumbezeichnern

NR-Bezeichner im
ganzen tag nutzbar

```

<meinNR:X
  xmlns:meinNR='http://www.musterfirma.de/schema' ... >
  ...
  <meinNR:E ... A='aaa'> .... </meinNR:E>
  ...
  <F ... meinNR:B='bbb'> .... </F>
</meinNR:X>
  
```

- Qualifizierter Typname: „NR-Bezeichner:lokaler Typname“.
- Wichtig: Auch in der DTD qualifizierte Typnamen benutzen.

Deklaration mehrerer Namensraumbezeichner

```
<X xmlns:meinNR ='http://www.musterfirma.de/schema'  
  xmlns:htmlNR ='http://www.w3.org/TR/REC-html40' ..>  
...  
</X>
```

Beliebige Anzahl verschiedener Deklarationen möglich.

Mehrere Bezeichner für den gleichen Namensraum

```
<X xmlns:meinNR    = 'http://www.musterfirma.de/schema'  
  xmlns:nocheinNR = 'http://www.musterfirma.de/schema'>  
  ...  
  <meinNR:E> .... </meinNR:E>  
  <nocheinNR:E> .... </nocheinNR:E>  
</X>
```

- Groß- / Kleinschreibung relevant.

Vorgegebene Namensräume

Vorgegebener Namensraum

- Qualifizierte Typnamen sind der Normalfall.
- Zusätzlicher Schreibaufwand lästig, falls überwiegend oder ausschließlich ein bestimmter Namensraum verwendet wird.
- Lösung: vorgegebener Namensraum (und Namensraumname).

Vorgegebener Namensraum

```
<X xmlns='http://www.musterfirma.de/schema' ... >  
  ... <E>.....</E> ...  
</X>
```

- Syntax: „xmlns=VorgabeNR-Name“.
- Wird implizit auf alle nichtqualifizierten (Element-) Typnamen innerhalb dieses Elements angewendet.

Vorgegebener Namensraum und Attribute

“default namespaces do not apply directly to attributes; the interpretation of unprefixed attributes is determined by the element on which they appear.”

Vorgegebener Namensraum und Attribute

- Beispiele: Parameter „match“ und „select“ in XSLT.
- Attribute repräsentieren “normalerweise” keine eigenständigen Entitäten, die aus verschiedenen Quellen gemischt werden.
- Explizite Zuordnung zu einem Namensraum ist möglich, aber ungewöhnlich.

Namenslose Namensräume

Der namenlose Namensraum

- Implizit vorhandener Namensraum, der keinen Namensraumnamen hat.
- Nur durch nichtqualifizierte Typnamen benutzbar.
- Für den Fall, dass ganz oder teilweise ohne Namensräume gearbeitet wird.

Beispiel

```
<root>
  <E1>
    .....
    <E2>.....</E2>
  </E1>

  <E3 xmlns='http://www.xxx.de'>
    .....
    <E2>.....</E2>
  </E3>
</root>
```

Die beiden „E2“-Elemente haben verschiedene Typen.

Beispiel

```

<root>
  <E1>
    .....
    <E2>.....</E2>
  </E1>

  <E3 xmlns='http://www.xxx.de'>
    .....
    <E2 xmlns="">.....</E2>
  </E3>
</root>
  
```

Abschalten den Namensraum-
names durch „xmlns="" “

Die beiden „E2“-Elemente haben nun den gleichen Typ.

Prüfungsstoff

- Konzepte von XML-Namensräumen erklären und anwenden können.
- Wesentliche Unterschiede zwischen XML-Namensräumen und den Namensraum-Konzepten gängiger Programmiersprachen diskutieren können.