

---

# XPath 1.0

## *Pfadausdrücke zur Adressierung von XML-Fragmenten*

Holger Meyer

# Überblick

---

- Umfeld
- Datenmodell
- Pfadausdrücke
- Details

# XPath 1.0

---

- <http://www.w3.org/TR/xpath> (Nov 1999)
- Basis für eine Reihe weiterer W3C „Standards“
- XSL Transformations (XSLT)
- XML Link (XLink)
- XML Pointer (XPointer)
- XML Query (XQuery, aber als XPath 2.0)
- ursprünglich Bestandteil von XSL

# XPath 1.0

---

- Adressierung von Teilen eines Dokumentes,
- Selektion von Knotenmengen,
- Formulierung von Bedingungen an diese Knotenmengen,
- grundlegendes Konstrukt sind XPath-Ausdrücke:
  - Pfadausdrücke (*Location path*),
  - logische und mathematische Verknüpfungen,
  - Funktionsaufrufe
  - sind vom Typ her Knotenmengen oder Werte (boolean, number, string)

# XPath-Ausdrücke

---

- kann aus mehreren Teilausdrücken (Bausteinen, Schritten) bestehen,
- Kopplung erfolgt über ' / ' (analog Dateisystemkomponenten),
- ein Schritt liefert eine Knotenmenge oder Werte,
- es gibt absolute und relative Pfadausdrücke,
- Abarbeitung zusammengesetzter Ausdrücke erfolgt „von links“,

# Beispieldokument für Anfragen

---

```
<bib>
```

```
  <book><publisher>Addison-Wesley</publisher>
```

```
    <author>Serge Abiteboul</author>
```

```
    <author><first-name>Rick</first-name>
```

```
      <last-name>Hull</last-name></author>
```

```
    <author>Victor Vianu</author>
```

```
    <title>Foundations of Databases</title>
```

```
    <year>1995</year></book>
```

```
  <book price='55'><publisher>Freeman</publisher>
```

```
    <author>Jeffrey D. Ullman</author>
```

```
    <title>Principles of Database and Knowledge Base S
```

```
    <year>1998</year></book>
```

```
</bib>
```

---

# XPath — Datenmodell

---

- Grob: abstrakter Baum, Knoten sind Elemente eines Dokumentes, Kanten Subelementbeziehungen.
- Genau: sieben Knotenarten: Wurzelknoten, Elementknoten, Attributknoten, Namensraumangaben, Textknoten (`#PCDATA`), Kommentarknoten, Verarbeitungshinweise (Processing instructions, PIs)
- ursprüngliche Markup bleibt nicht erhalten,
- logische und physische Modularisierung geht verloren (Entities)

# XPath — Datenmodell

---

- Datentypen: atomare Werte (boolesche, numerische und Zeichenkettenwerte: `boolean`, `number`, `string`) und Knotenmengen (`node-set`)
- Kontext eines Pfadausdruckes: aktueller Knoten (Kontextknoten), die Position innerhalb des Kontextes, Größe des Kontextes, verfügbare Funktionen, Namensraumangaben

# Wurzelknoten und Dokumentelemente

---

- Beispiel: `<bib><paper>...</paper>...</bib>`
- `bib` ist das Dokumentelement (Achtung: besser nicht als „Wurzelelement“ bezeichnen!)
- Der Wurzelknoten liegt oberhalb des Dokumentelements.
- `/bib` liefert das Dokumentelement
- `/` liefert den Wurzelknoten
- Es kann Kommentare und PIs parallel zum Dokumentelement geben!

# XPath — Einfache Ausdrücke

---

- $Q_1$ : `/bib/book/year`

- Ergebnis:

`<year>1995</year>`

`<year>1998</year>`

- $Q_2$ : `/bib/paper/year`

- Ergebnis: leer, Es gab keine Papiere!

# XPath — Kleene'sche Hülle

---

- $Q_3$ : //author

- Ergebnis:

```
<author>Serge Abiteboul</author>
```

```
<author><first-name>Rick</first-name>
```

```
    <last-name>Hull</last-name>
```

```
</author>
```

```
<author>Victor Vianu</author>
```

```
<author>Jeffrey D. Ullman</author>
```

- $Q_4$ : /bib//first-name

- Ergebnis: <first-name>Rick</first-name>

- eingeschränkte Kleene'sche Hülle,  $0..n$  (transitive,  $1..n$ )

- $a//b \mapsto a/descendant-or-self::node()/b$

# XPath — Wildcard

---

- $Q_5$ : //author/\*

- Ergebnis:

```
<first-name>Rick</first-name>
```

```
<last-name>Hull</last-name>
```

- \* „matched“ beliebiges Element und @\* beliebiges Attribut

# XPath — Knotentests

---

- $Q_6$ : `/bib/book/author/text()`

- Ergebnis:

Serge Abiteboul

Victor Vianu

Jeffrey D. Ullman

- Rick Hull nicht, da das Element nur Element-Inhalt enthält

# XPath — Knotentests

---

- bereits gesehen: \* beliebiger Element- und @\* beliebige Attributknoten
- `text()` textueller Inhalt, Textknoten
- `node()` Element-, Text- oder Attributknoten (\*, @\* oder `text()`)
- weitere: `comment()`,  
`processing-instruction([name])`

# XPath — Attributknoten

---

- $Q_7$ : `/bib/book/@price`
- Ergebnis:  
55
- `@price`, `price` muß ein Attributknoten sein

# XPath — Selektionsprädikate

---

- $Q_8$ : `/bib/book/author[firstname]`

- Ergebnis:

```
<author><first-name>Rick</first-name>  
      <last-name>Hull</last-name>  
</author>
```

# XPath — weitere Prädikate

---

- $Q_9$ : `/bib/book/author[first-name]...`  
`...[address[//zip][city]]/last-name`

- Ergebnis:

`<last-name>...</last-name>`

# XPath — weitere Prädikate

---

- $Q_{10}$ : `/bib/book[@price < 60]`
- $Q_{11}$ : `/bib/book[author/@age < 25]`
- $Q_{12}$ : `/bib/book[author/text()]`

# Selektionsprädikate — Details

---

- wenn wahr, dann wird jeweiliger Knoten in Ausgabemenge übernommen,
- wenn Typ des Prädikates `number` ist, wird Knoten der entsprechenden Kontextposition übernommen,
- die Knotennumerierung beginnt mit 1
- `and`, `or` als logische Operatoren bei Wahrheitsausdrücken,
- Vergleichsoperatoren: `<`, `<=`, `!=`, `...`,
- *Achtung*: wenn Ausdrücke in XML-Dokumenten, Kodierung mit `&gt;` und `&lt;`,
- für Zahlen: `+`, `-`, `mod` und `div`,
- `' | '` zur Vereinigung von Knotenmengen,

# XPath — Zusammenfassung

---

<code>bib</code>	ein <code>bib</code> element
<code>*</code>	beliebiges Element
<code>/</code>	der Wurzelknoten
<code>/bib</code>	ein <code>bib</code> Element unterhalb der Wurzel
<code>bib/paper</code>	ein <code>paper</code> Element in <code>bib</code>
<code>bib//paper</code>	ein <code>paper</code> Element irgendwo unterhalb <code>bib</code>
<code>//paper</code>	ein <code>paper</code> Element irgendwo
<code>paper book</code>	ein <code>paper</code> oder <code>book</code> Element
<code>@price</code>	ein <code>price</code> Attribut
<code>bib/book/@price</code>	<code>price</code> Attribut von <code>book</code> in <code>bib</code>

# Quicky

---

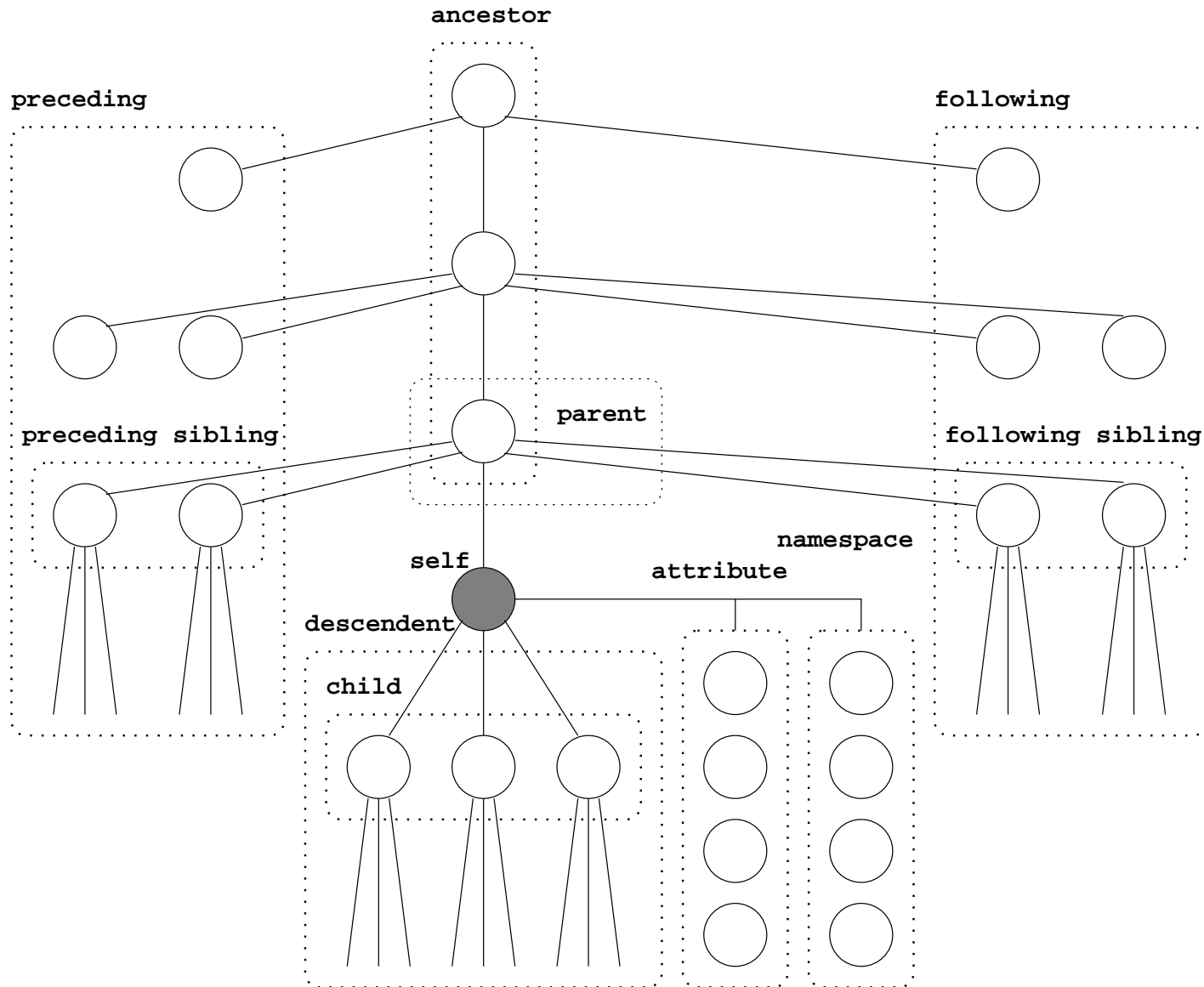
- Was bedeutet?  
`Q13: bib/book/[@price < 55]/author/last-name`

# XPath — weitere Details

---

- Es gibt insgesamt 13 Navigationsachsen:
  - self,
  - parent,
  - ancestor, ancestor-or-self,
  - child,
  - descendant, descendant-or-self,
  - following, following-sibling,
  - preceding, preceding-sibling,
  - attribute und
  - namespace.

# XPath — weitere Details



# XPath — weitere Details

---

- Äquivalenzen:
- `child::author/child:last-name`  $\mapsto$  `author/last-name`
- `child::author/descendant::zip`  $\mapsto$  `author//zip`
- `child::author/parent::*`  $\mapsto$  `author/..`
- `child::author/attribute::age`  $\mapsto$  `author/@age`
- Welche Bedeutung haben diese Pfadausdrücke?
  - `paper/publisher/parent::* /author`
  - `/bib//address[ancestor::book]`
  - `/bib//author/ancestor::* //zip`

# XPath — weitere Details

---

- `name()`, Name des aktuellen Elementes
- `/bib//*[name()='book']`  $\mapsto$  `/bib//book`
- Navigationsachsen sind machtvolles Instrument
- Quicky: Welche Bedeutung hat dieser Pfadausdruck?  
`/bib//*[ancestor::*[name()!='book']]`

# XPath — etwas formaler

---

- Ein XPath-Ausdruck  $p$  stellt eine Beziehung zwischen einem Kontextknoten und einem Knoten in der Ergebnismenge her.
- Formal:  $p$  definiert eine Funktion:  
 $S[p] : Nodes \rightarrow \{Nodes\}$
- Beispiele:
  - `author/first-name`
  - `.`  $\mapsto$  `self`
  - `..`  $\mapsto$  `parent`
  - `part/*/*[subpart]/../name`  $\mapsto$   
`part/*/*[subpart]/name`

# XPath — Äquivalenzen

---

mehr zu Äquivalenzen:

- „Olteanu et al: Xpath: Looking Forward. EDBT Workshop, 2002.“
- „Gottlob et al: Xpath Processing in a Nutshell. SIGMOD Record, Volume 32, Number 1, 2003.“

# XPath — Funktionsbibliothek

---

allgemeine und `string`, `boolean`, `number`-Funktionen,  
Auswahl:

`number last()` liefert die Position des letzten Elementes,

`number position()` gibt Kontextposition an,

`node-set id(object)` liefert den Knoten mit der  
referenzierten ID, Auflösung von IDREF,

`boolean contains(string, string)` wahr wenn  
zweites Argument Teil des ersten ist,

`boolean not(boolean)` Negation des angegebenen  
Wertes,

`number sum(node-set)` Summer der zu Zahlen  
gewandelten Argumentknoten,

... `number()`, `false()`, `boolean()`

---

# XPath — Achtung

---

- Prädikate sind nicht kommutative,  $a[b][2] \neq a[2][b]$
- es gibt Probleme mit vorwärts/rückwärts-Achsen und Prädikaten
- Expansion von `//` ist aufwendig
- numerische Prädikate
- Vergleiche mit Knotenmengen, Vergleich mit den `string`-Wert der Knotenmenge
- `string`-Wert einer Knotenmenge ist der `string`-Wert seines ersten Elementes!
- Navigationsachsen in XQuery eingeschränkt (`ancestor`, `descendant`)

# XPath — Ausblick

---

- Optimierung macht Push Down von Knotentests und Prädikaten notwendig
- Auswertung von XPath: XPath\*: PTIME, XPath//: PTIME,
- `/bib//*/author`  $\subseteq$  `/bib/*//author`?
- XPath\*,//: NP (Containment queries), für eingeschränkte Menge von Anfragen (lineare XPath\*,//  $\rightsquigarrow$   $\square\square$ ): praktische Algorithmen
- Unterstützung mit Vielzahl von Pfadindexstrukturen (siehe Kapitel Indizierung)
- XPath 2.0: Präfix (Variablen, Funktionen), Sequenzen, Werte-, (existenzquantifizierte) Mengen- und Identitätsvergleiche, weitere Operatoren

# XPath — Literatur

---

- Ressource: [www.w3.org/TR/XPath/](http://www.w3.org/TR/XPath/), die W3C Recommendation
- Buch: „Michael Kay: XSLT Programmer's Reference (2nd Edition). Wrox Press Ltd, 2002“
- Buch: „Meike Klettke, Holger Meyer: XML und Datenbanken. dpunkt.Verlag, 2002“, natürlich ;-), auch zu Unterschieden XPath 1.0 und XPath 2.0
- Ressource:  
<http://www.research.avayalabs.com/user/wadler>  
Phil Wadlers Homepage
- Artikel: „Gottlob et al: Xpath Processing in a Nutshell. SIGMOD Record, Volume 32, Number 1, 2003.“, guter Einstieg in XPath-Auswertung, auch Referenzen dort beachten