Chapter 9
XPointer and XLink

- Considered up to now: XML as a data model, data representation, queries
- only single, isolated documents

World Wide Web

- references between documents,
- links in HTML: point to a document, sometimes to an *anchor* in a document:
  `http://user.informatik.uni-goettingen.de/~may/Mondial/mondial.html#XML`
  (the target must be prepared in the remote document with `<A name="XML">`),
- browser: when clicking on a link, something happens.

What does that mean for XML? – concepts?

- a language for expressing references: XPointer
- a language for specifying the semantics of references: XLink

9.1 XPointer

- links in HTML: point to a document, sometimes to an *anchor* in a document:
  `http://user.informatik.uni-goettingen.de/~may/Mondial/mondial.html#XML`
  (the target must be prepared in the remote document with `<A name=‘‘XML’’>`),
- Goal in XML: express a pointer to *something* in another XML document.
- possibilities to address individual elements, attributes, or also characters in an XML document:
  - element-, attribute-, comment-, processing instruction nodes,
  - all “information” that can be selected on the monitor by “mousing” can also be addressed by an XPointer.
    (independent from borders of elements – can start in the middle of an element and end in the middle of another element).
  - each point directly before or after an element can be addressed.

XPointer

- XPointer is a *semantical*, not a syntactical (wrt. the target document) concept. XPointers must be transparent against mechanical changes in the target document (i.e., not "point to the 3rd character in the 6th line in the browser").
- extends the URL concept with the use of XPath:

  XPointer = *url#xpointer-expr*

  http://.../Mondial/mondial.xml#xpointer(descendant::country[@car_code="D"])
- "shorthand pointer": *url#id*
  – as in HTML: <a name="bla"> and <a href="http://filename#bla">
  addresses the element that has *id* as its ID-value
  (DTD: value of an attribute declared as ID)
- full form – "xpointer scheme" (there are also other schemes):

  url#xpointer(*xpointer-expr*)
- For this, XPath is extended with some constructs.
- alternative: element() scheme, e.g. element(D), element(/1/4/3), element(D/8/3)
  (last: third child of the eight child of the element identified by "D")

XPointer

- every XPath epression is also an XPointer expression
- *xpath-expr*$_1$ to *xpath-expr*$_2$ is a pointer, that selects an area in an XML document:

  mondial.xml#xpointer(//country[name="Germany"]/city[1]/range-to(../city[6]))

  selects the area from the 1st to the 6th city of Germany in mondial.xml.
  (*not* as set of nodes, but as an area. This can e.g. include changing from one province element to another).
- string-range(*xpath-expr*, *string*, m, n) selects sequences of characters in documents: for each result of *xpath-expr*, the first occurrence of *string* is searched, and the characters from positions m-n are "referenced".
  Markup is ignored in this sequence (including attribute values!)

Remark: since we speak about pointers, the result is not a fragment of an XML document, but simply two positions in a document!

XPOINTER: EXAMPLES

- Addressing via the id-function:

  mondial.xml#xpointer(id(“D”))
  shorthand: mondial.xml#D

  - robust against changes in the XML document structure,
  - requires knowledge about the schema definition (ID-declaration)

- “object-oriented” addressing via semantic “keys”:

  mondial.xml#xpointer(//country[name=“Germany”])
  mondial.xml#xpointer(//country[name=“Germany”]//city[name=”Berlin”])

9.2 XLink: World Wide Linking

- extended possibility for specifying hyperlinks.
- relationships between resources (documents, elements, ...)
  resources can also be programs, images, films, etc.
- - Language: “XLink”
  - Namespace xlink:
- uses (naturally) XPointer

Requirement Analysis

- What “kinds” of references are needed?
  Is the functionality of HTML’s <a>-tag sufficient?
- semantics of references?
  click? and then?
- ... up to now, XLink is officially only investigated for browsing applications.

Semantics of Existing Reference Types: HTML

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

- specified in the source document, unidirectional, only one target,
- either the whole page, or to a predefined anchor.
- behavior?
  - standard: when clicked, the target page is shown in the current window.
    user-activated, "replace"
  - alternative: when clicked, the target page is shown in a new window.
    user-activated, "new"
  - alternative: instead of building up a page, another page is shown in the current window (forwarding)
    automatically activated, "replace"
  - alternative: when building up a page in the browser, other pages are shown in small, separate windows
    automatically activated, "new"

... sufficient for clicking/browsing, but not for a data model.

HTML: <IMG SRC="*url*"/>

... is also a "link"!

- specified in the source document, unidirectional, only one non-HTML/XML target,
- behavior?
  - standard: when the page is loaded, the image is embedded at the given position.
    automatically activated, "embed"
  - alternative: when building up a page in the browser, show pictures in small, separate windows
    automatically activated, "new"

Semantics of existing Reference Types: ID/IDREF

ID/IDREF/IDREFS is already a reference mechanism in XML: Simplest kind of references *inside an XML document*:

- unidirectional, internal to the document, one or more targets
- “Activation”?
  ... when a query is executed (dereferencing; “user-activated”)

... insufficient for a data model, useless for clicking ...

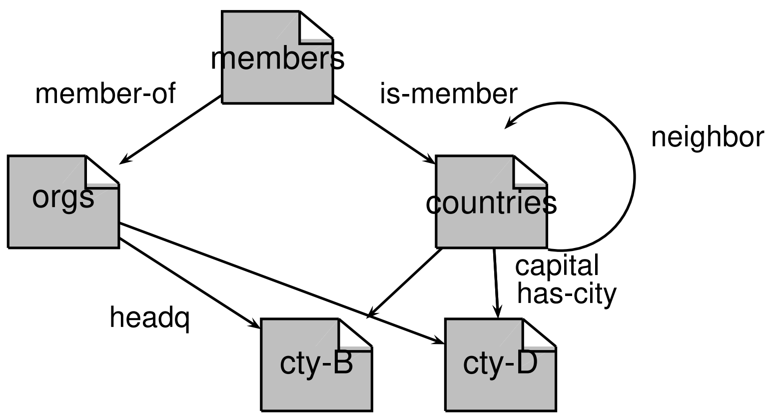
Example-Scenarios

World-Wide-Web

- Web pages
- Hyperlinks
- other kinds of relationships between Web pages

Storage of XML Data in XML (Mondial)

- Distribution over multiple documents
  - countries.xml
  - cities-*car-code*.xml
    (cities and provinces of each country)
  - organizations.xml
  - memberships.xml



XLink: Basic Notions

Resources: XML documents, parts of XML documents, HTML pages, images, films, Web services ...

- local resource: a resource that belongs as a structure to the content of the XLink element itself (or that is the link itself)
- remote resource: a resource that is given by a URI

Examples

- `<a href='http://www.goettingen.de'>Göttingen</a>` is a simple link:
  connects the (local) resource "Göttingen" (string to be clicked) with a (remote) resource located at the URL `www.goettingen.de` (Web page).
- `<img src='...'/>` is an even simpler link:
  has no local resource, but points only to a remote one

XLink: Basic Notions (Cont'd)

Arcs: directed connections between resources (starting point $\rightarrow$ endpoint)

- outbound: the starting point is a local resource, the end is a remote resource.
  - `<a href="..."> ...</a>`,
  - country-capital-relationship: a country element is the local resource, and city element is the other, remote, resource.
- inbound: the starting point is a remote resource, the endpoint is a local one.
  Inbound-arcs cannot be represented in the same document as their starting point.
- third-party: starting point and endpoint are remote resources.
  - e.g. own linkbase over the Web: each link connects two remote resources (an area of an HTML document with another URL).
  - e.g. memberships of countries in organizations:
    * each link connects two remote resources, a country and an organisation
    * n:m-relationship ... see later

XLink: Kinds of Links and their Semantics

XLink offers a meta-semantics for describing references that is then used by applications.

- different kinds of references
  - simple: like `<a href='...'>...</a>` or `<img src='filename'/>`
  - links to multiple targets/resources/documents
    activate several resources at the same time
    DB: a country has several cities
  - the links described above are *inline*-links, i.e., contained in the document itself (outbound arcs).
  - *out-of-line*-links: a user can define connections between (sets of) documents that are owned by somebody else (third-party arcs).
    "overlay" own hyperlinks for clicking over the Web
    DB: connections between countries and organizations
- timepoint of activation (onLoad, onRequest)
- action (new, replace, embed)

XLink Elements

- Element- and attribute names from the *xlink:* namespace
- *Each* element can become a link ...
- ... by adding an *xlink:type* attribute having one of the values defined by XLink, the element is assigned XLink functionality.
- Properties and substructures (chosen from a predefined set of XLink behavior) can then be specified.

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

- different possibilities for the *content model* (depending on xlink:type, some things are required).
- additional attributes (depending on xlink:type)

XLINK ELEMENT TYPES

- Definition of arbitrary *link element* types,
- XLink defines 5 basic types of element types:
  - simple: similar to what is known from HTML's <A href="...">,
  - extended: multiple targets,
  - locator: is used in extended links for specifying individual remote resources,
  - resource: is used in extended links for specifying individual local resources,
  - arc: is used in extended links for specifying connections between locator- and resource elements.

XLINK -ELEMENTS AND THEIR ATTRIBUTES

Structure

- xlink:type: chooses a link type,
- xlink:href: specification of target(s) (for simple or locator elements)
  (note that an XPointer can specify multiple targets)
- xlink:label: give names for locators and resources.
- xlink:from, xlink:to: for arcs – references to an xlink:label.

Behavior

- xlink:actuate: specifies, when the link is "activated":
  - onLoad: when loading/parsing the XML document,
  - onRequest: only when the user does something (e.g. clicking).
- xlink:show: specifies what happens when the link is activated:
  - new: target appears in a new window,
  - replace: the current document is replaced by the target,
  - embed: the target is embedded into the current document.

XLINK ELEMENTS AND THEIR ATTRIBUTES (CONT'D)

Further Attributes (application-specific)

- xlink:title: the user can give a comment about the target.
- xlink:role and xlink:arcrole: allows to group links to roles, e.g., xlink:role="Annotation" for giving additional information to the application.
- xlink:arcrole is mainly used in RDF (Resource Description Framework):

  object has property *rolename value*

  (RDF is e.g. used in the Semantic Web)

SIMPLE LINKS

- By setting xlink:type to "simple", a simple link analogous to the <A>-tag in HTML is defined.
- the xlink:href attribute specifies the target
- xlink:actuate and xlink:show specify the behavior.

Example: <A>-Element in HTML (known behavior)

```
<!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">
```

A sample element that embeds an HTML fragment when clicking on it:

```
For getting the whole list, please click
<A xlink:href="liste-fragment.html" xlink:show="embed">here</A>.
```

Simple Link: Example

- country/capital as 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>
```

In the XML document:

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

- the (empty) “capital” element is the local resource, the link is specified between country/capital and Berlin.

Exercise: Define the HTML <IMG> element with XLink.

XLink: Inline Extended Links

- Extended links can contain multiple XPointers,
- they have no own href attribute,
- the specification of the targets is done by locator subelements and resource subelements.

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

<!ELEMENT *locatorelement* (*contentmodel*)>
<!ATTLIST *locatorelement*
    xmlns:xlink ... ...
    xlink:type (simple|extended|locator|resource|arc) #FIXED “locator”
    xlink:href CDATA #REQUIRED >

resourceelement: any element can be made a local resource by xlink:type=“resource”.

Example: Extended Link in the WWW

Pictures in the Web:

```
<!ELEMENT a-loc EMPTY>
<!ATTLIST a-loc xlink:type (...) FIXED "locator"
                xlink:href CDATA>
<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>
```

when the user (in an XLink-enabled browser) clicks on such an <ext-a> element, all corresponding pictures are shown in separate new windows.

Example: Extended Link in Mondial

The country elements can also be regarded as extended links:

- each country element as a whole is a local resource
- with own resources: name etc.
- with references (locators) to other resources.

```
<!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 >
```

Analogously for encompassed, border, provinces, cities.

Exercise 9.1
Give a DTD for these elements (note that encompassed and border have further attributes). □

Example (cont'd)

XML instance:

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

Exercise 9.2
Complete the XML instance. □

Example (cont'd)

- the attributes xlink:show and xlink:actuate are not relevant here
- application: not browsing, but queries, transformations etc.
- not considered neither in XPath/XQuery nor in XLink.
- These aspects are investigated in the LinXIS project:
  `http://www.dbis.informatik.uni-goettingen/LinXIS`
  (see later)

- up to now: only outbound links
  - country/capital is an implicit outbound-arc from the local resource (country) to a city.
  - country/cities references multiple targets (city elements), defines multiple outbound arcs.

XLINK: OUT-OF-LINE-LINKS

Link elements that are not in the document, but in separate documents (i.e., possible to “add” links to other people’s documents):

- expressed by extended link elements with locators and resources;
  these are equipped with an xlink:label attribute.
- in addition to the locator elements (that address the (remote) resources), additional information must be stored:
  - which resources are connected by an arc,
  - and the direction of the connection.

⇒ additional *arc* elements
connect resources/locators by xlink:from and xlink:to attributes.

XLINK: OUT-OF-LINE-LINKS

- element content allows for subelements of locator element types (as above) and subelements of arc element types that describe relationships between locator elements:

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

- Note: xlink:label is not an ID; there can be several elements with the same label.

EXAMPLE OUT-OF-LINE-LINKS: WEB

Adding an own link base in addition to the <a href="..">-entries in HTML Web documents:

- two kinds of locator elements:
  1. XPointers on keywords and sentences where the link should start from:
     ```
     <univis xlink:type="locator" xlink:label="univis-ssdxml"
         xlink:href="www.univis.goe/somepath#xpointer(//h1[text()='...'])"/>
     ```
  2. URLs of target pages:
     ```
     <teaching xlink:type="locator" xlink:label="ssdxml-homepage"
         xlink:href="www.cs.uni-goettingen.de/teaching/ssdxml.html"/>
     ```
- Arc elements:
  ```
  <arc xlink:type="arc"
      xlink:from="univis-ssdxml" xlink:to="ssdxml-homepage"/>
  ```
- when using an XLink-enabled browser that has access to the link base, there is a link on the Univis page from the headline to the XML lecture.

EXAMPLE OUT-OF-LINE-LINKS: MONDIAL

Memberships of countries in organizations:

```
<!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  #IMPLIED
                     xlink:to            NMTOKEN  #IMPLIED
                     membership_type CDATA  #REQUIRED >
```

EXAMPLE OUT-OF-LINE-LINKS: MONDIAL

```
<memberships>
  <country xlink:label="B" xlink:type="locator"
      xlink:href="../countries.xml#xpointer(//country[@car_code='B']")/>
  <country xlink:label="D" xlink:type="locator"
      xlink:href="…/countries.xml#xpointer(//country[@car_code='D'])"/>
  <organization xlink:label="org-EU" xlink:type="locator"
      xlink:href="…/organizations.xml#xpointer(//organization[@abbrev='EU'])"/>
  <organization xlink:label="org-UN" xlink:type="locator"
      xlink:href="…/organizations.xml#xpointer(//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>
```


SEMANTICS OF ARCS

- In case that all xlink:label in an extended link element are unique, each arc element stands for the unique relationship given by the xlink:from and xlink:to attributes.
- In case that the labels are not unique, every arc stands for all relationships between pairs of locators that have the corresponding from- and to-labels.
- an arc that has no xlink:to attribute, stands for a connection to each locator (analogously for from).
- an arc that has neither from nor to stands for all possible relationships.

XLINK: USAGE

Browsing: obvious. xlink:show and xlink:actuate

- W3C Amaya (http://www.w3.org/Amaya): partially understands XLink and is open-source
  - use XLink for annotations to Web pages (→ RDF)

  Possible BSc Thesis: extending Amaya functionality for XLink (C++)
- queries against XML data sources:
  - The *W3C XML Query Requirements* state that the query language must support queries over references. The XLink/XQuery combination does not (yet) satisfy this.
  - behavior of XPath and XLink has not yet been considered in the W3C documents:
  - there is even no *data model* for XLinks
  - currently: requires real programming for resolving XLink elements and evaluating the references dynamically.

9.3 XInclude: Database-Style Use of XPointer

Include-elements are *replaced* by the corresponding included items:

<xi:include parse= “xml|text” href= “*url*” xpointer= “*xpointer*”/>

- no browsing semantics (XHTML: include must be resolved when loading)
- query/database semantics: obvious

```
<country xlink:type="extended" car_code="D" area="356910" ...>
  <xi:include parse="xml" href="mondial-D-cities.xml" xpointer="//city"/>
</country>
```

becomes

```
<country xlink:type="extended" car_code="D" area="356910" ...>
  <city><name>Berlin</name> ... </city>
  <city><name>Stuttgart</name> ... </city>
  :
</country>
```

  - resolve inclusion when loading
  - resolve inclusion on-demand when querying

9.4 The LinXIS Project – Linked XML Information Sources

Research project in the DBIS group at Göttingen:

... combine XLink with XInclude-style functionality

- extend XLink with data model semantics: insert referenced targets
- dbxlink: namespace for specifying the data model

```
<country car_code="D" area="356910" ...>
 <capital xlink:type="simple" dbxlink:transparent="keep-element insert-contents"
       xlink:href="cities-D.xml#xpointer(//city[name='Berlin'])"/>
</country>
```

```
<country car_code="D" area="356910" ...>
 <capital><name>Berlin</name><population>3472009</population></capital>
</country>
```

- query: `document(countries.xml)//country[name="Germany"]/capital/population)` resolves the xlink and “jumps” automatically to the target of the reference to `mondial/cities-D.xml#xpointer(//city[name="Berlin"])`
- http://www.dbis.informatik.uni-goettingen.de/LinXIS