



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

■ **Edirom Summer School 2010**

Workshop: XML und XML-Technologien
21.-24. November 2010

■ **M.Sc. Julian Dabbert**

Fakultät für Kulturwissenschaften der Universität Paderborn

■ Block 4: Abfragesprachen

■ Prinzip Abfragesprache

Eine Abfragesprache dient der Suche nach Informationen. Das Ergebnis einer Abfrage („Query“) ist eine Teilmenge der angefragten Datenbasis.

- Datenbasis kann komplexe Datenbank oder nur ein einfaches (XML-) Dokument sein
- Beispiele: Erweiterte Suchabfrage (Google)



The screenshot shows a search bar with the text "anfrage xml \"xpath\" -gepardenforelle". To the right of the search bar is a button labeled "Suche". Below the search bar, it says "Ungefähr 7.170 Ergebnisse (0,15 Sekunden)" and "Erweiterte Suche". At the bottom, there is a link "Seiten auf Deutsch" with a small icon to its right.

Basis: das von der Suchmaschine indizierte Internet

Filterbedingungen: Oder-Liste, Exakttreffer, Negation, Sprache

Ergebnis: 7.170 Weblinks

Block 4: Abfragesprachen

Abfragesprache XPath

Eine weitverbreitete Abfragesprache für XML ist XPath

- Version 1.0 seit 1999, Version 2.0 seit 2007
- Grundlage für andere Standards:
XSLT, XPointer, XQuery
- Navigation im Dokument auf dem XML-Baum über Achsen
(child, parent, self, ancestor...)
- Baumnavigation über einen Pfad ausgedrückt → (X)„Path“
- Keine Dateien, sondern Abfrageausdrücke

■ Block 4: Abfragesprachen

■ Lokalisierungsschritte in XPath

Ein XPath Ausdruck besteht aus:

- Einer Folge von Lokalisierungsschritten, getrennt durch ,/‘

Ein Lokalisierungsschritt besteht aus:

- Achsensschritt
- Knotentest
- Eine Folge von Prädikaten (optional)

Veranschaulichung: Wer hat blaue Augen?

- „Auf dem Personalausweis, auf der Rückseite, der Eintrag Augenfarbe – ist der Wert blau?“

Block 4: Abfragesprachen

Achsen in XPath

XPath navigiert in der logischen Baumstruktur über u.a. folgende Achsen:

- **parent:** direkt übergeordneter Knoten
- **child:** direkt untergeordneter Knoten
- **self:** der Kontextknoten selbst
- **ancestor:** ein (in-)direkt übergeordneter Knoten
- **descendant:** ein (in-)direkt untergeordneter Knoten
- **preceding:** ein im XML-Dokument vorhergehender Knoten
- **following:** ein im XML-Dokument nachfolgender Knoten
- **attribute:** Attributknoten

Beispiel: `/personalausweis/child::*[2]/descendant::datum`

■ Block 4: Abfragesprachen

■ Prädikate in XPath

Prädikate dienen der Filterung der Ergebnismenge

- **Werden in eckige Klammern gesetzt**
- **Können mehrmals auftreten, wobei die Reihenfolge wesentlich für das Ergebnis ist**
- **Können Funktionen auf Inhalten ausführen:**
`substring()`, `string-length()` ...
- **Können Funktionen auf Knotenmengen ausführen:**
`count()`, `name()` ...

Beispiel: `//rueckseite[./augenfarbe/text()='blau'][./groesse/text()='190']`

■ Block 4: Abfragesprachen

■ Beispiele für XPath Abfragen

Rückgabe von gesamten Knoten (samt Unterbäumen)

- `/personalausweis/rueckseite`
- `//rueckseite`

Rückgabe von Inhalten

- `/descendant::ablaufdatum/text()`

Rückgabe von boolschen Werten

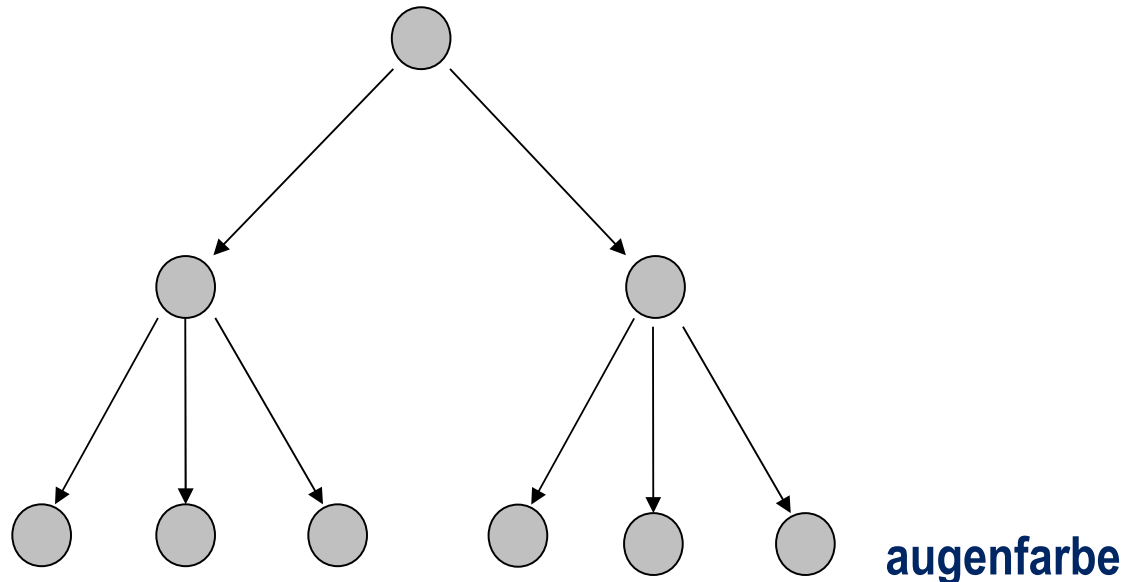
- `//groesse/text()=„190“`

■ Block 4: Abfragesprachen

■ Baumnavigation einer XPath Abfrage

Veranschaulichung: Wer hat blaue Augen?

- „Auf dem Personalausweis, auf der Rückseite, der Eintrag Augenfarbe – ist der Wert blau?“
- `/personalausweis/descendant::name[//augenfarbe/text()=„blau“]`



■ Block 4: Abfragesprachen

■ Übung: XPath Abfragen

Was ergeben diese Anfragen an den eigenen Personalausweis?

- `/descendant::vorname`
- `//name/text()`
- `//behoerde[text()='“behoerde“']`
- `//rueckseite[./augenfarbe/text()="braun"]`

Testen auf einem Testbett:

<http://www.whitebeam.org/library/guide/TechNotes/xpathtestbed.rhtm>

■ Block 4: Abfragesprachen

■ Übung: XPath Abfragen

Anfragen an den eigenen Personalausweis

- Wie lautet mein Nachname?
- Wie lautet meine Anschrift?
- Wie heisst das dritte Kind von „vorderseite“?
- Ist meine Augenfarbe „braun“?
- Ist meine Grösse genau 200 cm?

Testen auf einem Testbett:

<http://www.whitebeam.org/library/guide/TechNotes/xpathtestbed.rhtm>

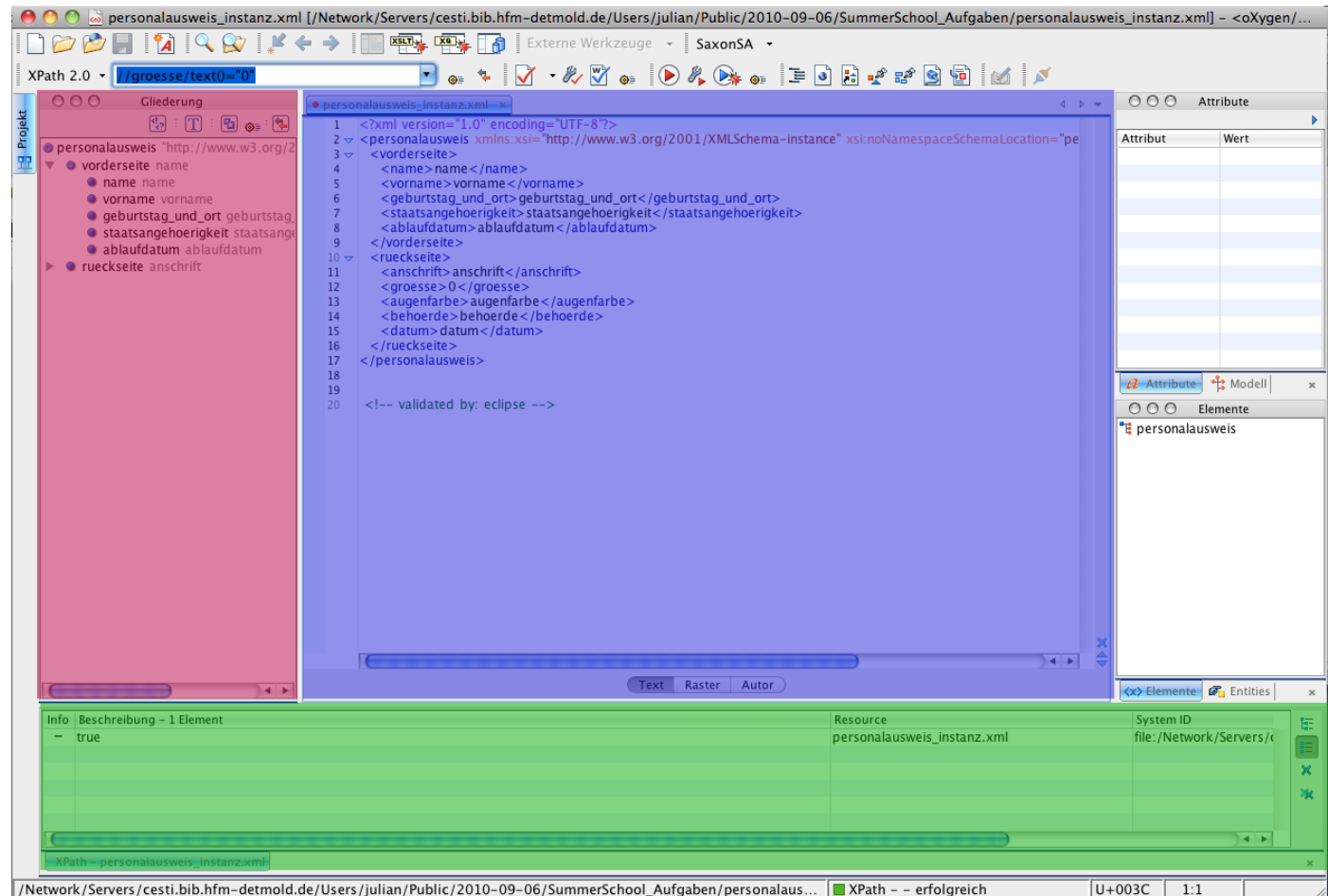
Block 4: Abfragesprachen

Vorstellung: oXygen XML Editor

Gliederung

Inhalt

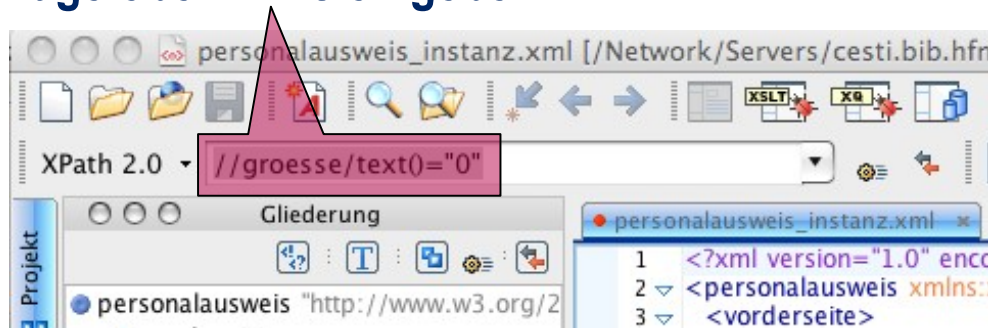
Ergebnisse



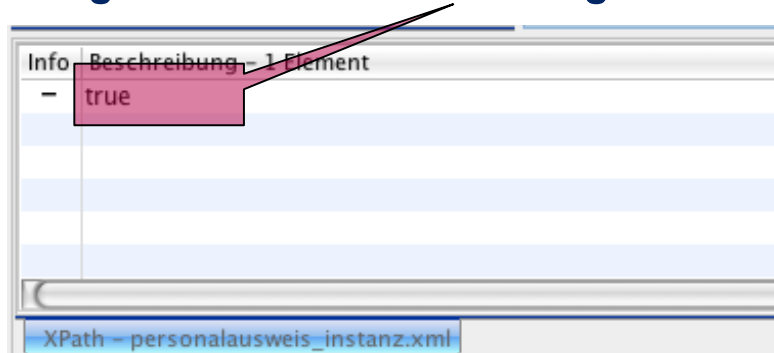
■ Block 4: Abfragesprachen

■ Vorstellung: XPath Abfragen mit oXygen

Der oXygen Editor hat eingebaute XPath Unterstützung (sehr nützlich)
Abfrage oben links eingeben:



Das Ergebnis wird unten im Ergebnisfenster angezeigt:



■ Block 4: Abfragesprachen

■ Übung: oXygen ausprobieren

Was ergeben diese Anfragen an den eigenen Personalausweis?

- `/descendant::vorname`
- `//name/text()`
- `//behoerde[/text()="behoerde"]`
- `//rueckseite[./augenfarbe/text()="braun"]`

Ergebnisauswertung:

- Query → `/descendant::vorname`
- Result → `/personalausweis[1]/vorderseite[1]/vorname[1]` - julian

■ Block 4: Abfragesprachen

■ XPath und XQuery

Die Abfragesprache XQuery

- benutzt für bestimmte Aufgaben XPath (XPath ist integriert)
- Ist „mächtiger“: kann grössere Menge an Aufgaben entscheiden
- Dateiendung: .xquery
- Die Abfragen funktionieren nach sog. „FLWOR“ (sprich flower)
Schema, ähnlich der „klassischen“ Abfragesprache SQL

for
let
where
order by
return

■ Block 4: Abfragesprachen

■ XQuery Beispiel: xquery_namecount.xquery

Abfrage:

```
for $doc in fn:collection('..?select=*.xml')
let $uri := fn:document-uri($doc)
where fn:count($doc//name) gt 0
order by fn:count($doc//name) descending
return
<document count="{fn:count($doc//name)}">
  {$uri}
</document>
```

①

②

Ergebnis:

```
<document count="2">file:/.../xslt_personen.xml</document>
<document count="1">file:/.../personalausweis_instanz.xml</document>
```



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft



Ende des Blocks 4