Einführung in XPath

Dr. Björn Rudzewitz

pagina GmbH

18.01.2023

pagina GmbH Publikationstechnologien -

Die Technologieexperten für das Publizieren im digitalen Zeitalter.

Nachhaltig.

Besuchen Sie unsere Webseite: https://pagina.gmbh/

Lernziele

Lernziele:

- ✓ einsteigerfreundliche Einführung in Grundlagen
- ✓ Anwendung und Zweck von XPath
- ✓ XPath und XML-Baumstruktur
- ✓ Struktur von XPath-Ausdrücken verstehen
- ✓ Ausdrücke selber schreiben

Lernziele

Nicht Teil der Lernziele:

- X komplexe, fortgeschrittene Ausdrücke mit Funktionen
- X Unterschiede zwischen Versionen von XPath
- X praktische Integration von XPath in Programme und Programmiersprachen
- X vollständige Beschreibung des Featureumfangs

Was ist XPath?

- Auswahlsprache für XML-Dokumente
- XML Inhalte Anwendungen Menschen und Anwendungen zugänglich machen
- XPath als Basis vieler fortgeschrittener XML-Technologien (z.B. XML Schema, XSLT, XQuery, . . .)
- viele Programmiersprachen unterstützen XPath (Java, Python, XSLT, . . .)
  - z.B. Selektion und Verarbeitung von XML-Dokumenten nach Einlesen in Java

XML-Baumstruktur

- XML-Dokumente haben eine Baumstruktur (*“document tree”*)
- XPath sucht Elemente im Baum relativ zu anderen Elementen
- z.T. Ähnlichkeiten zu UNIX-Dateisystembaum

Knoten

- XPath arbeitet mit Knoten (“*nodes*”) und nicht unmittelbar mit XML-Elementen, Attributen, . . .
- XPath sieht jede XML-Komponente als Knoten: Elemente, Attribute, Text, Dokument, . . .
- “*node set*”: Menge an Knoten

Knoten I

Knotenarten:

- **Wurzelkonten**: beinhaltet *alles*, inkl. Wurzel*element*, Verarbeitungsanweisungen, Kommentare, ...
- **Elementknoten**: entspricht einem XML-Element
- **Attributknoten**: hängt an Elementkonten und repräsentiert XML-Attribut
- **Textknoten**: stetiger (nicht unterbrochener) textueller Inhalt (#PCDATA)

Knoten II

Knotenarten:

- **PI-Knoten**: Verarbeitungsanweisungsknoten
- **Kommentarknoten**: für XML-Kommentare
- **Namensraumknoten**

Knoten II

Knotenarten:

- **PI-Knoten**: Verarbeitungsanweisungsknoten
- **Kommentarknoten**: für XML-Kommentare
- **Namensraumknoten**

$\Rightarrow$ XPath sieht Dokumente als *Knoten*

Beziehungen zwischen Knoten I

- aktueller Knoten, **Kontextknoten** (“*self*”)
- **Eltern**, **Kind**, **Geschwister**
- **übergeordnet**: Elternknoten, Elternknoten des Elternknotens, . . . (“*ancestor*”)

Beziehungen zwischen Knoten II

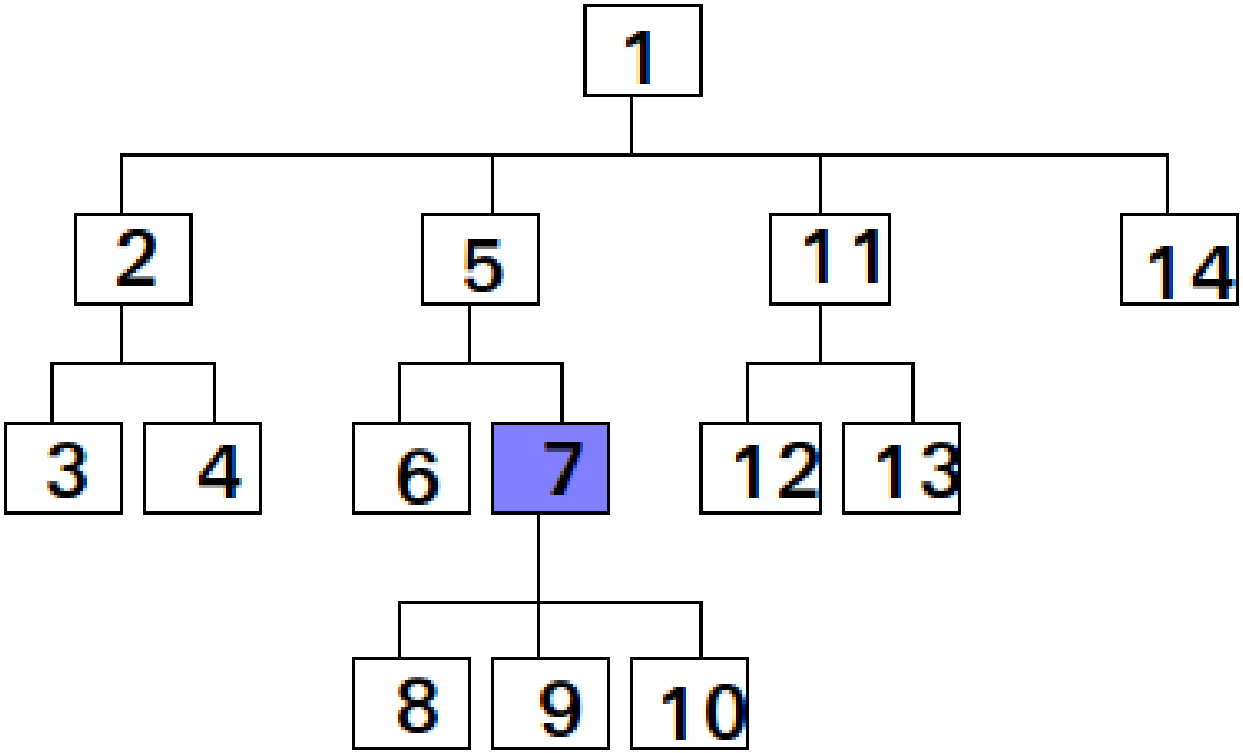
- **untergeordnet**: Kind, Kindeskinder, . . . (“*descendant*”)
- **Geschwister**: gleicher Elternknoten (“*sibling*”)

Beziehungen zwischen Knoten III

- **vorangehend**: Knoten, die vor dem aktuell ausgewählten Knoten durchschritten werden ohne Vorfahren (“*preceding*”)
- **nachfolgend**: Knoten durchschritten nach dem aktuellen Knoten ohne Nachfahren . . . (“*following*”)
- Kombinationen: untergeordnet-oder-selbst, . . . (“*descendant-or-self*”)

guter Überblick: `https://wiki.selfhtml.org/wiki/XML/XSL/XPath`

Beziehungen zwischen Knoten



Welche Knoten sind Eltern, übergeordnete, untergeordnete, vorangehende Knoten, nachfolgende Geschwisterknoten des aktuellen (blauen) Kontextknoten ?

Quelle: `https://wiki.selfhtml.org/wiki/Datei:Xml_parent.gif`, zuletzt aufgerufen am 17.01.2023

Ausdrücke

- XPath basiert auf dem Konzept (virtueller) **Achsen**
- Knoten liegen auf den virtuellen Achsen, mit unterschiedlichen Knotentypen auf unterschiedlichen Achsen
- Beispiel: ein Kindelementkonten liegt auf der Elementachse unter dem Elternkontenelement, ein Attribut liegt auf einer dazu geneigten Achse
- [Simpson, 2002, page 16]:
  an axis "turns the view" from a given starting point in the document

XPath Selektoren

Allgemeines Muster von XPath-Ausdrücken/Pfaden:

Achse::Knotentest[Prädikat]

- Achse: in welcher Richtung/Dimension
- Knotentest: Eigenschaften eines Knotens (Name, Typ, . . .)
- Prädikat: Verfeinerung der Suchergebnisse

XPath Selektoren

Allgemeines Muster von XPath-Ausdrücken/Pfaden:

Achse::Knotentest[Prädikat]

- Achse: in welcher Richtung/Dimension
- Knotentest: Eigenschaften eines Knotens (Name, Typ, . . .)
- Prädikat: Verfeinerung der Suchergebnisse

- meist werden mehrere solcher Ausdrücke aneinandergereiht
- nur der Knotentest ist explizit anzugeben

XPath-Selektoren

Knotentest:

- Auswahl Elemente nach Name: `/course/description`
- Auswahl aller Kindelemente: `/courses/*`
- Auswahl aller Textknoten: `/courses/description/text()`
  analog dazu: `comment()`, `processing-instruction()`
- Auswahl aller Knoten aller Art: `/courses/node()`

Übungsaufgabe

Übung

Aufgabe: Schreiben Sie einen XPath-Ausdruck, um die **Titel-Elemente aller Bücher** zu bekommen.

- Beispieldaten: `https://www.w3schools.com/xml/books.xml`.
- Online-Editor zum Ausprobieren: `http://xpather.com/`

Achse:

- **gegeben einen Kontextknoten durch den Knotentest**
- gibt eine Achse an, in welche Richtung/Dimension gesucht wird
- 13 Achsen
- spezielle Achsen nur von bestimmten Knoten erreichbar (z.B. Attributachsen nur bei Elementachsen)

```
/campus/person/attribute::node()
```

XPath-Selektoren

Achse

Achse	wählt aus (gegeben Kontextknoten)
child::	Kindelementknoten
parent::	Elternelementknoten
descendant::	Nachfahrenelementknoten
ancestor::	Vorfahrenelementknoten ...

XPath-Selektoren

Achse

Achse	wählt aus (gegeben Kontextknoten)
self::	aktuellen Kontextknoten
attribute::	Attributknoten
preceding::	vorangehende Elemente
preceding-sibling::	vorangehende Schwesterelemente

following::, following-sibling::, namespace::

Konventionen bei Achsen

Konventionen/Abkürzungen:

- Kindachse ist Standard:
  `child::course` äquivalent zu `course`
- Elternknoten: `parent::node()` äquivalent zu `..`
- aktueller Kontextknoten: `self::node()` äquivalent zu `.`
- `/descendant-or-self::node( )` äquivalent zu `//`
- `/attribute::` äquivalent zu `@`

Übungsaufgabe

Übung

Aufgaben:

1. Schreiben Sie einen XPath-Ausdruck, um **alle Attribut-Wertpaare aller Buchelemente** zu extrahieren.
2. Schreiben Sie einen XPath-Ausdruck, um alle **Autoren** eines Buches zu extrahieren, die **nicht an erster Stelle** stehen (d.h. Autoren, die anderen Autoren folgen):

- Beispieldaten: `https://www.w3schools.com/xml/books.xml`.
- Online-Editor zum Ausprobieren: `http://xpather.com/`

Prädikate

- Prädikate spezifizieren zusätzliche Bedingungen für das Matching von Knoten
- verfeinern die Suchergebnisse mithilfe zusätzlicher XPath-Ausdrücke
- spezifiziert in eckigen Klammern [ ]
- Muster:
  expr1 operator expr2
- expr1, expr2 sind XPath-Ausdrücke
- operator: =, !=, >, <, >=, <=
- operator expr2 sind optional

Prädikate

- Prädikate ergeben Boolesche Ausdrücke: wahr oder falsch
- es genügt, wenn der Wert einmal wahr ist
  - z.B. wenn eine Bedingung Kindelemente berücksichtigt, reicht es, wenn die Bedingung für eines der Kindelemente wahr ist, muss nicht für alle gelten

Prädikate

Beispiel aus [Simpson, 2002, Ch. 3.3]:

```
<person name="John">
   <child name="John"/>
   <child name="Connie"/>
   <child name="Cindy"/>
   <child name="Mike"/>
</person>
```

Prädikate

Beispiel aus [Simpson, 2002, Ch. 3.3]:

```
<person name="John">
   <child name="John"/>
   <child name="Connie"/>
   <child name="Cindy"/>
   <child name="Mike"/>
</person>
```

Der XPath-Ausdruck `/person[child/@name='Cindy']` gibt den Wurzelelementknoten zurück, da das Prädikat für eines der Kindelemente zutrifft

Prädikate

in der Praxis werden Ausdrücke mit Prädikaten oft verkettet:

/root/courses[@faculty='sfs']//information[@participants >10]

wählt alle "information" Elemente aus, bei denen einerseits das Attribut "participants" den Wert größer als 10 hat und die andererseits Nachfahren von "course" Elementen sind mit dem Attribut "faculty" mit dem Wert "sfs", welche wiederum Kindelemente sind des Wurzelelements mit dem Elementnamen "root"

Compound Predicates

```
<participants>
   <person name="John" age="20" />
   <person name="John" age="23"/>
   <person name="Paul" age="20"/>
   <person name="Emma" age="20"/>
</participants>
```

Compound Predicates

```
<participants>
   <person name="John" age="20" />
   <person name="John" age="23"/>
   <person name="Paul" age="20"/>
   <person name="Emma" age="20"/>
</participants>
```

- zusammengesetzte Prädikate mit `and` oder `or`
- runde Klammern für Gruppierung

```
/participants/person[@name="John"and @age="20"]
```

```
//person[(@name="John"or @name="Paul") and @age="20"]
```

Reduced Predicates

- Prädikate können Bedingungen unabhängig vom eigentlichen Wert festlegen
- Beispiel: `//metadata[descendant::license]`
  wählt diejenigen *metadata* Elemente aus, die ein *license* Element als Nachfahre haben
- abgekürzte Positionsangabe (nur wenn nicht in zusammengesetzem Ausdruck):
  //person[4] äquivalent zu //person[position() = 4]
  wählt das vierte *person* Element aus
- weitere Positionsangaben: `first()`, `last()`

Übungsaufgabe

Übung

Aufgaben:

1. Schreiben Sie einen XPath-Ausdruck, um die **Titelelemente** aller Bücher der **Kategorie web** zu extrahieren.
   - Beispieldaten: `https://www.w3schools.com/xml/books.xml`.
   - Online-Editor zum Ausprobieren: `http://xpather.com/`

Funktionen

- XPath erlaubt die Benutzung von Funktionen
- Beispiele:
  - arithmetische Operationen, z.B. `count()`, `sum()`
  - Textfunktionen, z.B. `text()`, `contains()`, `matches()`
- XPath-Funktionen haben immer einen Rückgabewert (nicht void)

Funktionen

XPath Datentypen:

- **string**: Text
- **nodeset**: XPath Ausdruck/Verweis auf Knoten, kann leer sein
- **boolean**: wahr/falsch
- **number**: Zahl
- **anytype**: beliebig/unbestimmt

siehe [Simpson, 2002, Kapitel 4.2]

Funktionen

Einige wichtige XPath-Funktionen auf Knotensets:

Funktion	Rückgabewert	Beispiel
position()	Position in Knotenset (1-basiert)	/participants/person [position() = 2]
last()	Anzahl Knoten	/participants/person[last()]
count(nodeset)	Anzahl Knoten	count(//person)

Funktionen

Einige wichtige XPath-Funktionen für Text:

Funktion	Rückgabewert	Beispiel
string(anytype)	Textrepräsentation (nur 1. Knoten für Sets!)	string(count(//person))
concat(a,b,...)	Textverkettung	concat(count(//person), "people")
contains(a,b)	beinhaltet a b	//person[contains(@name,"j")]

Funktionen

Einige wichtige XPath-Funktionen für Text:

Funktion	Rückgabewert	Beispiel
substring(a,1,2)	Teilzeichenkette von a von 1 bis 2	substring(//person[2]/@name, 1, 3

indexes start at 1 for XPath

Funktionen

Einige wichtige XPath-Funktionen für Zahlen:

Funktion	Rückgabewert	Beispiel
sum(nodeset)	Summe	sum(//person/@age)
floor(n)	Abrundungsfunktion	floor(1.3 * 4)
ceiling(n)	Aufrundungsfunktion	ceiling(1.3 * 4)
round(n)	Rundungsfunktion	round(1.4*3)

Hinweis: div zum Teilen anstelle von /,
+,-,* werden direkt verwendet

Übungsaufgabe

Übung

Aufgaben:

1. Schreiben Sie einen XPath-Ausdruck, um **alle Buchtitel** zu extrahieren von Büchern, die mehr als 35 kosten.
2. Schreiben Sie einen XPath-Ausdruck, um die **Namen aller Autoren** zu extrahieren, die den **Großbuchstaben "K"** enthalten.
3. Schreiben Sie einen XPath-Ausdruck, der den **Durchschnittspreis** über alle Bücher berechnet.

- Beispieldaten: `https://www.w3schools.com/xml/books.xml`.
- Online-Editor zum Ausprobieren: `http://xpather.com/`

Quellen und weiterführende Materialien

Simpson, John. XPath and XPointer: Locating Content in XML Documents. O'Reilly Media, Inc., 2002.

Die Struktur der Folien folgt lose diesem Buch und integriert Teile der Inhalte der Folien aus dem Text Technology Kurs von Prof. Chris Culy aus dem Jahr 2013. Die vorliegenden Folien wurden von Dr. Björn Rudzewitz erstellt und in mehreren Iterationen im SS 2017/18/19/20/21 an der Universität Tübingen zur Lehre verwendet.

Begriffe für deutsche Übersetzung: `https://www.homepage-webhilfe.de/XML/XPath/` (zuletzt aufgerufen am 14.01.2023)

Christopher Culy. Proseminar Text Technologies SS 13, 2013.

Elliotte Rusty Harold, W Scott Means, and Katharina Udemadu. *XML in a Nutshell*, volume 3. O'reilly Sebastopol, CA, 2004.

John Simpson. *XPath and XPointer: Locating Content in XML Documents*. Ö'Reilly Media, Inc.", 2002.