

XML Linking Language (XLink) - Inhalt

- I. Wozu XLink? Konzepte und Design
- II. Beispiele
- III. Beschränkungen von XLink
- IV. Verbreitung von XLink

XML Linking Language (XLink)

- W3C Recommendation 27. Juni 2001
Version 1.0 (neueste Version)
- das Design von XLink wurde vor allem von *HTML*, *HyTime* und *Text Encoding Initiative Guidelines* beeinflusst
- XLink definiert XML-Attribute, mit denen man sowohl einfache gerichtete Links (ähnlich wie in HTML) als auch komplexe Links beschreiben kann.

Wozu XLink?

- XLink definiert Attribute, mit denen anderen XML-Dokumenten(typen) Linking-Funktionalität hinzugefügt werden kann
- XLink definiert **keinen** Dokumententyp
- XLink ist sowohl für Hyperlinks im WWW als auch für Links gedacht, die von Computern verarbeitet werden sollen

Vergleich mit HTML (a bzw. img)

- in HTML wird der Link an einem seiner beiden Anker definiert
- der Benutzer kann den Link nur vom Quellanker zum Zielanker traversieren
- in XLink haben Links ein viel allgemeineres Konzept
- XLink stellt ein Elemententyp `simple` zur Verfügung, der eine ähnliche Funktionalität wie `a` hat (und ähnlich einfache Syntax)

Vorteile von XLink

- XLink erlaubt es, mehr als zwei Quellen mit **einem** Link zu verknüpfen
- XLink macht das Assoziieren von Metadaten mit einem Link möglich
- in XLink können Links ausgedrückt werden, deren Quellen (sowohl Start als auch Ziel) sich an anderer Stelle als der Link befinden

XLink Konzepte

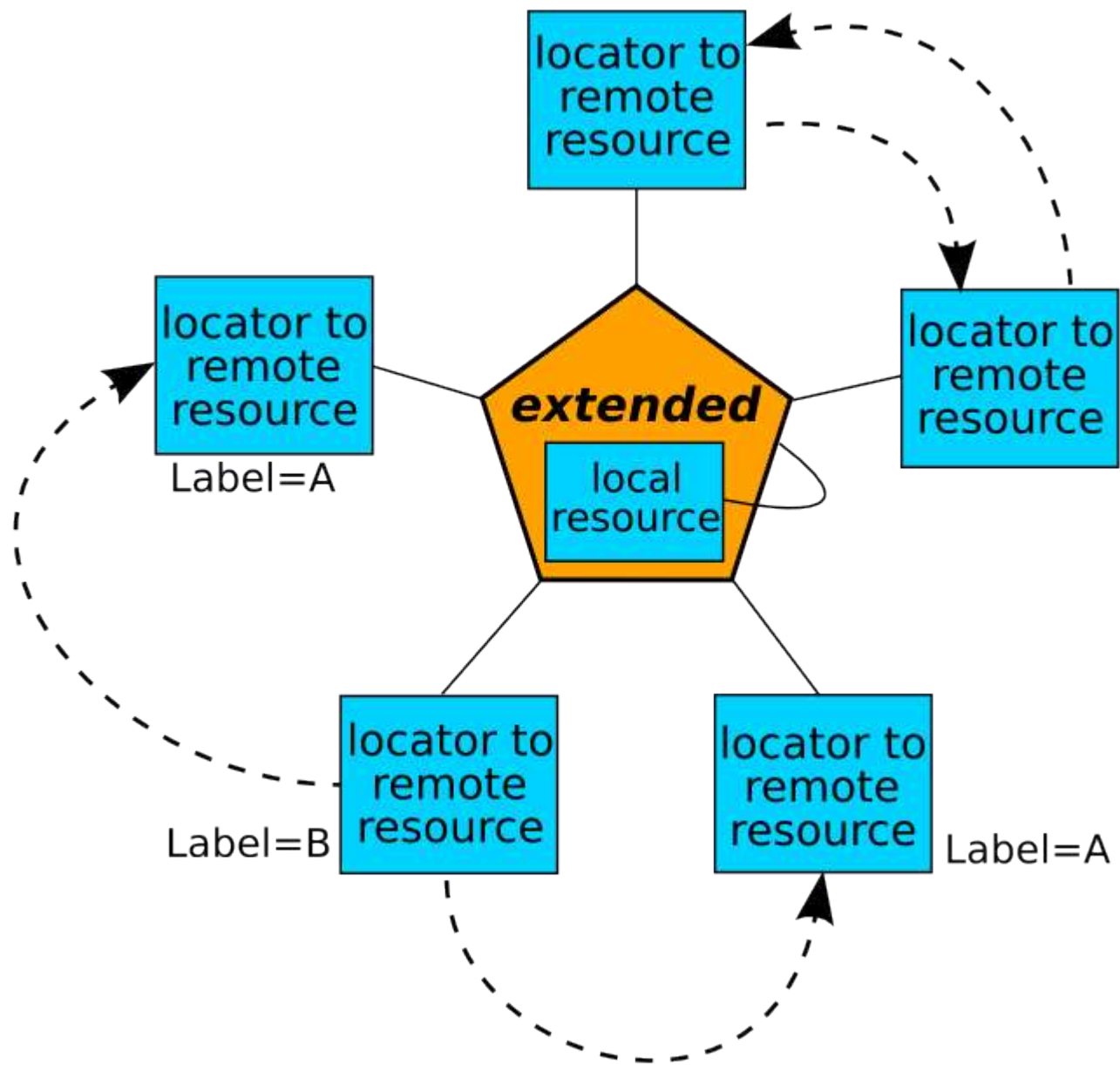
- in XLink können beliebig viele Quellen zu einem Link verbunden werden
- in das *Traversieren* eines Links sind aber immer genau 2 Quellen (oder Teile davon) involviert.
- Informationen darüber, wie ein Paar von Quellen traversiert werden kann, werden in einem *arc* gespeichert.

XLink Konzepte

- Eine *lokale Quelle* eines Links ist ein spezielles Kind-Element des Links.
- Eine *remote Quelle* ist eine Quelle, die durch eine URI angegeben wird.
- Ein arc mit lokaler Quelle als Startpunkt und remote Quelle als Endpunkt heißt *auswärts gehend*
- analog: *einwärts gehender arc*

XLink Konzepte

- Sind beide Ressourcen remote, heißt ein arc *third-party arc*
- XML-Dokumente, die Sammlungen von einwärts gehenden und third-party Links enthalten heißen Linkdatenbanken.

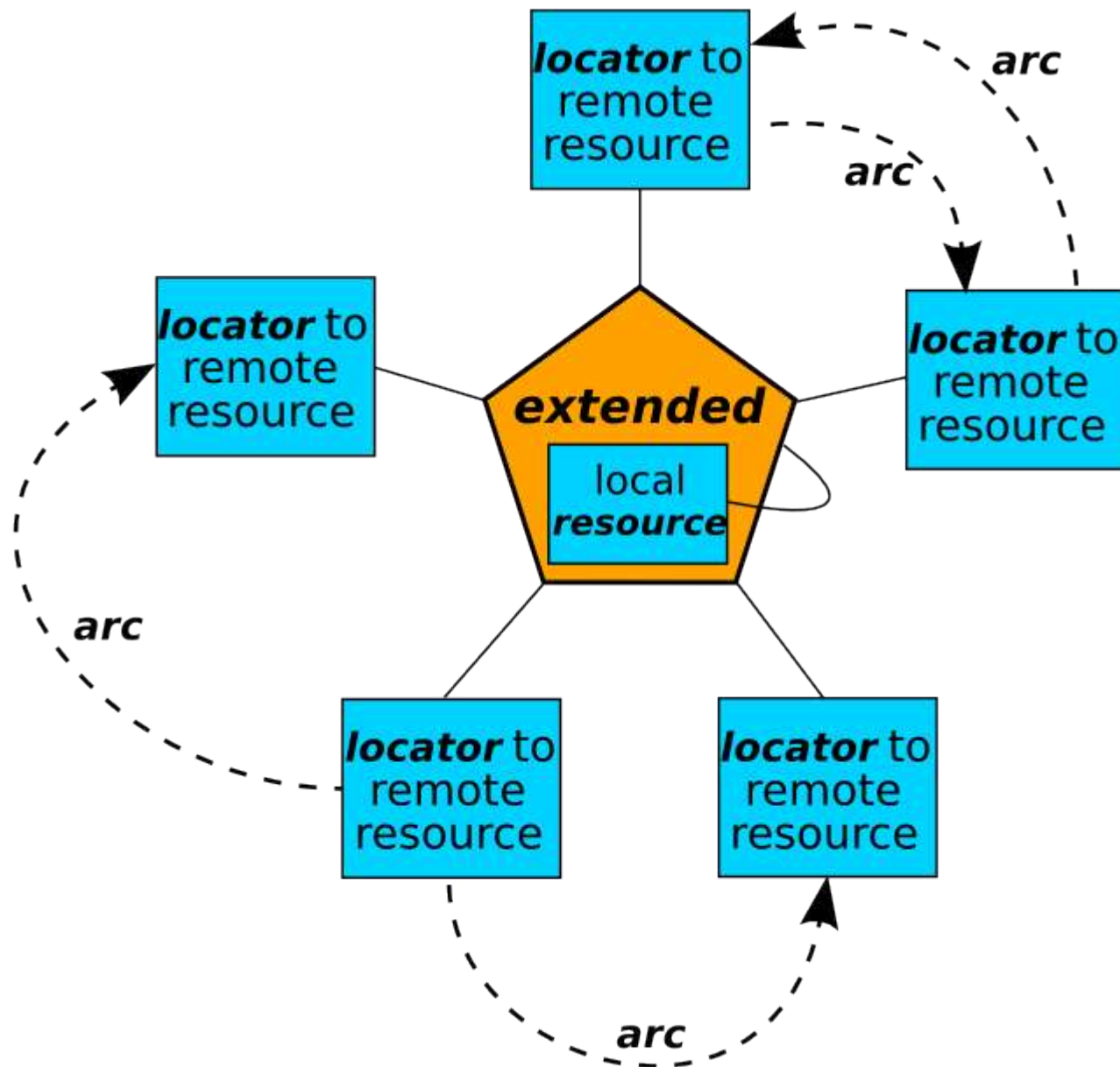


XLink Design

- XLink Namespace:
<http://www.w3.org/1999/xlink>
- XLink definiert lediglich Attribute und Attributwerte. Dadurch können XLink-Attribute problemlos mit Elementen aus anderen Namespaces verwendet werden
- Diese **globalen Attribute** sind: `type`, `href`, `role`, `arcrole`, `title`, `show`, `actuate`, `label`, `from` and `to`

XLink Design

- Ein XML-Element genügt XLink, wenn
 - es ein **type** Attribut aus dem XLink Namespace hat, das einen der sieben Werte “**simple**”, “**extended**”, “**locator**”, “**arc**”, “**resource**”, “**title**” oder “**none**” hat
 - es konform mit den Beschränkungen des gewählten Typs ist (später mehr)



extended und simple

- zwei verschiedene Arten von Links: **extended** Links und **simple** Links
- Extended Links bieten die volle Funktionalität von XLink, wie beliebig viele Quellen, einwärts gehende und third-party arcs.
- Simple Links bieten eine abkürzende Syntax für auswärts gehende Links mit genau 2 Ressourcen. Sie haben keine spezielle interne Struktur.

Attribute

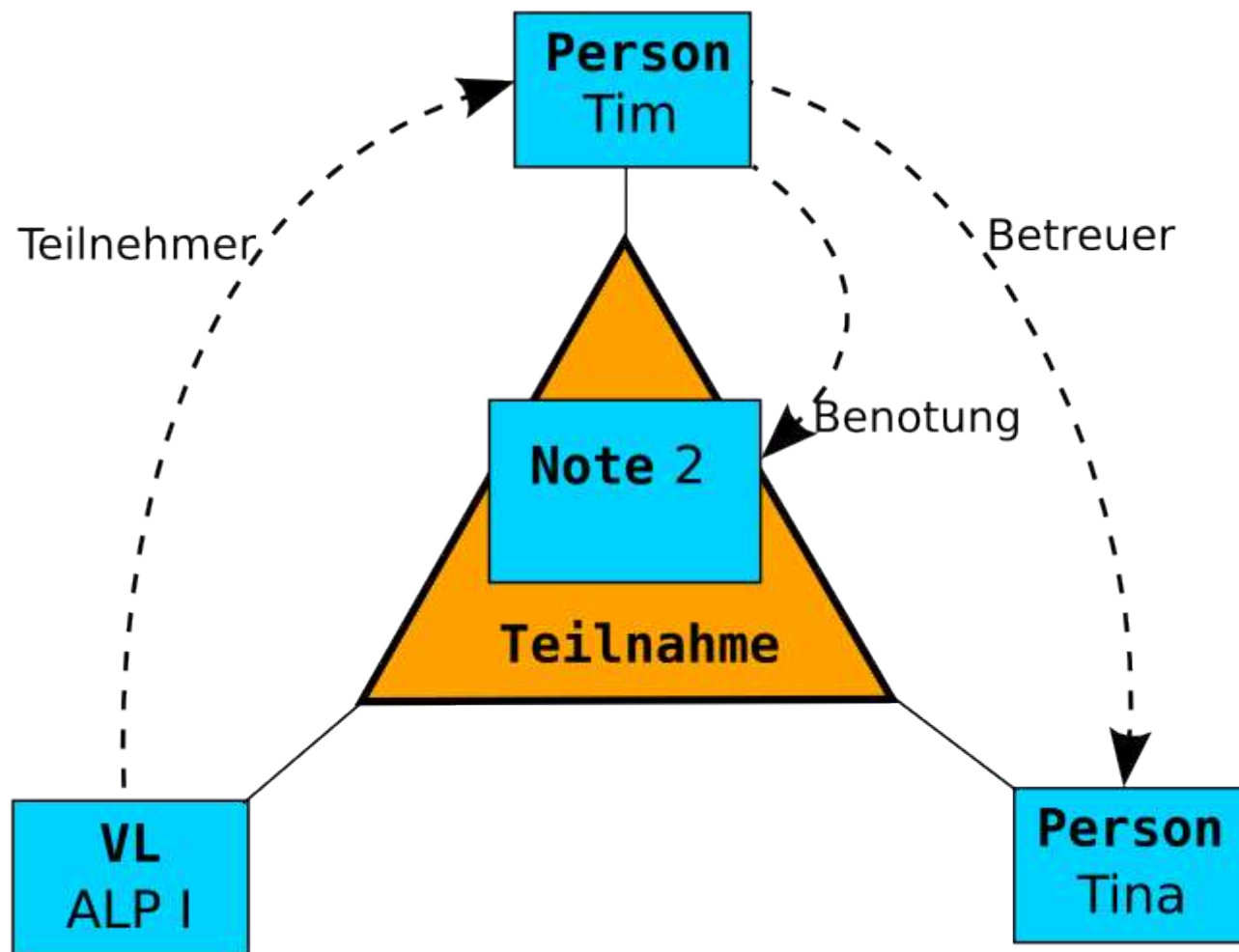
- XLink Elementen-Typ Attribut: `type`
- Lokalisierer Attribut: `href`
- semantische Attribute: `role`, `arcrole` and `title`
- Verhalten spezifizierende Attribute: `show` and `actuate`
- das Traversieren spezifizierende Attribute: `label`, `from` and `to`

Ein Beispiel

- Der Student Tim
 - hört die Veranstaltung ALP I
 - wird dabei von der Tutorin Tina betreut
 - erhält durch seine Teilnahme eine Note
- Dadurch werden Tim, Tina, ALP I und Tims Note in Beziehung zueinander gesetzt.
- Dies wollen wir durch einen Link darstellen.

Ein Beispiel (2)

- Zu Tim, Tina und ALP I gibt es bereits Ressourcen:
 - Tims Webseite: `students/tim.html`
 - Tinas Webseite: `students/tina.html`
 - ALP I Webseite: `lehre/alp1.html`
- Tims Note ist noch nirgendwo gespeichert.



```
<teilnahme xlink:type="extended">
  <person xlink:type="locator"
    xlink:href="student/tim.html"
    xlink:role="http://bsp.de/def/student"
    xlink:label="student" />
  <person xlink:type="locator"
    xlink:href="student/tina.html"
    xlink:role="http://bsp.de/def/tutor"
    xlink:label="tutor" />
  <vl xlink:type="locator"
    xlink:href="lehre/alp1.html"
    xlink:role="http://bsp.de/def/vorlesung"
    xlink:label="alp1" />
  <note xlink:type="resource"
    xlink:role="http://bsp.de/def/note"
    xlink:label="note">2</note>
```

```
<rel xlink:type="arc"
      xlink:from="student"
      xlink:to="tutor"
      xlink:arcrole="http://bsp.de/def/betreuer" />
<rel xlink:type="arc"
      xlink:from="alp1"
      xlink:to="student"
      xlink:title="zur Webseite von Tim"
      xlink:arcrole="http://bsp.de/def/teilnehmer" />
<rel xlink:type="arc"
      xlink:from="student"
      xlink:to="note"
      xlink:arcrole="http://bsp.de/def/benotung" />
</teilnahme>
```

```
<!ELEMENT teilnahme <(person|vl|note|rel)*>
<!ATTLIST teilnahme
  xmlns:xlink CDATA #FIXED
                                "http://www.w3.org/1999/xlink"
  xlink:type (extended) #FIXED "extended">

<!ELEMENT vl EMPTY>
<!ATTLIST vl
  xlink:type (locator) #FIXED "locator"
  xlink:href CDATA #REQUIRED
  xlink:role CDATA #FIXED
                                "http://bsp.de/def/vorlesung">

<!ELEMENT person EMPTY>
<!ATTLIST person
  xlink:type (locator) #FIXED "locator"
  xlink:href CDATA #REQUIRED
  xlink:role CDATA #REQUIRED>
```

XPointer

- in **href** kann auch auf Teile einer XML Datei verwiesen werden
- Syntax dazu definiert in XPointer
- XPointer besteht aus
 - Framework, xpointer-, element-, und xmlns-Schema-Spezifikationen
- XPointer benutzt Definitionen aus
 - XPath, XML-Infoset

XPointer Beispiele

→ `href="http://bsp.de/#xpointer(/3/27/6)"`

→ **XPointer Expressions sind sehr vielseitig**

- `id("inhalt")/li[3]`
- `//bank[@ort="berlin"]`
- `/chap[position()=3]/sec[position()<4]`
- `//kapitel/descendant::anfang/range-to(following::ende[1])`

Verhaltensattribute

- bezieht sich auf das Traversieren von arcs
- **show** – Wie soll Ziel dargestellt werden?
 - Werte: **new**, **replace**, **embed**, **other**, **none**
- **actuate** – Wann soll der arc traversiert werden?
 - Werte: **onLoad**, **onRequest**, **other**, **none**

Simple Links

- repräsentiert einen **arc** von einer lokalen zu einer remote Quelle
- ```
<meinlink xlink:type="simple"
 xlink:href="http://bsp.de">
 Hier kommt der Inhalt.
 </meinlink>
```
- unterstützt **role**, **arcrole**, **title**, **show**, **actuate**



# Beschränkungen von XLink

- pro Link höchstens ein **arc** von A nach B (verschiedene **arcroles** machen keinen Unterschied)
  - ♦ wollen wir einer Vorlesung Dozent und Tutoren zuordnen, so könnte keine Person sowohl Dozent als auch Tutor sein
  - ♦ wir müssten diese Information auf verschiedene Links verteilen

# Beschränkungen von XLink

- **arcroles** müssen absolute URIs enthalten
  - ♦ nur dazu gedacht dem **arc** eine semantische Bedeutung zuzuordnen
  - ♦ nicht dazu gedacht dem **arc** einen Wert zuzuordnen
    - z.B. eine Turing-Maschine könnte man nicht gut mit einem Xlink beschreiben

# Beschränkungen von XLink

- damit ein Link in einer HTML Seite auch eine von XLink definierte Bedeutung hat müsste man schreiben
  - `<a html:href="ziel.html" xlink:href="ziel.html">link</a>`
- man muss also hrefs in beiden Namespaces definieren, damit a eine sowohl von XHTML als auch von XLink definierte Bedeutung hat

# Verbreitung von XLink

- das W3C listet auf seiner Seite 7 Projekte auf, die XLink implementieren
- eines davon ist Mozilla, eines Amaya, der W3C eigene Webbrowser, beide implementieren nur das Traversieren von simple Links
- der Internet Explorer implementiert XLink überhaupt nicht

# Verbreitung von XLink

- lediglich 2 der aufgelisteten Projekte implementieren *extended link traversal*, ein einziges *third-party links*
- von den beiden Projekten, die *extended Links* implementieren, soll eines nicht außerhalb Japans veröffentlicht werden (Fujitsu), das andere ist aus dem Netz verschwunden
- *simple Links* sind Teil von SVG