XLink

XML Linking: XLink, XPointer und XPath

30.11.2001 - Martin Klossek

Seminar „XML: Datenrepräsentation und Abfragesprachen “ - WS2001/2002

Motivation

Im World Wide Web sind unzählbar viele Dateien zu unterschiedlichsten Themenbereichen verfügbar...

Motivation

...wodurch Menschen Informationen aller Art finden und ihren Wissensschatz verbreitern können.

Mit ein – wenn nicht sogar **der** – Hauptgrund für die Erfolgsstory des WWW:

Hyperlinks

„Klicken Sie hier für weitere Infos“

Motivation

HTML-**Hyperlinks**

Verbinden Resourcen wie HTML-Dateien, Dokumente, Texte, Bilder, Grafik, Multimedia...

Fest in den HTML-Standard eingebettet
(und nur in diesem Kontext einsetzbar)

```
<a href="resource-uri">click here</a>

<img src="image-uri">
```

Motivation

HTML-**Hyperlinks**

- Zielen auf Interaktivität ab.
  - Bekannte Konzept: „Browsen“ oder „Surfen“
- Für Maschinen schwierig zu verarbeiten, insbesondere bei Änderungen
  - da feste Einbindung in HTML-Dokumente, die verglichen mit XML schwer zu modifizieren sind

Motivation

XLink tritt an für

- Idee eines „**eXtensible Linkkonzepts**“ (wie bei XML auch!)
- Flexible Einsatzmöglichkeiten im **Resourcenmanagment**
- Überwindung der Schwächen von HTML-Hyperlinks

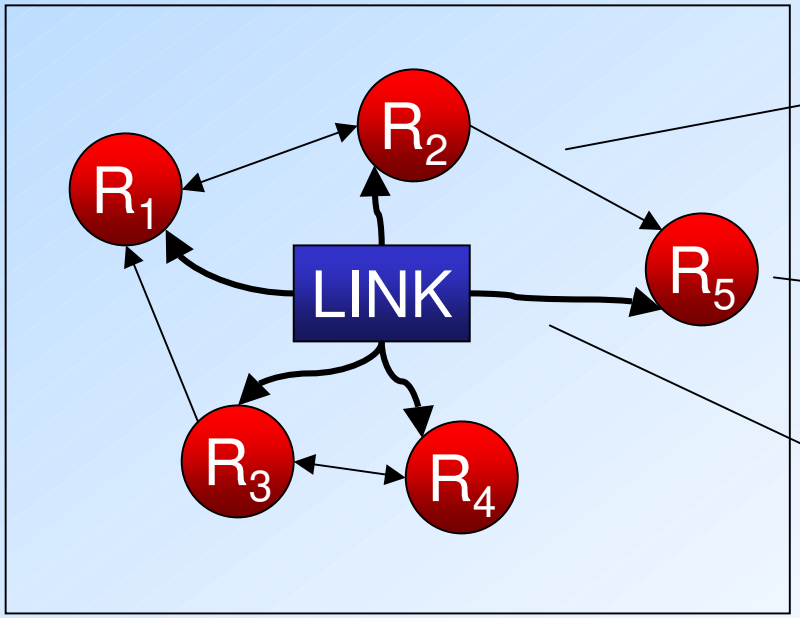
Schauen wir uns das doch näher an!

Übersicht

1. Motivation
2. XLink zur Verlinkung
3. XPath zum Adressieren
4. XPointer zum Zeigen
5. Ausblick

XLink - Idee

Idee ist die Resourcenverknüpfung

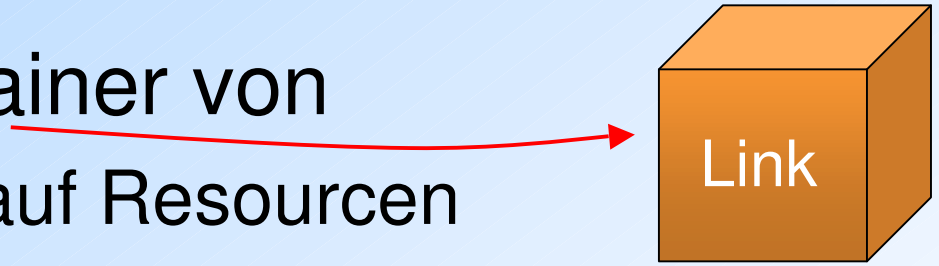


- XML-Daten
- HTML-Daten
- Grafiken
- Dokumente
- ...

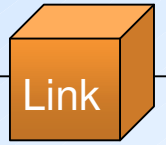
XLink - Idee

Link als Container von

- Referenzen auf Resourcen
- Lokalen Resourcen
- Verbindungen zwischen diesen Resourcen
- Metainformationen (wie Titel des Links oder Verhaltensanweisungen)



XLink - Idee



Damit bietet ein XLink-Link:

- uni- und bidirektionale Verknüpfungs-strukturen
- mehr als 2 Resourcen pro Link möglich
- Metadaten können angegeben werden
- Trennung von Resource und Link durch externes Speichern von Links
  - achja, und für XML-Resourcen geht es auch mit Subresourcen

XLink - Spezifikation

XLink ist spezifiziert vom W3C (als Recommendation = Empfehlung)

offiziell seit dem 27. Juni 2001

verfügbar unter dem Titel

„XML Linking Language (XLink) Version 1.0“

unter der URL

http://www.w3.org/TR/2001/REC-xlink-20010627/

XLink - Syntax

1

Im Folgenden das **Wesentliche** aus der XLink-Spezifikation, so zunächst:

XLink kennt zwei syntaktische Konstrukte für Links

- Extended Links
- Simple Links (als Vereinfachung von Extended Links ähnlich wie <a href>)

in XML-Syntax (natürlich J)

XLink - Syntax

2

Um eXtensible gerecht zu werden:

XLink definiert keine XML-Elemente, sondern globale Attribute in einem Namespace http://www.w3.org/1999/xlink

Beispiel

```
<link xlink:type="extended"
 xmlns:xlink="http://www.w3.org/1999/xlink">
    ...
</link>
```

XLink - Syntax

3

Beliebige XML-Elemente können durch XLink-Attribute zu XLinks werden!

Zentrales Attribut ist **`type`**

`type` gibt an, von welchem XLink-Typ das Element sein soll...

- je nach Wert sind weitere Attribute nötig, um XLink-konform zu sein

XLink - Syntax

Typen und
mögliche Attribute

type="resource"
role, title, label

type="locator"
href, role, title, label

type="extended"
role, title

type="title"
Titel im Element

type="simple"
href, role, arcrole,
title, show, actuate

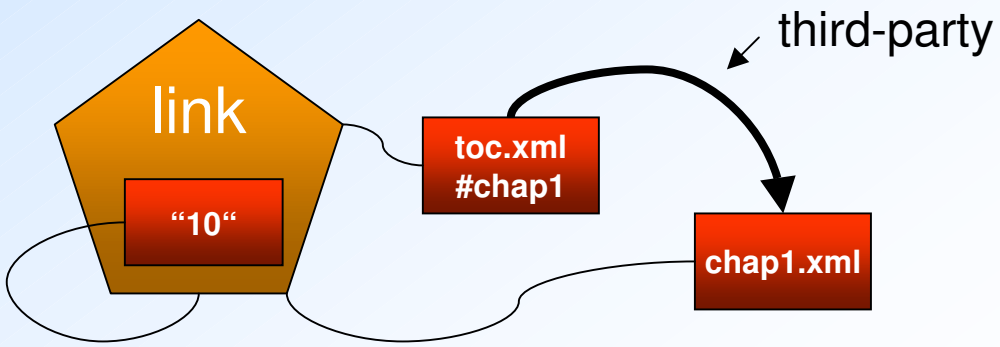
type="arc"
arcrole, title, show,
actuate, from, to

XLink - Syntax

Beispiel

3

```
<link xlink:type="extended" xlink:role="connector">
   <title xlink:type="title">Von TOC zum Chapter 1</title>
   <res xlink:type="resource">10</res>
   <loc xlink:type="locator" xlink:href="toc.xml#chap1"
    xlink:label="a"/>
   <loc xlink:type="locator" xlink:href="chap1.xml"
    xlink:label="b"/>
   <arc xlink:type="arc" xlink:from="a" xlink:to="b"/>
</link>
```



XLink - Syntax

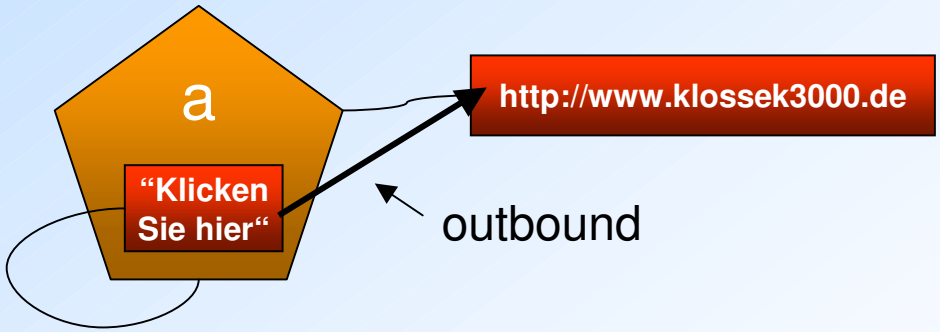
- Extended Links unterstützen **alle** Features von XLink
- Zur Vereinfachung: Simple Links
  - Subset von Extended Links
  - einfach anzuwenden
  - gehen von einer lokalen Resource zu einer externen Resource
  - ähnlich den HTML-Hyperlinks!

XLink - Syntax

Beispiel

4

```
<a xmlns:xlink="http://www.w3.org/1999/xlink"
 xlink:type="simple"
 xlink:href="http://www.klossek3000.de"
 xlink:actuate="onRequest"
 xlink:show="replace"
 xlink:title="Link zu Martins Homepage">Klicken Sie hier</a>
```



entsprechend auch inbound, z. B. für Einbettungen wie bei <img>

XLink - Linkbases

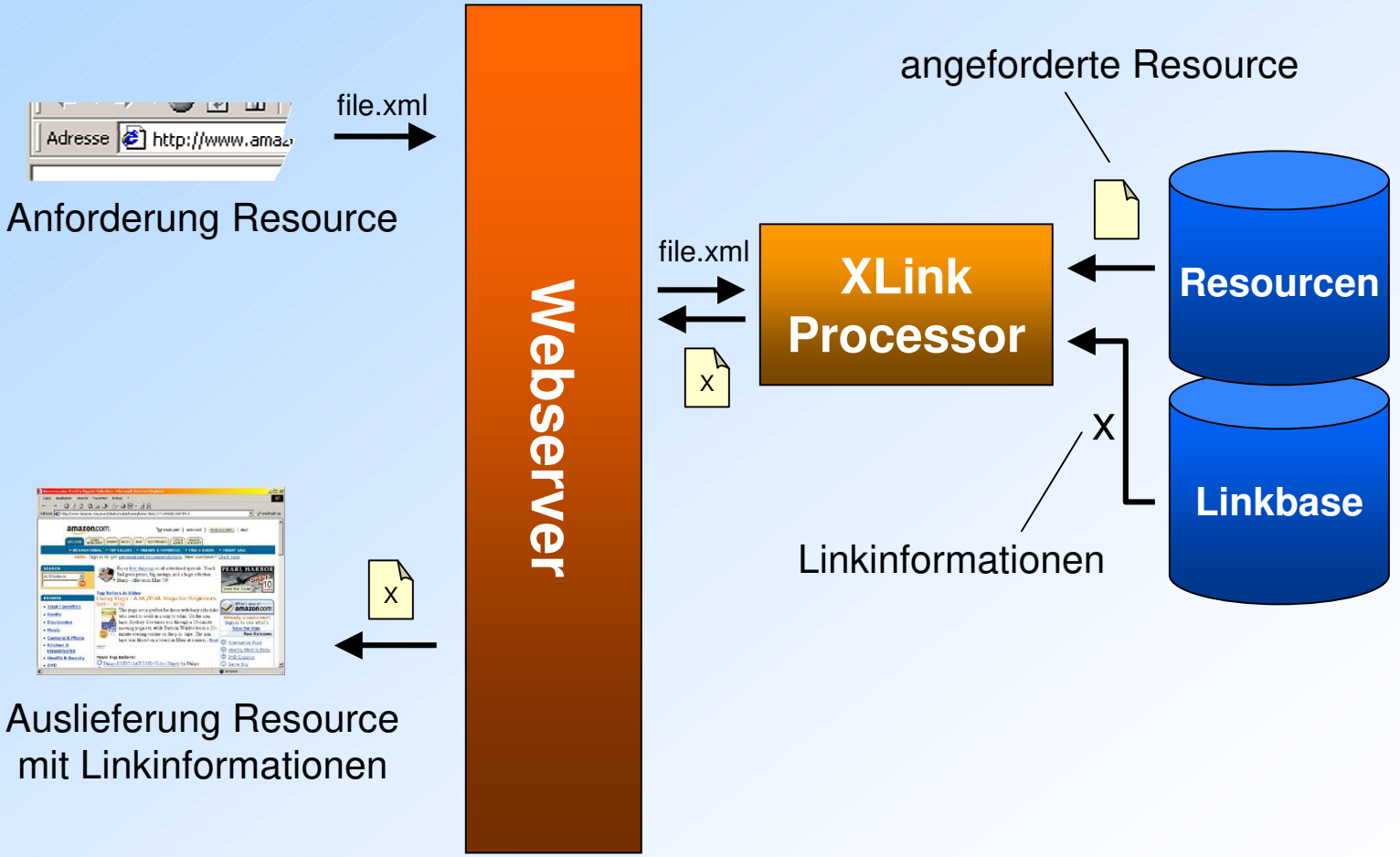
5

Weiteres Feature von XLink

Externe Links und Linkdatenbanken

- Erlauben leichte Änderung von Links
- Essentiell bei nicht änderbaren Resourcen wie Grafiken
- Nach Spezifikation im XLink-Format

Anwendungsbeispiel



Übersicht

1. Motivation
2. XLink zur Verlinkung
3. XPath zum Adressieren
4. XPointer zum Zeigen
5. Ausblick

XPath - Idee

XPath ist eine Sprache zur **Adressierung von Teilresourcen** von XML-konformen Daten

kein XML-Dialekt, um in XML-Attributen und URIs genutzt zu werden

sehr mächtig, um flexible Anwendungen zu ermöglichen

wird von **XSLT** + **XPointer** verwendet

XPath - Spezifikation

XPath ist spezifiziert vom W3C (als Recommendation = Empfehlung)

offiziell seit dem 16. November 2001

verfügbar unter dem Titel

„XML Path Language (XPath) Version 1.0“

unter der URL

http://www.w3.org/TR/xpath

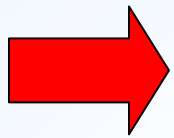
XPath - Anforderungen

- Zugriff auf **Teilresourcen** eines XML-konformen Dokuments ist das Ziel
- Berücksichtigung der **Hierarchie** und Semistrukturiertheit der XML-Daten
- Sprache soll Variablen, Rechen-operationen und Stringverarbeitung kennen für Flexibilität
- **einfache** nicht-XML Notation

XPath - Konzept

XPath operiert auf einer **Baumrepräsentation** der XML-Daten

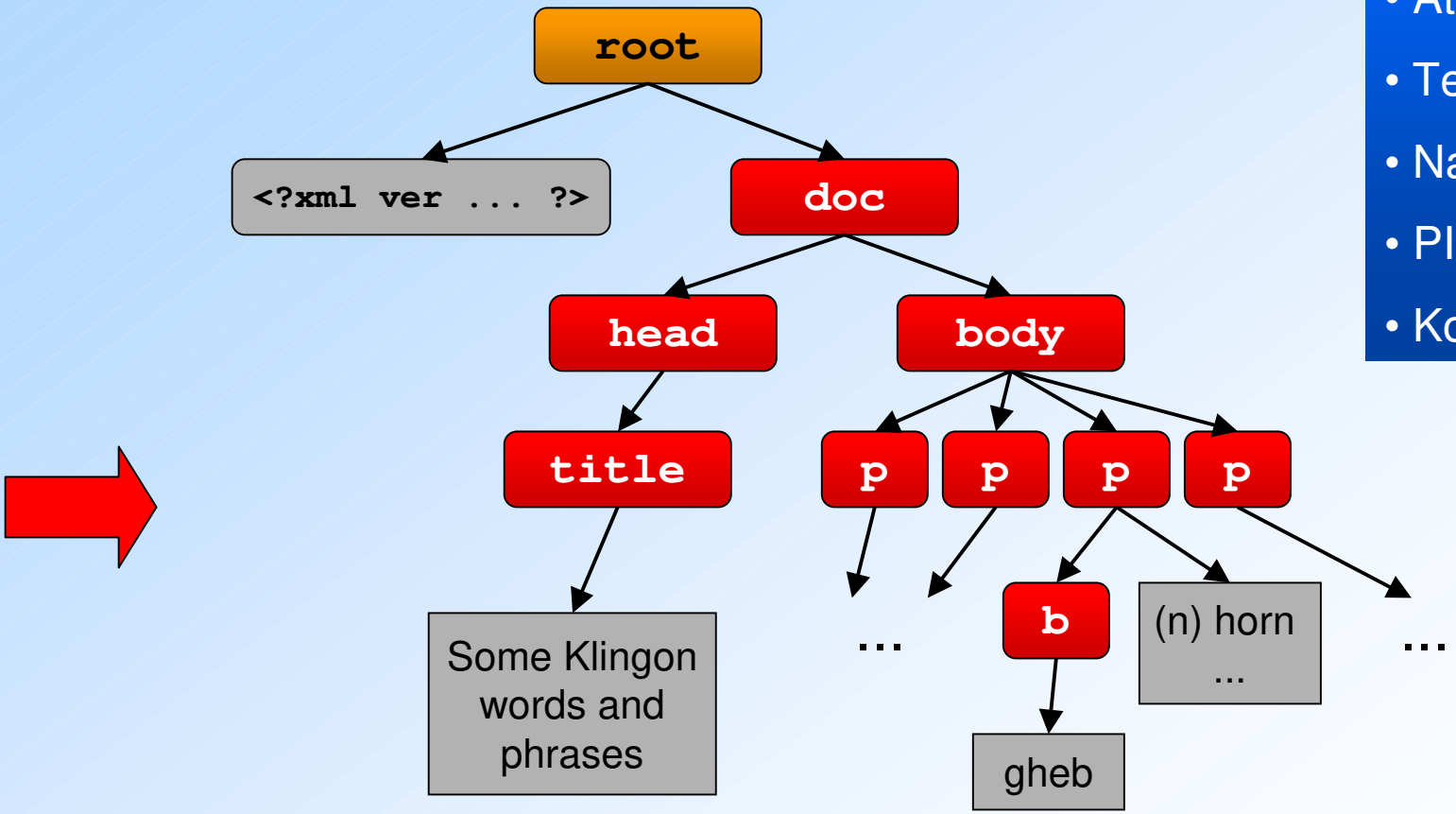
```
<doc>
 <head><title>Some Klingon words and phrases</title></head>
 <body>
   <p><b>amarklor-kalish</b>(n) is a Klingon word used to
    describe a color on the ultra-violet scale seen by
    Klingons. It falls between amarklor (very dark violet
    or purple) and kalish (completely black).</p>
   <p><b>cheb</b>(n) unit weight approx. 5 pounds (2.25kg)</p>
   <p><b>gheb</b>(n) horn (musical instrument)</p>
   <p><b>nuqDaq 'oH puchpa''e'</b>'Where is thebathroom?'</p>
 </body>
</doc>
```



XPath - Konzept

Knotentypen

- Wurzelknoten
- Elemente
- Attribute
- Texte
- Namespace
- PIs
- Kommentare



XPath - Konzept

Beispiel

Ein beispielhafter XPath-Ausdruck auf diesen Daten

```
/doc/body/p[2]
```

würde die Subresource

```
<p><b>cheb</b>(n) unit weight
approx. 5 pounds (2.25kg)</p>
```

liefern (den 2. Absatz)

XPath - Syntax

Zentrales Element in XPath ist der Location Path

ein Pfad durch den Baum!

Syntax

z.B. /doc/body/p

```
LocationPath ::= [/]Step(/Step)*

Step ::= Axis `::` NodeTest Predicate*
```

XPath - Syntax

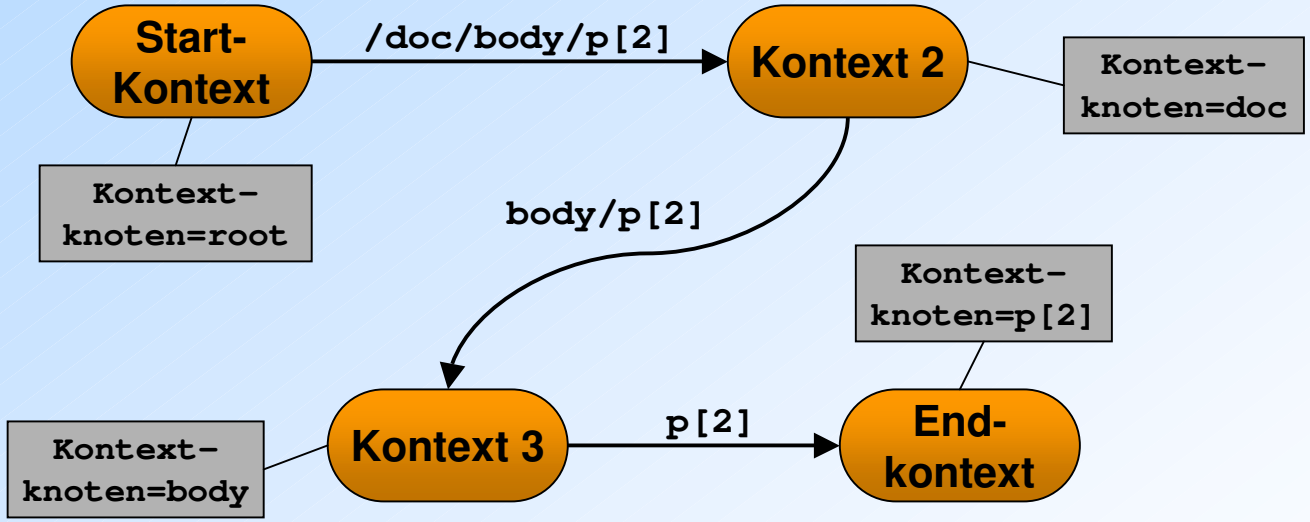
Abarbeitung eines Location Path erfolgt von links nach rechts

Zu jedem Zeitpunkt mit **Kontext**

- Kontextknoten (node)
- Kontextposition (Position Kontextknoten)
- Kontextgröße (max. Pos. Kontextknoten)
- Menge von Funktionen
- Variablen mit Werten

XPath - Syntax

Am Beispiel `/doc/body/p[2]`



XPath - Syntax

`Axis`: verschiedene Achsen für Baumdurchlauf (Kinder, Geschwister)

in XPath mögliche Achsen sind

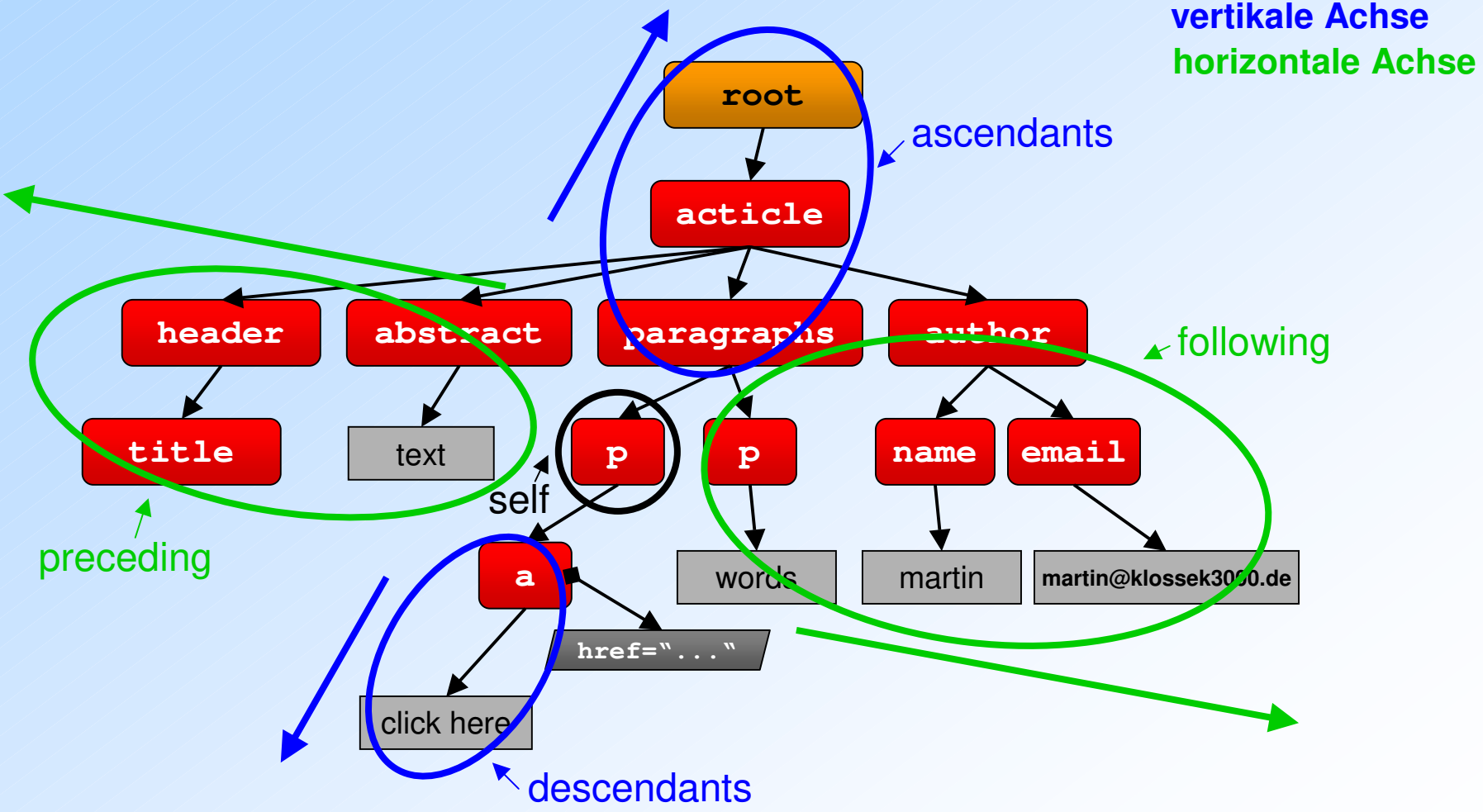
child, descendant, parent, ancestor

following-sibling, preceding-sibling, following, preceding

attribute, namespace

self, descendant-or-self, ancestor-or-self

XPath - Syntax



XPath - Syntax

Beispiel

Axis: Beispiele

- **descendant::a[self::text()='click here']**
- **child::a[attribute::href='...']**
- **/descendant::p**
- **/child::article/child::author/child::name**

XPath - Syntax

`NodeTest`: die Knoten in der Achse werden mit NodeTest gefiltert

node()	* DT
typ DT	namespace[:typ] DT
text()	comment()
processing-instructin[name]	DT = Defaulttyp!

XPath - Syntax

Beispiel

`NodeTest`: Beispiele

- `child::text()`
- `/descendant::loc/attribute::xlink:label`
- `//loc/@xlink:label`

XPath - Syntax

`Predicate*`: Filterung der Knotenmenge mit Prädikaten

Syntax

```
Predicate ::= '[' Expr ']'
```

Erfüllt ein Knoten der Knotenmenge das Prädikat, wird er in die Ergebnismenge aufgenommen, andernfalls nicht

XPath - Syntax

XPath kennt für Ausdrücke (expr) die Datentypen

- node-set (Knotenmenge, auch leer)
- boolean
- number (Fließkommazahl)
- string (Zeichenkette)

Vergleichen, Verknüpfen, Verrechnen von Ausdrücken möglich

XPath - Syntax

Darüberhinaus verfügt XPath über eine umfangreiche **Funktions-bibliothek**, beispielsweise

string **concat** (string, string, string*)

number **last** ()

number **floor** (number)

...

XPath - Syntax

Beispiel

Predicate: Beispiele

- `/child::module/child::item[5]`
- `/child::module/child::item[$i]`
- `/child::module/child::item[last()]`
-
```
(string(./treelink) and
 string(/communitysys/session/loginname))
 or string(./newlink) or
 string(./doshowinvisible)
```

Übersicht

1. Motivation
2. XLink zur Verlinkung
3. XPath zum Adressieren
4. XPointer zum Zeigen
5. Ausblick

XPointer - Idee

XPointer dient zum Adressieren von **Fragmenten in XML-Daten**

Das klingt ja fast so wie XPath!?

- richtig, aber nur fast, denn XPointer geht weiter!

Genauso wie XPath nur für XML-Dokumente anwendbar

XPointer - Spezifikation

XPointer ist spezifiziert vom W3C (Candidate Recommendation)

offiziell seit dem 11. September 2001

verfügbar unter dem Titel

“XML Pointer Language (XPointer) Version 1.0“

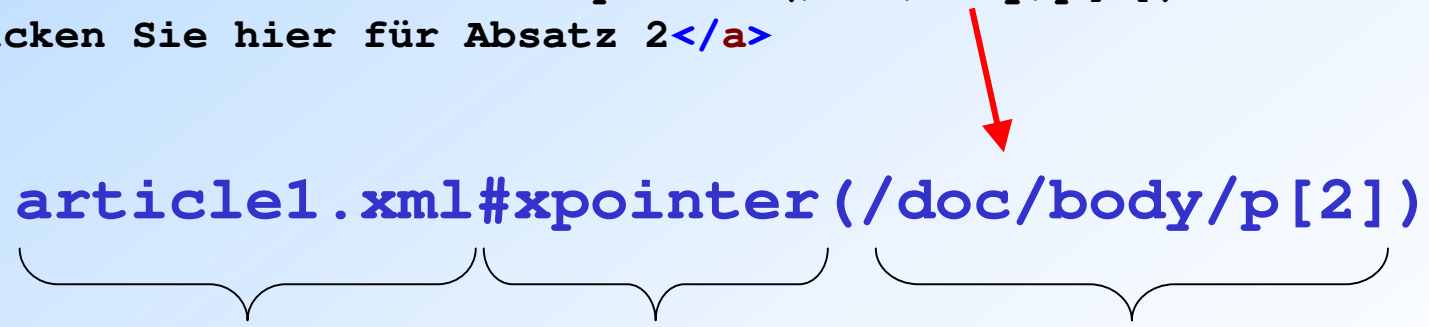
unter der URL

http://www.w3.org/TR/xptr/

XPointer - Syntax

XPointer sehen **Anchors** in HTML ähnlich:

```
<a xmlns:xlink="http://www.w3.org/1999/xlink"
 xlink:type="simple"
 xlink:href="article1.xml#xpointer(/doc/body/p[2])"
>Klicken Sie hier für Absatz 2</a>
```



XPointer - Syntax

XPath kann Knoten des XML-Dokuments adressieren

XPointer kennt zusätzlich

- Punkte (`point`)
  - gibt die Möglichkeit, Knoten **und** auch einzelne Zeichen anzusprechen
- Bereiche (`range`)
  - alle XML Informationen zwischen zwei Punkten (z. B. Zeichenkette oder Elemente)

XPointer - Syntax

`point` und `range` verlangen nach einer Generalisierung der XPath Node und Node-Set-Datentypen:

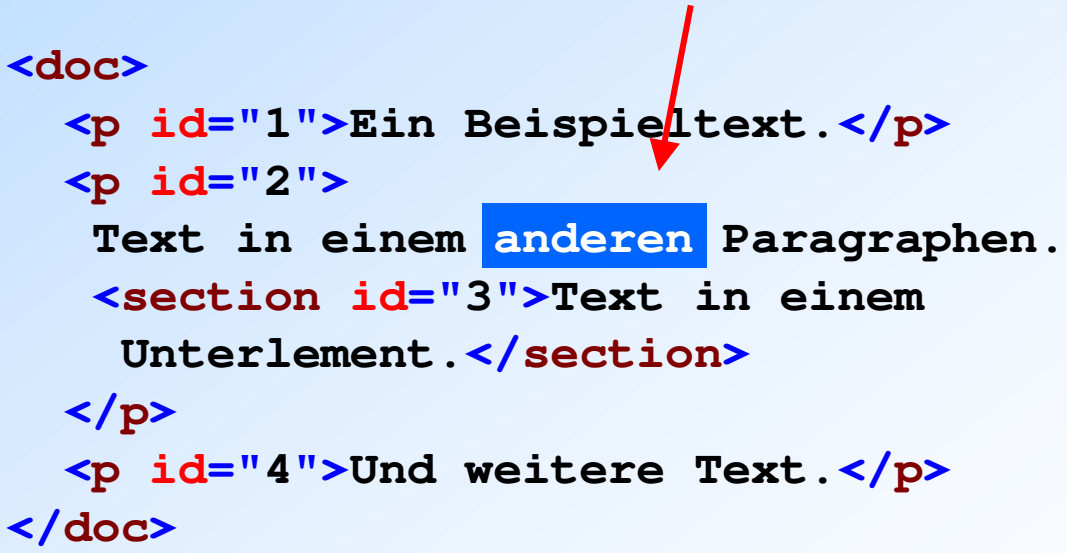
- location = node | point | range
- location-set = geordnete Liste von Daten des Typs location
- singleton = location-set der Größe 1

XPointer - Syntax

Beispiel

Entsprechend auch eine Reihe neuer Funktionen für **`range`** und **`point`**

xpointer(range(string-range(//*,'anderen')))



XPointer - Syntax

Die neuen Funktionen sind

location-set **range-to** (location-set)

location-set **string-range** (location-set, string, number?, number?)

location-set **range** (location-set)

location-set **range-inside** (location-set)

location-set **start-point** (location-set)

location-set **end-point** (location-set)

location-set **here** () + location-set **origin** ()

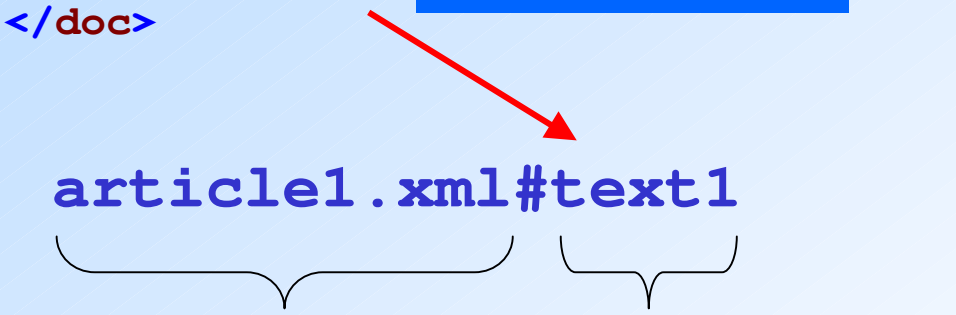
XPointer - Syntax

Beispiel

Es geht aber auch einfacher!

mit **bare names**

```
<doc>
  <p id="text1">Ein Beispieltext.</p>
</doc>
```



Dokumentname

bare name = id des referenzierten Elements

- Damit greift man - direkt wie bei HTML-Anchors - auf Elemente mit der **ID=“bare name“** zu!

XPointer - Syntax

Beispiel

Kindersequenzen (child sequences)

```
<doc>
   <p id="2">
    Text in einem anderen Paragraphen.
    <section id="3">Text in einem
     Unterlement.</section>
   </p>
</doc>
```

```
article1.xml/1/2/3
```

- Damit greift man direkt auf die Kinderelemente zu.

- Jeder Schritt schreitet eine Ebene tiefer und sucht das i-te Kinde dort

Dokumentname

childsequence = durch / separierte Index der Kinderelemente

XPointer

FAZIT:

- XPointer bieten eine Reihe von Vereinfachungen und Erweiterungen gegenüber XPath, um Fragmente von XML-Resourcen zu adressieren
- Prinzipiell bedienen sie sich aber der Mächtigkeit von XPath
- Für XLink mit der bare name Adressierung essentiell

Übersicht

1. Motivation
2. XLink zur Verlinkung
3. XPath zum Adressieren
4. XPointer zum Zeigen
5. Ausblick

Ausblick

W3C XML Pointer, XML Base and XML Linking

http://www.w3.org/XML/Linking

XLink: Recommendation seit 27.06.2001

XPath: Recommendation seit 16.11.1999, 2.0 in Arbeit

XPointer: Candidate Recommendation seit 11.09.2001

Ausblick

- Implementierungen von XLink durch
  - teilweise durch Mozilla

  - empolis x2x - Linkmanagment

  - einige experimentelle Software
- XPath dagegen wesentlich häufiger
  - XSLT-Prozessoren (alle!)
  - in XML-Datenbanken wie tamino von Software-AG

Ausblick

Beobachtungen und Vermutungen

- XLink ist **Abrundung** von XML um Verknüpfungsfähigkeiten
- Linkinformationen könnten auch in **Datenbanken** beschrieben werden!
- Primärer Einsatz von Links sind im Moment **Hyperlinks** in **HTML**. Ob sie je durch XLink abgelöst werden, ist fraglich...

Fazit

XLink ist die logische Ergänzung zu XML, um Resourcen verknüpfen zu können und dabei wie bei XML selbst flexibel und erweiterbar zu sein.

Kontakt

Kontakt:

martin@klossek3000.de

Folien und Ausarbeitung in verschiedenen Formaten unter

http://www.stormzone.de/uni/Hauptstudium/seminare/xml/list.php3

Das war‘s