Kapitel 9
XPointer und XLink

- Bisher behandelt: XML als Datenmodell, Datenrepräsentation, Anfragen
- immer nur einzelne Dokumente

World Wide Web

- Referenzen zwischen Dokumenten
- Links in HTML: zeigen auf ein Dokument, bestenfalls noch auf einen *Anker* in einem Dokument:
  `http://user.informatik.uni-goettingen.de/~may/Mondial/mondial.html#XML`
  (erreicht das, was dort mit <A name="XML"> gekennzeichnet ist)
- Browser: wenn man draufklickt, passiert etwas

Was bedeutet das für XML? Konzepte?

- eine Sprache zur Formulierung von Referenzen: XPointer
- eine Sprache zur Formulierung der Semantik von Referenzen: XLink

9.1 XPointer

- Links in HTML: zeigen auf ein Dokument, bestenfalls noch auf einen *Anker* in einem Dokument:
  `http://www.informatik.uni-goettingen.de/~may/Mondial/mondial.html#XML`
  (erreicht das, was dort mit <A name="XML"> gekennzeichnet ist)
- Ziel in XML: Zeiger auf irgendetwas in einem XML-Dokument setzen.
- Möglichkeiten, einzelne Elemente, Attribute, oder auch Zeichen eines XML-Dokuments zu adressieren:
  - Element-, Attribut-, Kommentar-, Processing-Instruction-Knoten,
  - jede "Information", wie sie durch Klicken/Markieren am Bildschirm ausgewählt werden kann, kann durch einen XPointer adressiert werden.
    (unabhängig von Elementgrenzen – kann in der Mitte eines Elements beginnen und in der Mitte eines anderen Elements enden).
  - jeder Punkt direkt vor oder nach einem Element kann adressiert werden.

XPOINTER

- XPointer sind ein *semantisches*, kein syntaktisches (bzgl. dem Zieldokument) Konzept. Sie müssen gegenüber rein mechanischen Änderungen im Zieldokument transparent sein (also nicht auf 3.Zeichen der 4.Zeile).
- erweitert URL-Konzept durch Verwendung von XPath:

  XPointer = *url#xpath-expr*

  .../Mondial/mondial.xml#descendant::country[@car_code="D"]
- volle Form:

  url#xpointer(*xpointer-expr*)
- Kurzform:

  url#xpointer-expr
- Dazu wird XPath um einige Konstrukte erweitert.

XPOINTER

- *xpath-expr*$_1$ to *xpath-expr*$_2$ ist ein Pointer, der einen Bereich in einem XML-Dokument selektiert

  Mondial.xml#//country[name="Germany"]/(city[0] to city[5])

  selektiert den Bereich von der 1. bis zur 6. Stadt Deutschlands in Mondial.xml.
  (*nicht* als Menge von Knoten, sondern als Bereich. Hier kann z.B. auch der Übergang zwischen zwei Bundesländern dabei sein – vgl.

  Mondial.xml#//country[name="Germany"]/descendants::city[0-5]

  wären die ersten 6 Stadt-Knoten!)
- string-range(*xpath-expr*, *string*, m, n) selektiert Zeichenketten in Dokumenten: Für jedes Ergebnis von *xpath-expr* wird das erste Auftreten von *string* gesucht, und vom m-ten Zeichen aus n Zeichen übernommen.
  Markup wird dabei ignoriert (einschließlich Attributwerten!)

Anmerkung: da es hier um Pointer geht, ist das Ergebnis kein Teil eines XML-Dokuments, sondern zwei Punkte!

XPointer: Beispiele

- Adressierung über die id-Funktion:

  Mondial.xml#id("D")
  - sehr robust gegenüber Änderungen der XML-Dokument-Struktur
  - benötigen Wissen über die Schemadefinition (ID-Deklaration)

- "objektorientierte" Addressierung über "Schlüssel":

  Mondial.xml#//country[name="Germany"]

9.2 XLink: World Wide Linking

- erweitere Möglichkeit, Hyperlinks zu spezifizieren.
- Beziehungen zwischen Resourcen (Dokumente, Elemente, ...)
  Resourcen können auch Programme, Bilder, Filme etc. sein!
- - Sprache: "XLink"
  - Namespace xlink:
- verwendet (natürlich) XPointer

Bedarfsanalyse

- Welche "Arten" von Referenzen will man haben?
  Genügt das, was HTML mit dem <a>-Tag bietet?
- Semantik von Referenzen?
  Klicken? was dann?
- ... XLink ist bisher nur für Browsing-Applikationen gedacht

SEMANTIK BESTEHENDER VERWEISTYPEN: HTML

HTML: <A HREF="*url#anker*">

- im Ausgangsdokument spezifiziert, unidirektional, nur ein Verweisziel,
- entweder die ganze Seite, oder auf einen vorbereiteten Anker.
- Verhalten?
  - Standard: wenn man draufklickt, wird die Zielseite im aktuellen Fenster angezeigt.
    Benutzeraktiviert, "Ersetzen"
  - Alternative: wenn man draufklickt, wird die Zielseite in einem neuen Fenster angezeigt.
    Benutzeraktiviert, "Neu"
  - Alternative: Beim Aufbau einer Seite wird stattdessen eine andere Seite im aktuellen Fenster angezeigt (Forwarding)
    automatisch aktiviert, "Ersetzen"
  - Alternative: Beim Aufbau einer Seiten werden andere Seiten in kleinen neuen Fenstern angezeigt
    automatisch aktiviert, "Neu"

... zum Durchklicken genügt das, für ein Datenmodell jedoch nicht.

SEMANTIK BESTEHENDER VERWEISTYPEN: ID/IDREF

ID/IDREF/IDREFS ist bereits ein Referenzmechanismus in XML: Einfachste Art von Verweisen *innerhalb eines XML-Dokuments*:

- unidirektional, dokumentenintern, ein oder mehrere Verweisziele
- "Aktivierung"? keine - existieren nur im Datenmodell

... für ein Datenmodell genügt es nicht, zum Durchklicken bringt es überhaupt nichts ...

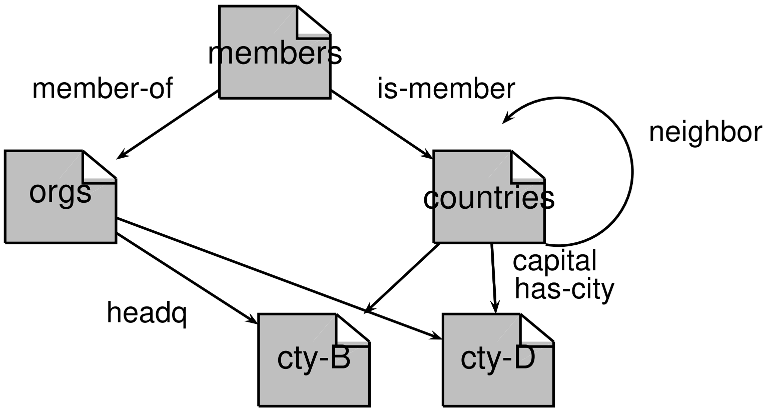
BEISPIEL-SZENARIEN

World-Wide-Web

- Web-Seiten
- Hyperlinks
- weitere Arten von Beziehungen zwischen Web-Seiten

Speicherung von XML-Daten in XML (Mondial)

- Verteilung auf mehrere Dokumente
  - mondial-countries.xml
  - mondial-cities-*car-code*.xml (Städte je eines Landes)
  - mondial-organizations.xml
  - mondial-memberships.xml



XLINK: GRUNDBEGRIFFE

- Resourcen: XML-Dokumente, Teile von XML-Dokumenten, HTML-Seiten, Bilder, Filme, Web-Services
- Links: Verbindungen zwischen Resourcen
  - lokale Resource: Resource, die als Struktur innerhalb eines Link-Elements liegt (oder das Link-Element selber)
  - entfernte (remote) Resource: Resource, die durch eine URI gegeben ist
- Arcs: gerichtete Verbindungen (Anfang $\rightarrow$ Ende)
  - outbound: der Anfang ist eine lokale Resource, das Ende eine entfernte.
    `<a href=‘‘...’’>`, Land-Hauptstadt-Beziehung
  - inbound: der Anfang ist eine entfernte Resource, das Ende eine lokale.
    inbound-Arcs können also nicht im selben Dokument wie ihre Anfangs-Resource dargestellt werden.
  - third-party: Anfang und Ende sind entfernte Resourcen.
    z.B. eine persönliche WWW-Link-Basis
    Mitgliedschaften von Ländern in Organisationen

RESOURCEN UND ARCS: BEISPIELE

- `<a href='www.goettingen.de'>Göttingen</a>` ist ein einfaches Link:
  - verbindet die (lokale) Resource "Göttingen" (String zum Anklicken) mit der an der URL `www.goettingen.de` liegenden (entfernten) Resource (Web-Seite)
- `<img src='...'>` ist ein noch einfacheres Link:
  - hat keine lokale Resource, sondern zeigt nur auf eine entfernte (die beim Aufbauen der Seite im Browser automatisch eingebunden wird)
- eigene Link-Basis über das WWW: verbindet jeweils entfernte Resourcen (einen Bereich eines HTML-Dokuments mit einer URL).
- in der Beziehung country/capital ist ein country-Element die eine Resource, und ein city-Element die andere.
  (in unserer Modellierung mit einer lokalen und einer entfernten Resource)
- in der Mitgliedschafts-Beziehung von Ländern in Organisationen sind je ein Land und eine Organisation die beiden beteiligten Resourcen
  - n:m-Beziehung
  - Modellierung: mit zwei entfernten Resourcen und einem third-Party Arc.

XLINK: VERWEISTYPEN UND -SEMANTIK

XLink bietet Meta-Semantik zur Beschreibung von Referenzen an, die von Anwendungen dann genutzt wird

- verschiedene Arten von Referenzen
  - einfach: wie `<a href='...'>` (zum Durchklicken)
    `<img src='filename'>` (automatisch beim Laden aktiviert)
  - Verweise auf mehrere Ziele/Resourcen/Dokumente
    mehrere Resourcen gleichzeitig aktivieren
    DB: ein Land hat mehrere Städte
  - die oben beschriebenen Verweise sind *inline*-Verweise, d.h. im Dokument selber enthalten (outbound arcs).
  - *out-of-line*-Verweise: Benutzer können selber Verweise zwischen ihnen nicht gehörenden (Mengen von) Dokumenten legen (third-party arcs).
    Eigene Sprung-Links "über das WWW" legen
    DB: Verbindungen zwischen Ländern und Organisationen
- Aktivierungszeitpunkt (onLoad, onRequest)
- Aktion (new, replace, embed)

XLINK-ELEMENTE

- Element- und Attributnamen aus dem *xlink:* namespace
- *jedes* Element kann damit ein Link werden
- Durch die Angabe eines *xlink:type*-Attributes mit einem der von XLink definierten Werte wird mitgeteilt, daß das definierte Element Link-Element-Funktionalität besitzt
- Eigenschaften und Substrukturen (aus einer in XLink vordefinierten Auswahl) können dann angegeben werden.

<!ELEMENT *linkelement* (*contentsmodel*)>
<!ATTLIST *linkelement*
    xmlns:xlink CDATA #FIXED "http://www.w3.org/xml/xlink"
    xlink:type (simple|extended|locator|resource|arc) #FIXED "..."
    ... >

- verschiedene Möglichkeiten für *contentsmodel* (je nach xlink:type werden einige Dinge vorgegeben).
- diverse zusätzliche Attribute (je nach xlink:type)

XLINK-ELEMENTTYPEN

- Definition beliebiger *linkelement*-Elementtypen,
- XLink sieht 5 grundlegende Arten von Elementtypen vor:
  - simple: etwa das was man von <A href="..."> kennt,
  - extended: mehrere Ziele,
  - locator: wird in extended benötigt um Ziele (entfernte Resourcen) zu beschreiben,
  - resource: wird in extended benötigt um Ziele (lokale Resourcen) zu beschreiben,
  - arc: wird in extended Links benötigt, um Arcs in einer Menge von Locator- und Resource-Elementen zu definieren.

XLINK-ELEMENTE UND IHRE ATTRIBUTE

Strukturaufbau

- xlink:type: Auswahl des Link-Typs.
- xlink:href: Angabe eines Ziels (bei simple und locator).
- xlink:label: Namen für Locators und Resourcen.
- xlink:from, xlink:to: bei Arcs – referenziert auf xlink:label.

Verhalten

- xlink:actuate: Gibt an, wann der Verweis "aktiviert" wird:
  - onLoad: beim Laden und Parsen des XML-Dokuments,
  - onRequest: erst wenn der Benutzer etwas macht (z.B. anklicken).
- xlink:show: Gibt an, was passiert, wenn das Link aktiviert wird:
  - new: Verweisziel erscheint in einem neuen Fenster,
  - replace: Das aktuelle Dokument wird ducrh das Verweisziel ersetzt,
  - embed: Das Verweisziel wird in das aktuelle Dokument eingefügt.

XLINK-ELEMENTE UND IHRE ATTRIBUTE (FORTS.)

Weitere Attribute (Anwendungssemantik)

- xlink:title: Hier kann der Anwender beliebig eine Charakterisierung des Verweisziels angeben.
- xlink:role und xlink:arcrole: Durch diese Attribut kann der Anwender z.B. Links gruppieren, z.B. xlink:role="Annotation"; man könnte dann einen Server haben, der nur Annotations verfolgt.
- xlink:arcrole wird insbesondere im Zusammenhang mit dem RDF (Resource Description Format) verwendet:

  object has property *rolename value*

  (RDF wird u.a. auch im Bereich Semsntic Web verwendet)

SIMPLE LINKS

- Durch Setzen des xlink:type-Attributs auf “simple” wird ein einfaches Link, analog dem <A>-Tag in HTML erzeugt.
- Dann muss das xlink:href-Attribut das Ziel der Referenz angeben
- xlink:actuate und xlink:show spezifizieren das Verhalten.

Beispiel: <A>-Element in HTML (bekanntes Verhalten)

```
<!ELEMENT A (#PCDATA)>
<ATTLIST A xmlns:xlink CDATA #FIXED "http://www.w3.org/xml/xlink"
    xlink:type (simple|extended|locator|resource|arc) #FIXED "simple"
    xlink:href CDATA #REQUIRED
    xlink:actuate (onLoad|onRequest) "onRequest"
    xlink:show (new|embed|replace) "new" >
```

Ein mögliches Element, das ein HTML-Fragment auf Klicken *einbettet*:

```
Um die ganze Liste zu sehen, klicken Sie
<A xlink:href="liste-fragment.html" xlink:show="embed">hier</A>.
```


XLINK: BEISPIEL

- country/capital als simple link

```
<!ELEMENT country (name,population?, ..., capital,cities, ...)>
<!ELEMENT capital EMPTY>
<!ATTLIST capital xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink"
                  xlink:type (simple|extended|locator|resource|arc)
                        #FIXED "simple"
                  xlink:href CDATA #IMPLIED>
```

Im XML-Dokument:

```
<country car_code="D" area="356910">
  <name>Germany</name>
    :
  <capital xlink:type="simple"
           xlink:href="cities-D.xml#//city[name='Berlin']"/>
</country>
```

- Das (leere) capital-Element ist die lokale Resource,
  das Link besteht zwischen country/capital und Berlin,
  nicht zwischen country und Berlin.

XLINK:AUFGABEN

Aufgabe 11 (Links in HTML)

- Definieren Sie ein <IMG>-Element mit XLink, das dasselbe Verhalten hat, wie das <IMG>-Element in HTML. □

XLINK: INLINE EXTENDED LINKS

- Extended Links können auf mehrere Resourcen zeigen.
- dafür fällt das href-Attribut weg,
- die Zielreferenzierung übernehmen die locator-Elemente
- ausserdem sind resource-elemente möglich.

```
<!ELEMENT linkelement (... locatorelement* ... resourceelement* ...)>
<!ATTLIST linkelement
      xmlns:xlink   ...     ...
      xlink:type    (simple|extended|locator|resource|arc)  #FIXED "extended" >

<!ELEMENT locatorelement (contentsmodel)>
<!ATTLIST locatorelement
      xmlns:xlink   ...     ...
      xlink:type   (simple|extended|locator|resource|arc)  #FIXED "locator"
      xlink:href   CDATA  #REQUIRED >
```

analog für *resourceelement*.

BEISPIEL: EXTENDED LINK IM WWW

Bereitstellung von Bildern im WWW:

```
<!ELEMENT a-loc EMPTY>
<!ATTLIST a-loc xlink:type (...) FIXED "locator"
                xlink:href CDATA
                xlink:show (new|embed|replace) "new"
                xlink:actuate (onLoad|onRequest) "onLoad">
<ul> <li> <ext-a xlink:type="extended">Goettingen
            <a-loc xlink:type="locator" xlink:href="goe1.jpg">
            <a-loc xlink:type="locator" xlink:href="goe2.jpg">
            <a-loc xlink:type="locator" xlink:href="goe3.jpg">
          </ext-a>
     <li> <ext-a xlink:type="extended">Freiburg
            <bla-loc xlink:type="locator" xlink:href="fr1.jpg">
            <bla-loc xlink:type="locator" xlink:href="fr2.jpg">
          </ext-a>
</ul>
```

Wenn der Benutzer (in einem XLink-fähigen Browser) auf ein solches `<ext-a>`-Element klickt, werden alle darin angegebenen Bildchen in neuen Fenstern geöffnet.

BEISPIEL: EXTENDED LINK IN MONDIAL

Man kann country-Elemente auch als extended Links auffassen:

- damit ist jedes country-Element als Ganzes eine lokale Resource
- mit eigenen Resourcen: Name etc.
- mit Referenzen (Locators) auf andere Resourcen

```
<!ELEMENT country (name,...,capital,encompassed*,border*,cities)>
<!ATTLIST country car_code ID #IMPLIED
                  :
                  xlink:type CDATA #FIXED 'extended'>
<!ELEMENT capital EMPTY>
<!ATTLIST capital xlink:type CDATA #FIXED 'locator'
                  xlink:href CDATA #REQUIRED >
```

Analog für encompassed, border, provinces, cities.

Aufgabe 12
Geben Sie die DTD für diese Elemente komplett an (beachten Sie, dass encompassed und border noch weitere Attribute haben). □

BEISPIEL (FORTSETZUNG)

XML-Instanz:

```
<country xlink:type="extended"
         car_code="D" area="356910" ...>
  <capital xlink:type="locator"
           xlink:href="cities-D.xml#//city[name='Berlin']"/>
  <cities xlink:type="locator"
           xlink:href="cities-D.xml#//city"/>
   :
</country>
```

Aufgabe 13
Vervollständigen Sie auch die XML-Instanz. □

BEISPIEL (FORTSETZUNG)

- Hier sind xlink:show und xlink:actuate egal
- Anwendung: nicht Browsing, sondern Anfragen, Transformationen etc
- weder in XPath/XQuery noch XLink betrachtet
- Fragestellung wird im Projekt LinXIS
  `http://www.dbis.informatik.uni-goettingen/LinXIS`
  behandelt (siehe später).

- bisher: nur outbound Links
  - country/capital ist im obigen Beispiel ein impliziter outbound-arc von der lokalen Resource (country) auf eine Stadt.
  - country/cities referenziert mehrere Ziele (city-Elemente), definiert mehrere outbound arcs.

XLINK: OUT-OF-LINE-LINKS

- Verweis-Elemente nicht im Dokument, sondern in separaten Dokumenten (d.h., man kann Verweise in fremde Dokumente “einfügen”),
- Link wird hier nicht nur als (gerichteter) Verweis, sondern als Beziehung zwischen Resourcen gesehen,

- Verwendet wieder Extended-Elemente mit Locators
  diese sind mit xlink:label-Attribut versehen.
- Neben den Locator-Elementen (die die (Remote-)Resourcen lokalisieren) muß zusätzliche Information gespeichert werden:
  - welche der angegebenen Resourcen durch einen Arc verbunden werden sollen,
  - sowie die Richtung der Verbindung

⇒ zusätzliche *arc*-Elemente
benennen Resourcen/Locators durch xlink:from und xlink:to-Attribute

XLINK: OUT-OF-LINE-LINKS

- Der Elementinhalt muß nun Subelemente von Locator-Elementtypen (wie oben) sowie Subelemente von Arc-Elementtypen zulassen, die die Locator-Elemente in Beziehung zueinander setzen:

<!ELEMENT *linkelement* (... *locatorelement** ... *resourceelement** ...)>
<!ATTLIST *linkelement* wie oben >
<!ELEMENT *locatorelement* wie oben >
<!ATTLIST *locatorelement* wie oben
    type, href, label: NMTOKEN , title, role >
<!ELEMENT *arcelement* (*contentsmodel*)>
<!ATTLIST *arcelement*
    xmlns:xlink CDATA #FIXED “http://www.w3.org/xml/xlink”
    xlink:type (simple|extended|locator|arc) #FIXED “arc”
    xlink:from NMTOKEN #REQUIRED
    xlink:to NMTOKEN #REQUIRED >

- Hinweis: xlink:label ist nicht ID; es kann mehrere gleiche Labels geben.

Beispiel Out-of-Line-Links: Web

Möglichkeit, eine eigene Link-Basis zusätzlich zu den vorhandenen `<a href=‘‘..’’>`-Einträgen zu erstellen:

- Zwei Arten von Locator-Elementen:
  1. XPointer auf Stichworte und Sätze, von denen das Link ausgehen soll:
     ```
     <entry xlink:type="locator" xlink:label="univis-ssdxml"
            xlink:href="www.univis.goe/irgendeinpfad#//h1[text()='...']">
     ```
  2. URLs von Seiten, die als Sprungziele angegeben werden sollen
     ```
     <target xlink:type="locator" xlink:label="ssdxml-homepage"
             xlink:href="www.cs.uni-goettingen.de/teaching/ssdxml.html">
     ```
- Arc-Elemente:
  ```
  <arc xlink:type="arc"
       xlink:from="univis-ssdxml" xlink:to="ssdxml-homepage">
  ```
- Wenn man einen XLink-fähigen Browser hat, der die eigene Link-Basis auswertet, hat man jetzt beim Betrachten der Univis-Seite ein Link von der Überschrift auf die XML-Vorlesung.

Beispiel Out-of-Line-Links: Mondial

Mitgliedschaften von Ländern in Organisationen

```
<!ELEMENT memberships (country*,organization*,membership*)>
<!ATTLIST memberships xlink:type (...) #FIXED “extended” >

<!ELEMENT country EMPTY>
<!ATTLIST country xlink:type   (...) #FIXED “locator”
                  xlink:href   CDATA #REQUIRED
                  xlink:label  NMTOKEN #REQUIRED >

<!ELEMENT organization EMPTY>
<!ATTLIST organization xlink:type   (...) #FIXED “locator”
                       xlink:href   CDATA #REQUIRED
                       xlink:label  NMTOKEN #REQUIRED >

<!ELEMENT membership EMPTY>
<!ATTLIST membership xlink:type   (...) #FIXED “arc”
                     xlink:from        NMTOKEN #REQUIRED
                     xlink:to          NMTOKEN #REQUIRED
                     membership_type CDATA #REQUIRED >
```

Beispiel Out-of-Line-Links: Mondial

<memberships>
<country xlink:label="B" xlink:type="locator"
xlink:href="../countries.xml#//country[@car_code='B']"/>
<country xlink:label="D" xlink:type="locator"
xlink:href=".../countries.xml#//country[@car_code='D']"/>
<organization xlink:label="org-EU" xlink:type="locator"
xlink:href=".../organizations.xml#//organization[@abbrev='EU']"/>
<organization xlink:label="org-UN" xlink:type="locator"
xlink:href="../organizations.xml#//organization[@abbrev='UN']"/>
<membership xlink:from="B" xlink:to="org-EU" xlink:type="arc"
membership_type="member"/>
<membership xlink:from="B" xlink:to="org-UN" xlink:type="arc"
membership_type="member"/>
</memberships>

Semantik von Arcs

- Sind alle xlink:label innerhalb eines Extended-Elements eindeutig, so steht jedes arc-Element für die durch sein xlink:from und xlink:to gegebene einzelne Verbindung.
- Kommen xlink:labels innerhalb eines Extended-Elements mehrfach vor, so steht jeder arc für alle Verbindungen zwischen Locator-Paaren, die die entsprechenden from- und to-Labels haben.
- hat ein arc kein xlink:to-Attribute, so steht er für Verbindungen zu jedem einzelnen Locator (analog für from).
- hat ein arc weder from, noch to, steht er für alle möglichen Verbindungen.

XLINK: VERWENDUNG

- Browsing: klar. xlink:show und xlink:actuate
  Amaya (www.w3.org/Amaya) versteht XLink und ist open-source.
- Anfragen an XML-Datenquellen:
  - Verhalten von XPath zu XLink bisher in den W3C-Dokumenten nicht betrachtet
  - erfordert momentan echte Programmierung, die XLink-Elemente auflöst, die erhaltenen XPath-Ausdrücke dynamisch evaluiert und die Anfrage weitergibt
  - Forschung der DBIS-Arbeitsgruppe Göttingen: Projekt LinXIS:
    Linked XML Information Sources
    * dbxlink:-Namespace zur Beschreibung des Datenmodells
    * Links "transparent" machen:
      `mondial/countries.xml#/country[name=‘‘Germany’’]/capital`
      löst das xlink auf und "springt" automatisch zum Ziel der Referenz nach
      `mondial/cities-D.xml#//city[name=‘‘Berlin’’]`
    * Auswertung des XPath-Ausdruckes wird ggf. dort fortgesetzt
    * Projektinformationen auf
      `http://www.dbis.informatik.uni-goettingen.de/LinXIS`