

# **XLink**

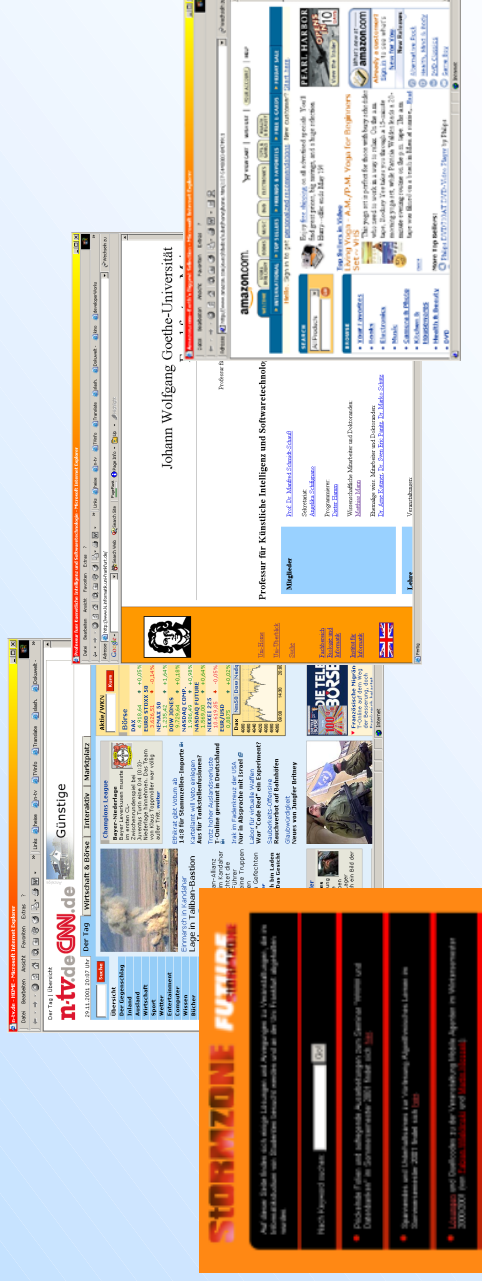
**XML Linking: XLink, XPointer und XPath**

30.11.2001 - Martin Klosek

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

# Motivation

Ø ImWorldWideWebsindunzählbar  
vieleDateienzuunterschiedlichsten  
Themenbereichenverfügbar...



# Motivation

- ∅ ...wodurchMenschenInformationenaller ArtfindenundihrenWissensschatz verbreiternkönnen.
- ∅ Mitein – wennnichtsogar **der** – HauptgrundfürdieErfolgsstorydesWWW:

Hyperlinks

„**KlickenSie hier fürweitererInfos**“



# Motivation

## Hyperlinks



- Ø HTML-Hyperlinks
- Ø Verbinden Ressourcen wie HTML - Dateien, Dokumente, Texte, Bilder, Grafik, Multimedia...
- Ø Festen HTML - Standard eingebettet (und nur in diesem Kontext einsetzbar)

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

```

```

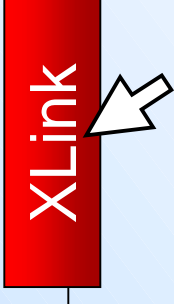
# Motivation

Hyperlinks



- Ø **HTML-Hyperlinks**
- Ø Zielen auf Interaktivität ab.
- Ø Bekanntes Konzept: „Browsen“ oder „Surfen“
- Ø Für Maschinenschwierig zu verarbeiten, insbesondere bei Änderungen
- Ø das feste Einbinden in HTML - Dokumente, die verglichen mit XML schwer zu modifizieren sind

# Motivation



- Ø **XLink** trittan für
- Ø Idee eines „**eXtensibleLink**konzepts“  
(wie bei XML auch!)
- Ø Flexible Einsatzmöglichkeiten
- Ø **Ressourcenmanagement**
- Ø Überwindung der Schwächen von HTML -  
Hyperlinks
- Ø Schauen wir uns das doch näher an!

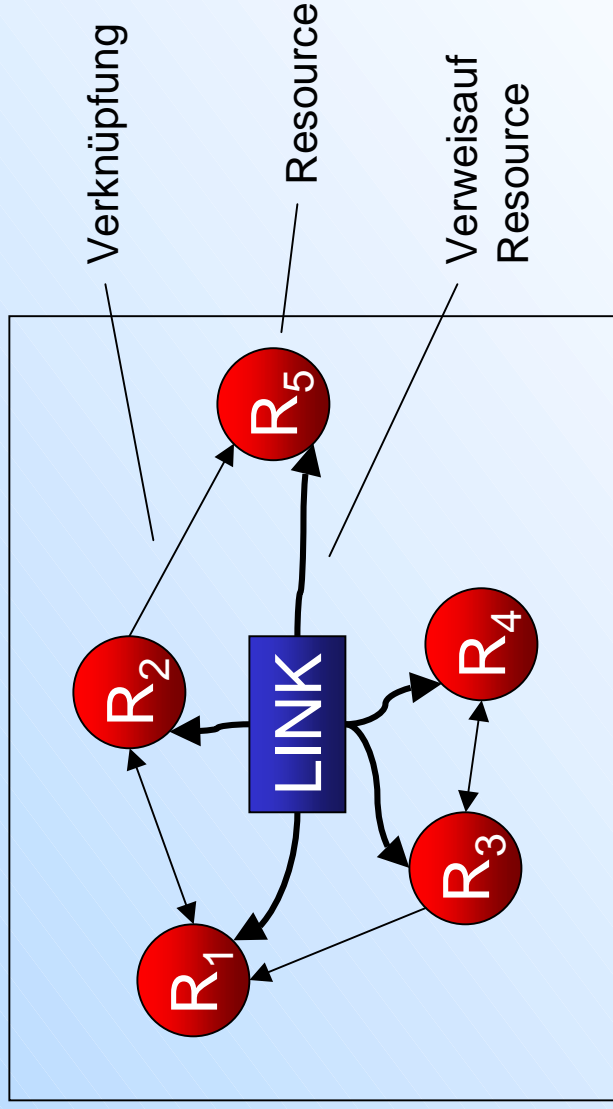
# Übersicht

1. Motivation
2. XLinkzurVerlinkung
3. XPathzumAdressieren
4. XPointerzumZeigen
5. Ausblick

# XLink - Idee

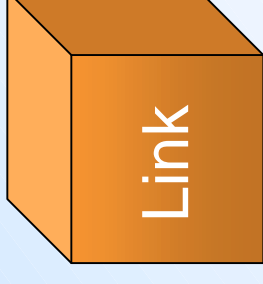
Ø Idee ist die Ressourcenverknüpfung

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

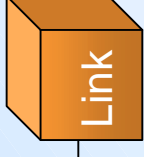


# XLink - Idee

- Ø LinkalsContainervon
- Ø ReferenzenaufRessourcen
- Ø LokalenRessourcen
- Ø Verbindungenzwischen diesen Ressourcen
- Ø Metainformationen(wieTiteldesLinks oderVerhaltensanweisungen)



# XLink - Idee



- Ø Damit bietet ein XLink -Link:
  - Ø uni- und bidirektionale Verknüpfungsstrukturen
  - Ø mehr als 2 Ressourcen pro Link möglich
  - Ø Metadaten können angegeben werden
  - Ø Trennung von Resource und Link durch externes Speichern von Links
  - Ø ach ja, und für XML - Ressourcengehtes auch mit Subressourcen

# XLink - Spezifikation

- Ø XLink ist spezifiziert vom W3C (als Recommendation = Empfehlung)
- Ø offiziell seit dem 27. Juni 2001
- Ø verfügbar unter dem Titel

„XML Linking Language (XLink) Version 1.0“

- Ø unter der URL

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

# XLink - Syntax

1

- Ø Im Folgenden das **Wesentliche** aus der XLink - Spezifikation, so zunächst:
- Ø XLink kennt zwei syntaktische Konstrukte für Links
  - Ø **Extended Links**
  - Ø **Simple Links** (als Vereinfachung von Extended Links ähnlich wie `<ahref>`)
- Ø in XML - Syntax (natürlich J )

# XLink - Syntax

2

- Ø UmeXtensiblerechtzuwerden:
- Ø XLinkdefiniertkeineXML -Elemente, sondernglobaleAttributeineinem Namespace <http://www.w3.org/1999/xlink>

## Beispiel

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

# XLink - Syntax

3

- Ø BeliebigeXML -Elementekönnen durchXLink -AttributezuXLinks werden!
- Ø ZentralesAttributist **type**
- Ø **type** gibtan, vonwelchemXLink -Typ dasElementseinsoll...
- Ø jenachWertsindweitereAttributenötig, umXLink -konformzusein

# XLink - Syntax

3

∅ Typen und  
mögliche Attribute

**type="resource"**  
role, title, label

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

**type="title"**  
TitleElement

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

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

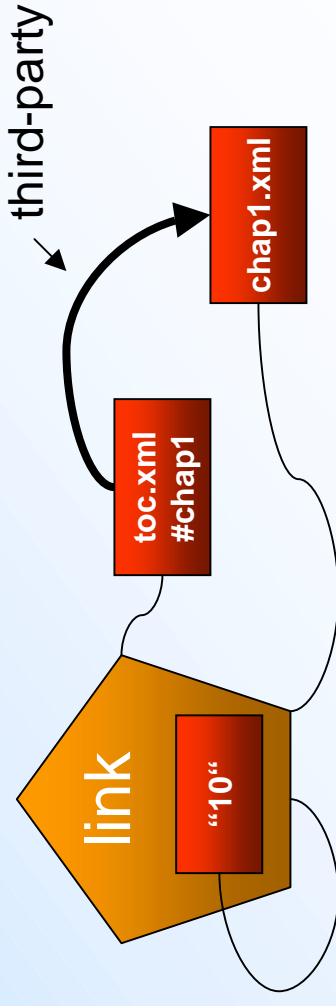
**type="extended"**  
role, title

# 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

4

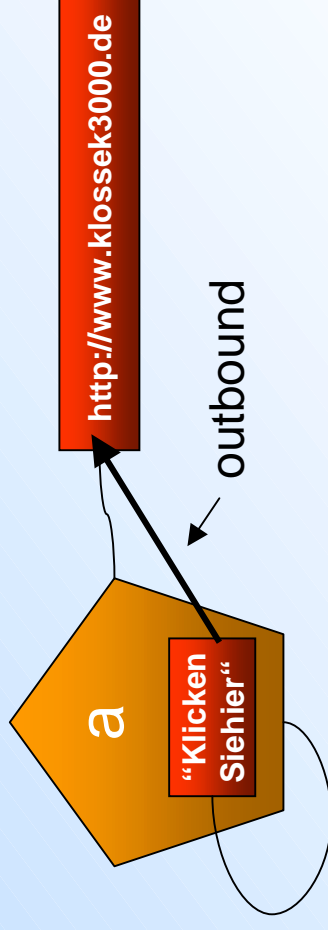
- ∅ ExtendedLinks unterstützen **alle** Features von XLink
- ∅ Zur Vereinfachung: **SimpleLinks**
  - ∅ Subset von ExtendedLinks
  - ∅ einfach anzuwenden
  - ∅ gehen von einer lokalen Resource zu einer externen Resource
  - ∅ ähnlichen HTML -Hyperlinks!

# 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>
```



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

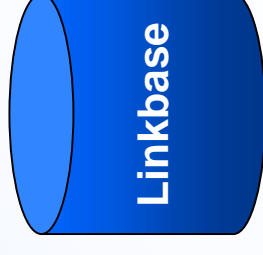
# XLink - Linkbases

5

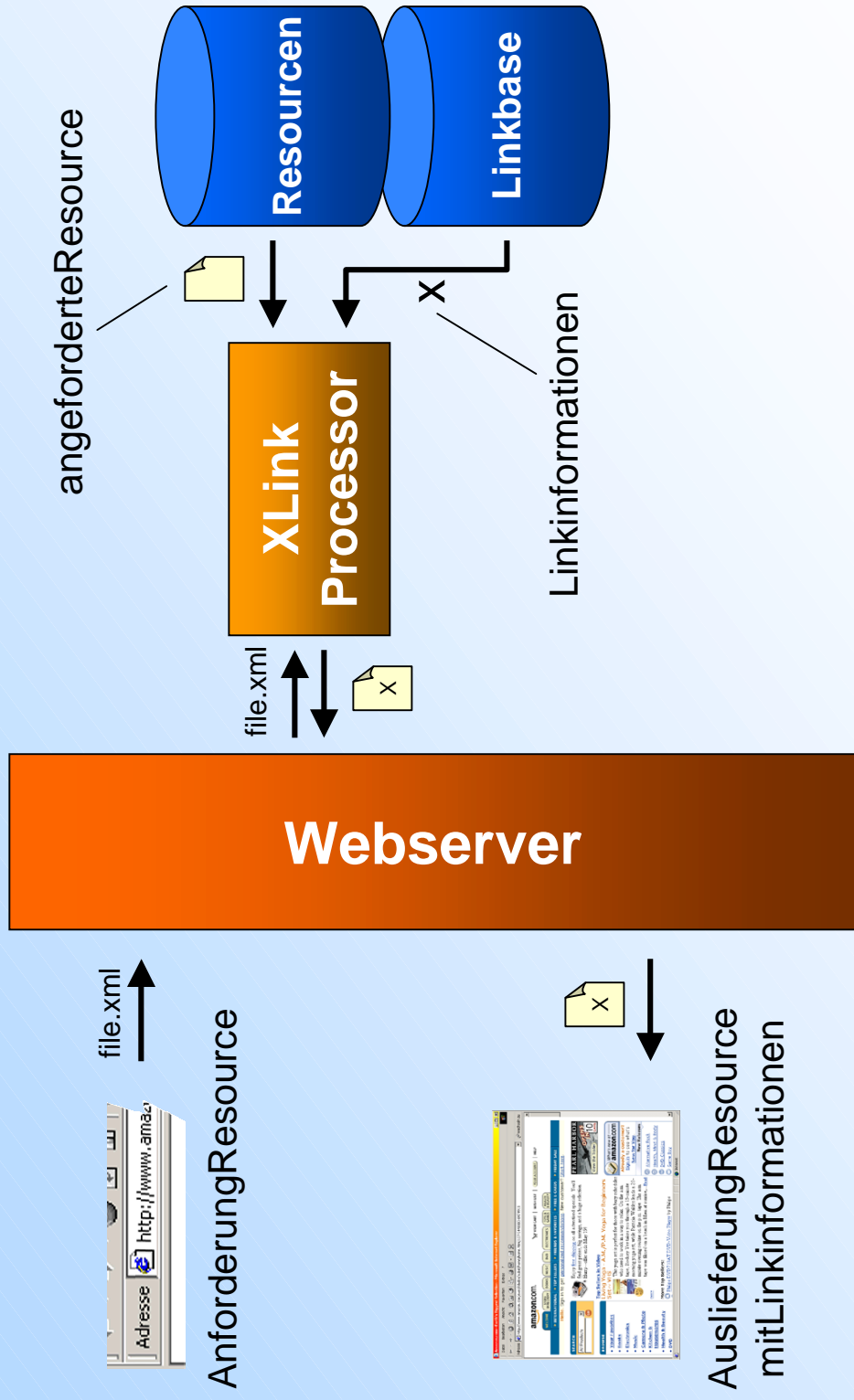
Ø Weiteres Feature von XLink

## Externe Links und Linkdatenbanken

- Ø Erlauben leichte Änderungen von Links
- Ø Essentiell bei nicht änderbaren Ressourcen wie Grafiken
- Ø Nachspezifikation im XLink -Format



# Anwendungsbeispiel



# Übersicht

---

1. Motivation
2. XLinkzurVerlinkung
3. XPathzumAdressieren
4. XPointerzumZeigen
5. Ausblick

# XPath - Idee

- Ø XPath ist eine Sprache zur **Adressierung von Teilressourcen** von XML-konformen Daten
- Ø **kein XML-Dialekt**, um in XML - Attributen und URIs genutzt zu werden
- Ø **sehr mächtig, um flexible Anwendungen zu ermöglichen**
- Ø **wird von XSLT + XPath verwendet**

# XPath - Spezifikation

- Ø XPath ist spezifiziert vom W3C (als Recommendation = Empfehlung)
- Ø offiziell seit dem 16. November 2001
- Ø verfügbar unter dem Titel

„XMLPathLanguage(XPath)Version1.0“

- Ø unter der URL

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

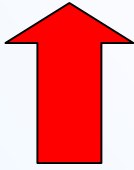
# XPath - Anforderungen

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

# XPath - Konzept

## Ø XPatheriertafeiner Baumrepräsentation derXML -Daten

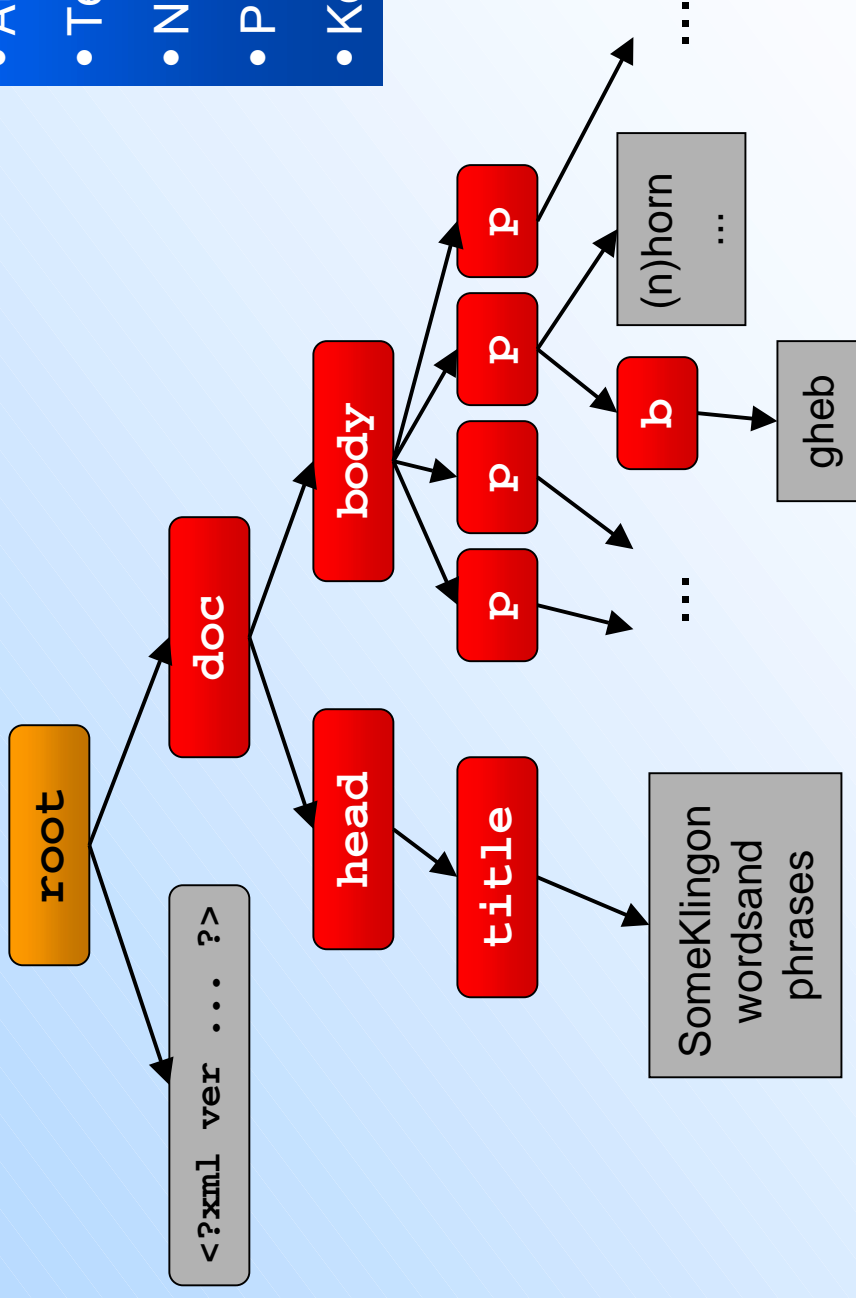
```
<doc>  
<head><title>Some Klingon words and phrases</title></head>  
<body>  
  <p><b>amarklor-kalish</b>(n) is a Klingon word used to  
  describe a color on the ultra-violet scale seen by  
  Klingons. It falls between amarklor (very dark violet  
  or purple) and kalish (completely black).</p>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



# XPath - Konzept

## Knotentypen

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



# XPath - Konzept

Beispiel

Ø EinbeispielhafterXPath -Ausdruckauf diesenDaten

`/doc/body/p[2]`

würdedieSubresource

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

liefern(den2.Absatz)

# XPath - Syntax

Ø Zentrales Element in XPath ist der  
LocationPath

Ø ein Pfad durch den Baum!

Ø Syntax

z.B. /doc/body/p

**LocationPath ::= [ / ] step ( / step ) \***

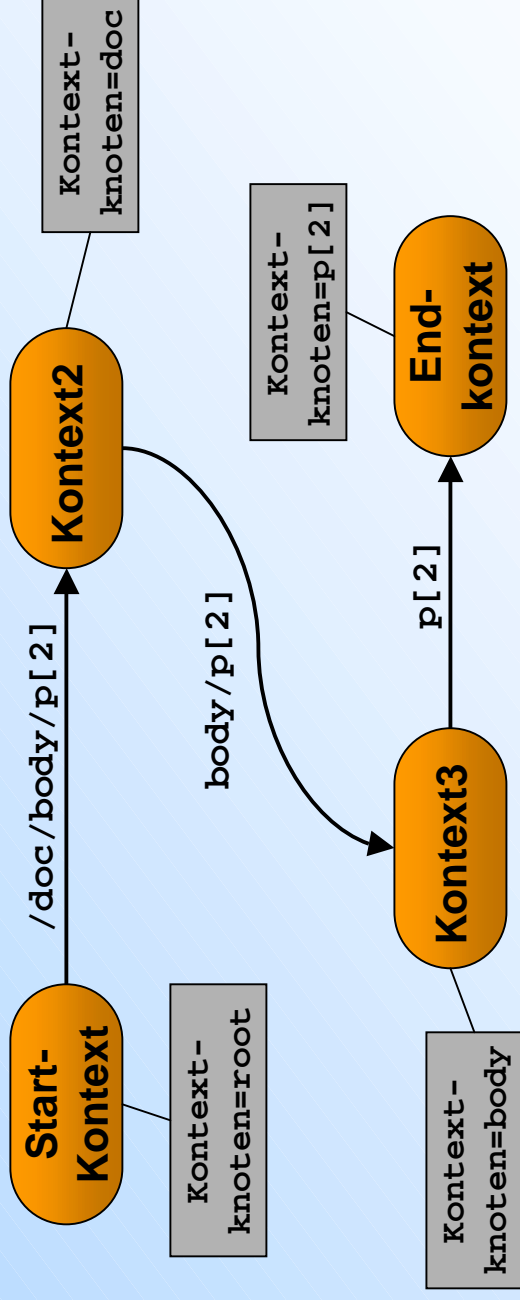
**→ Step ::= Axis ':: ' NodeTest Predicate \***

# XPath - Syntax

- Ø Abarbeitung eines **LocationPath** erfolgt von links nach rechts
- Ø Zu jedem Zeitpunkt mit **Kontext**
  - Ø Kontextknoten(node)
  - Ø Kontextposition(PositionKontextknoten)
  - Ø Kontextgröße(max.Pos.Kontextknoten)
  - Ø Menge von Funktionen
  - Ø Variablen mit Werten

# XPath - Syntax

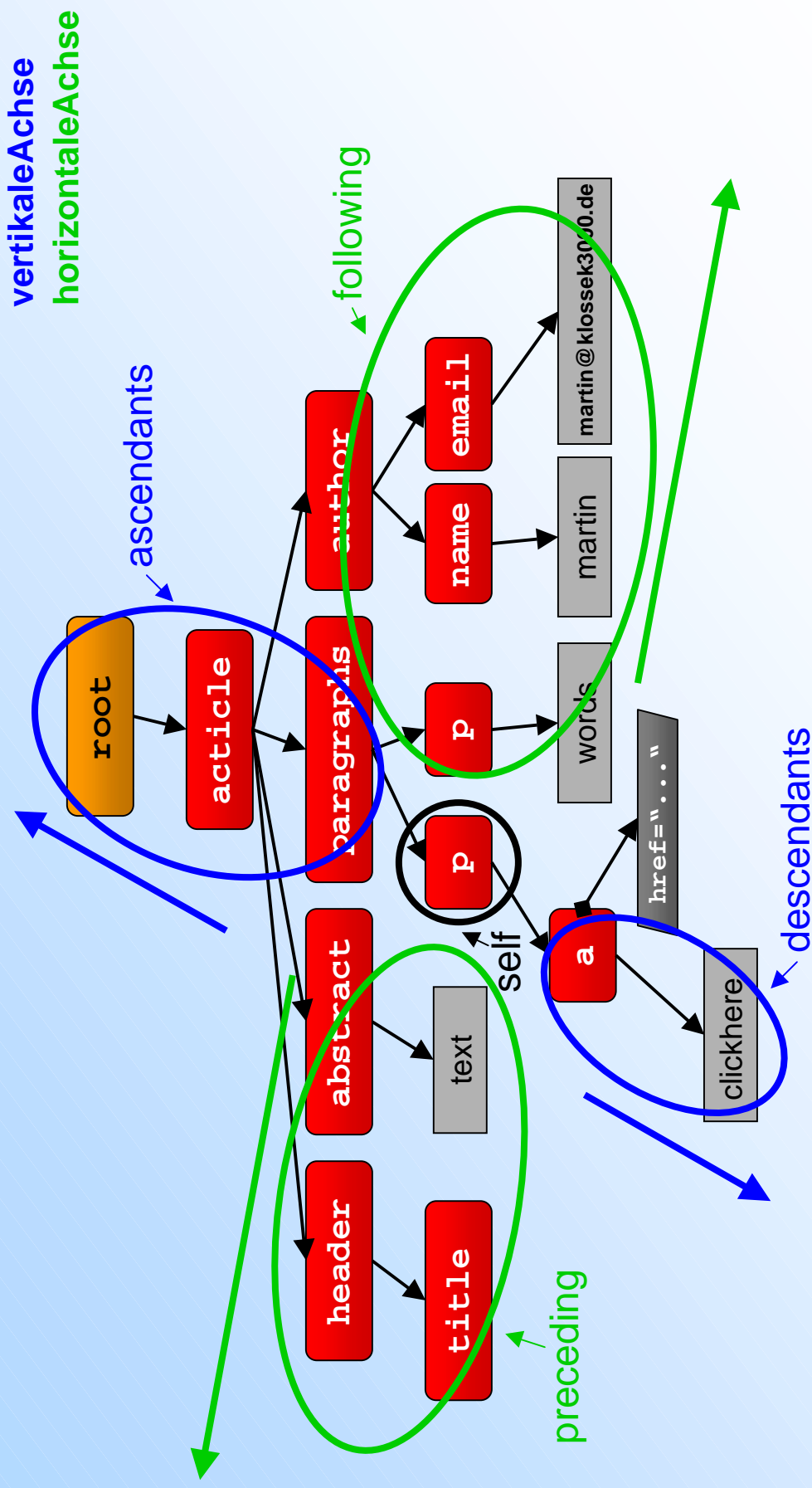
Ø AmBeispiel `/doc/body/p[2]`



# XPath - Syntax

- Ø **Axis**: verschiedene Achsen für Baumdurchlauf (Kinder, Geschwister)
- Ø in XPath mögliche Achsen sind
  - Ø child, descendant, parent, ancestor
  - Ø following-sibling, preceding-sibling, following, preceding
  - Ø attribute, namespace
  - Ø self, descendant-or-self, ancestor-or-self

# XPath - Syntax



# XPath - Syntax

Beispiel

## Ø Axis: Beispiele

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

# XPath - Syntax

Ø **NodeTest**: die Knoten in der Achse werden mit `NodeType` gefiltert

<code>node()</code>	*	DT
<code>typ</code>	DT	DT
<code>text()</code>		<code>comment()</code>
<code>processing-instruction[name]</code>		<b>DT=Defaulttyp!</b>

# XPath - Syntax

Beispiel

## ∅ NodeTest:Beispiele

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

# XPath - Syntax

Ø **Predicate\***: Filterung der Knotenmenge mit Prädikaten

Ø Syntax

**Predicate ::= '[' Expr ']'**

Ø Erfüllte Knoten der Knotenmenge das Prädikat, wird in die Ergebnismenge aufgenommen, andernfalls nicht

# XPath - Syntax

Ø XPathkenntfürAusdrücke( **expr**)die

## Datentypen

Ø node-set( Knotenmenge, auchleer )

Ø boolean

Ø number(Fließkommazahl)

Ø string( Zeichenkette)

Ø **Vergleichen, Verknüpfen, Verrechnen  
vonAusdrückenmöglich**

# XPath - Syntax

Ø Darüberhinausverfügt XPath über  
eine umfangreiche **Funktions-**  
**bibliothek**, beispielsweise

Ø **string concat** (string, string, string\* )

Ø **number last** ()

Ø **number floor** (number)

Ø ...

# 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. XLinkzurVerlinkung
3. XPathzumAdressieren
4. XPointerzumZeigen
5. Ausblick

# XPointer - Idee

- Ø XPointer dient zum Adressieren von **Fragmenten in XML - Daten**
- Ø Dasklingt ja fast so wie XPath!?
  - Ø richtig, aber nur fast, denn XPointer geht weiter!
- Ø Genauso wie XPath nur für XML - Dokumenten anwendbar

# XPointer - Spezifikation

- Ø XPointer ist spezifiziert vom W3C (Candidate Recommendation)
- Ø offiziell seit dem 11. September 2001
- Ø verfügbar unter dem Titel

“XMLPointerLanguage(XPointer)Version1.0 “

- Ø unter der URL

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

# XPointer - Syntax

## Ø XPointer sehen Anchors inHTML 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>
```

`article1.xml#xpointer(/doc/body/p[2])`

Dokumentname    “#xpointer“    XPointer-Ausdruck

# XPointer - Syntax

- Ø XPath kann Knotendes XML - Dokument adressieren
- Ø XPointer kennt zusätzlich
  - Ø Punkte (**point**)
    - gibt die Möglichkeit, Knoten und auch einzelne Zeichen anzusprechen
  - Ø Bereiche (**range**)
    - alle XML Informationen zwischen zwei Punkten (z.B. Zeichenkette oder Elemente)

# XPointer - Syntax

- ∅ **point** und **range** verlangenacheiner GeneralisierungderXPathNodeund Node-Set-Datentypen:
  - ∅ location=node|point|range
  - ∅ location-set=geordneteListevonDaten desTypslocation
  - ∅ singleton=location -setderGröße1


# XPointer - Syntax

Beispiel

- Ø Entsprechend auch eine Reihe neuer Funktionen für `range` und `point`

```
xpointer(range(string-range(//*[ 'anderen' ])))
```

```
<doc>  
  <p id="1">Ein Beispieltext.</p>  
  <p id="2">  
    Text in einem anderen Paragraphen.  
    <section id="3">Text in einem  
      Unterlement.</section>  
  </p>  
  <p id="4">Und weitere Text.</p>  
</doc>
```



# XPointer - Syntax

## Ø DieneuenFunktionsind

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 **barenames**

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

article1.xml#text1

Dokumentname    barename=iddes  
referenziertenElements

- Damit greift man -  
direkt wie bei HTML -  
Anchors - auf  
Elemente mit der  
**ID="barename"** zu!

# XPointer - Syntax

Beispiel

## ∅ Kindersequenzen( childsequences )

```
<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 Kind dort

**Dokumentname**    **childsequence=durch/**  
**separierterIndex der Kinderelemente**

# XPointer

- Ø **FAZIT:**
- Ø XPointer bietet eine Reihe von Vereinfachungen und Erweiterungen gegenüber XPath, um Fragmenten von XML-Resourcen zu adressieren
- Ø Prinzipiell bedient sich aber der Mächtigkeit von XPath
- Ø Für XLink mit der **barename** Adressierung essentiell

# Übersicht

1. Motivation
2. XLinkzurVerlinkung
3. XPathzumAdressieren
4. XPointerzumZeigen
5. Ausblick

# Ausblick

## Ø W3CXMLPointer, XMLBase and XMLLinking



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

- Ø XLink: Recommendation seit 27.06.2001
- Ø XPath: Recommendation seit 16.11.1999, 2.0 in Arbeit
- Ø XPointer: Candidate Recommendation seit 11.09.2001

# Ausblick

Ø Implementierungen von XLink durch

Ø teilweise durch Mozilla



Ø empolisx2x - Linkmanagement



Ø einige experimentelle Software

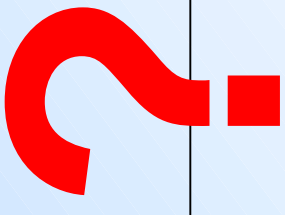
Ø XPath dagegen wesentlich häufiger

Ø XSLT-Prozessoren (alle !)

Ø 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 Linkssindim  
Moment **Hyperlinks** in **HTML**. Obsieje durch XLink abgelöst werden, ist fraglich...

# Fazit

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

# Kontakt

- Ø Kontakt:
  - Ø [martin@klossek3000.de](mailto:martin@klossek3000.de)
  - Ø Folien und Ausarbeitung in verschiedenen Formaten unter <http://www.stormzone.de/uni/Hauptstudium/seminare/xml/list.php3>



# Daswar's

