Lecture 9: XPointer, XInclude and XML applications

Maciej Ogrodniczuk

MIMUW, 6 December 2010
The standard defines addressing XML documents and their fragments using standard URI syntax: http://www.sejm.gov.pl/ustawa.xml#forest-def

3 W3C recommendations dated 2002-03:

- XPointer Framework
  (http://www.w3.org/TR/xptr-framework/),

- XPointer element() Scheme
  (http://www.w3.org/TR/xptr-element/),

- XPointer xmlns() Scheme
  (http://www.w3.org/TR/xptr-xmlns/),

- XPointer xpointer() Scheme
  (http://www.w3.org/TR/xptr-xpointer/, Working Draft).
Lecture 9: XPointer, XInclude and XML applications

**xpointer:**
- XPaths:
  - xpointer(/art[5]/par[2])

**xmlns:**
- Namespaces
- To be used in further part of the path:
  - ustawa.xml#xmlns(pr=http://www.sejm.gov.pl/prawo)
  - xpointer(/art[5]/par[2])
Przykłady:

- element carrying ID attribute with given value:
  document.xml#element(def-forest)

- element with given position (absolute or relative to element carrying ID with given value):
  document.xml#element(/1/4/3)
  document.xml#element(def-forest/2/3)

- short syntax:
  document.xml#def-forest
  document.xml#/1/4/3
  document.xml#def-forest/2/3
XInclude — including XML content

The idea: include (parts of) one XML document to another XML document.

W3C Recommendation XML Inclusions (XInclude) 1.0 from 2006.

Capabilities:

- inclusion of complete documents (identified by filename or URL),
- inclusion of fragments (identified by XPointer),
- defining content to be used in case of an error.

http://www.w3.org/TR/xinclude/
Source:

```xml
<result>
  <xi:include
      xmlns:xi="http://www.w3.org/2001/XInclude"
      href="salad.xml#xpointer(/recipe/title)">
    <xi:fallback>
      <error>No such recipe.</error>
    </xi:fallback>
  </xi:include>
</result>
```

Result of the processing:

```xml
<result>
  <title>Cucumber salad</title>
</result>
```
Typical XML applications

Possibilities:

- content management (documents created by and for people, long-lasting and important), e.g. encyclopaedias,
- electronic data interchange — communication between applications (documents created and processed automatically, usually deleted after communication ends), e.g. XML-encoded error messages,
- many, many more!

There are plenty of applications since there can be many XML data structures to be used for different purposes.
Popular XML applications

Different document models (DTD/XML Schema) for different purposes:

- **text encoding:**
  - XML DocBook,
  - TEI – Text Encoding Initiative,
  - ODF (OpenDocument),
  - Office Open XML,

- **metadata encoding:**
  - Dublin Core,
  - RDF – Resource Description Framework,
  - Topic Maps,

- **scientific data representation:**
  - MathML – Matematical Markup Language,

- **multimedia:**
  - SVG – Scallable Vector Graphics.
DocBook

Designed for software documentation:
- book and article structure,
- additional software documentation specific structures.

Customizable:
- highly parametrized DTD (allowing to include/exclude element sets),
- DTD modification not needed in most cases.

Project started in 1991, current version: 5.1b2, maintained by OASIS (Organization for the Advancement of Structured Information Standards).
Enthusiastically adopted by open-source environment, used for documentation of many projects (KDE, GNOME, Linux kernel, ...)

Mature standard, many tools available (WYSIWYG editors, XSLT transformations, ...)

http://www.docbook.org
<book xml:id="myBook" version="5.0"
     xmlns="http://docbook.org/ns/docbook">
  <title>Sophie’s World</title>
  <chapter xml:id="chap1">
    <title>The Garden of Eden</title>
    <para>He who cannot draw on three thousand years Is living from hand to mouth</para>
    <para>--- <emphasis>Goethe</emphasis></para>
  </chapter>
  <chapter xml:id="chap2">
    <title>The Top Hat</title>
    <para>The only thing we require to be good philosophers is the faculty of wonder.</para>
  </chapter>
</book>
TEI (Text Encoding Initiative) — a standard for encoding of electronic texts in the humanities:

- work started in 1987,
- previously SGML-based, currently in XML,
- most recent version: P5 from 2007,
- the schema contains modules (tag sets) included when necessary (＝ playing Lego).
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader> ... </teiHeader>
  <text>
    <body>
      <schemaSpec ident="odd1" start="TEI">
        <moduleRef key="header"/>
        <moduleRef key="core"/>  
        <moduleRef key="tei"/>
        <moduleRef key="textstructure"/>
      </schemaSpec>
      ... 
    </body>
  </text>
</TEI>
Lecture 9: XPointer, XInclude and XML applications

XML and Content Management
Lecture 9: XPointer, XInclude and XML applications

XML and Content Management 14
MathML — Mathematical Markup Language

MathML can be used for representing equations and mathematical symbols.

Current version: 3.0 (from October 21, 2010).

Two MathML dialects (may be combined):

- **presentation markup** – 30 elements, 50 attributes,
- **content markup** – approx. 100 elements, 12 attributes,
- **strict content markup** – a subset of content markup: a minimal set of elements to represent the meaning of a mathematical expression in a uniform structure: 13 elements.

http://www.w3.org/TR/MathML3/
MathML — an example

\[(a + b)^2\]

Presentation markup:

```
<mrow>
  <msup>
    <mfenced>
      <mrow>
        <mi>a</mi>
        <mo>+</mo>
        <mi>b</mi>
      </mrow>
    </mfenced>
    <mn>2</mn>
  </msup>
</mrow>
```

Content markup:

```
<mrow>
  <apply>
    <power/>
    <apply>
      <plus/>
      <ci>a</ci>
      <ci>b</ci>
    </apply>
    <cn>2</cn>
  </apply>
</mrow>
```
For representing bidimensional graphics:

- vector shapes (composed of straight lines and curves),
- raster images,
- text (which remains text, not a set of shapes),
- grouping, transformation, filters, ...


Blue circle with a red border and a black triangle:

```xml
<svg>
  <circle style="fill: blue; stroke: red"
       cx="200" cy="200" r="100"/>
  <path d="M250 150 L150 350 L350 350 Z"/>
</svg>
```

http://www.w3.org/Graphics/SVG/
Dublin Core

Metadata representation standard (ISO 15836).
Maintained by DCMI (Dublin Core Metadata Initiative) – an organization promoting metadata standards and development of specialized vocabularies for resource representation.

http://dublincore.org
The standard defines 15 elements:

- <dc:title>,
- <dc:creator>,
- <dc:subject>,
- <dc:description>,
- <dc:publisher>,
- <dc:contributor>,
- <dc:date>,
- <dc:type>,
- <dc:format>,
- <dc:identifier>,
- <dc:source>,
- <dc:language>,
- <dc:relation>,
- <dc:coverage>,
- <dc:rights>.
Dublin Core — an example

RDF:

<rdf:Description>
  <dc:title>Energy of drinks</dc:title>
  <dc:publisher>Jürgen Lager</dc:publisher>
  <dc:description>Are Full or Empty Beer Bottles Sturdier and Can Their Break the Human Skull?</dc:description>
  <dc:date>2009-06-01</dc:date>
  <dc:language>pl</dc:language>
</rdf:Description>

HTML:

<meta name="DC.title" lang="pl" content="Applied vaccaonomastics">
<meta name="DC.creator" content="John Bull">
<meta name="DC.description" lang="pl" content="Cows with names produce more milk.">

- scope: text documents, spreadsheets, graphs, multimedia presentations,
- history:
  - work started in 2002; first version of the format based on OpenOffice.org package format, currently developed by OASIS,
  - ISO standard from 2006,
  - current version: 1.1 (2007),
  - version 1.2 is now a Committee Draft and will most likely include: additional accessibility features, RDF-based metadata, a spreadsheet formula specification based on OpenFormula, support for digital signatures.
- idea: an open alternative for MS Office formats.
ODF? Is there anyone using it?

Software packages using OpenDocument:

- Google Docs,
- OpenOffice.org,
- KOffice,
- Corel WordPerfect Office X4,
- IBM Lotus Symphony,
- MS Office (starting from Office 2007 SP2),
- ...

Presented as an alternative to formats advertised as open (e.g. Office Open XML), but not created in cooperation with users (including governments).

NATO uses ODF as a mandatory standard for all members, EU also promotes it!

Two methods of document representation:

- in a single XML document (<office:document>),
- in the package containing:
  - content.xml: <office:document-content>,
  - styles.xml: <office:document-styles>,
  - meta.xml: <office:document-meta>,
ODF namespaces (corresponding to tag sets)

- office – general elements, not belonging anywhere else,
- meta – metadata,
- config – application settings,
- text – text fragments, including spreadsheet cell contents,
- table – tabular elements,
- drawing,
- presentation,
- form,
- script,
- style,
- ...
ODF — metadata example

Why do woodpeckers not suffer from headaches?

Philip May

1999-10-19T15:16:17

PT5H10M10S

Woodpecker

Concussion

source

IgNoble 2006
<office:document-content xmlns:office="..." ...
    office:version="1.0">
    <office:automatic-styles>
        <style:style style:name="P" style:family="paragraph">
            <style:paragraph-properties fo:text-align="center"/>
            <style:text-properties style:font-name="Arial"
                fo:font-size="18pt" fo:font-weight="bold"/>
        </style:style>
    </office:automatic-styles>
    <office:body>
        <office:text>
            <text:p text:style-name="P">OpenDocument format is a standard by <text:a xlink:type="simple"
        </office:text>
    </office:body>
</office:document-content>
Office Open XML (OOXML, OpenXML, MSOOXML) is an OpenDocument alternative intended to maintain backward compatibility for Microsoft Office formats and representation of all Office features.

History:

- 2006: ECMA International standard,
- 2007: failed to enter the fast track to ISO standards,
- 2008: anyway, standard has been accepted.

http://www.ecma-international.org/publications/standards/Ecma-376.htm
File structure:

- OOXML document as a package (compressed ZIP),
- inside: XML files with data, metadata, ... according to own markup languages:
  - WordprocessingML (\word\document.xml,\word\styles.xml...),
  - SpreadsheetML,
  - PresentationML,
  - ...
- compatible with binary MS Office formats (able to represent all their features),
- incompatible with Office XP and Office 2003 XML formats.
Typical document structure:

- `<w:document>` – root element,
- `<w:body>` – collection of paragraphs and section properties,
- `<w:p>` – paragraph (a collection of fragments),
- `<w:r>` – run, a text fragment carrying specific properties,
- `<w:t>` – plain text (*text range*).

```xml
<w:document xmlns:w="...">
  <w:body>
    <w:p>
      <w:r>
        <w:t>Hello, world.</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```
Is it really you?
ODF believers:
Your specs is 6000 pages! Can be difficult to implement...

OOXML fanatics:
That’s because there are examples and detailed descriptions. Your ODF has just a tag list!
Anyway, we must be Office-compatible since there are many Office docs out there – million times more than ODF documents...

ODF people:
ODF can represent more than OOXML. More and better, since your specs is very defective. Why 1900 is a leap year?

OOXML people:
ODF is oversimplified as compared to OOXML! You have one type of table for all types of documents...

ODF people:
Noooo, we were working on that since 2002. And we were first! If we have one standard, why not using it? You don't like standards since ODF uses MathML or SVG and you want everything your way! Your language codes are different than ISO. And all that patents and copyright...

OOXML people:
Aaargh!!

Lecture 9: XPointer, XInclude and XML applications

XML and Content Management 31
XLink

HTML links:
- link two documents: link source and target,
- link source is always in the linking element (<A>, <IMG>.

XLink — an extended idea of linking:
- link information represented in any element:
  - element name is not important,
  - attributes coming from XLink namespace are,
- more than two ends of link (hyperlink → relation),
- possibility to represent link outside linked resources.

Status:
- historical roots: HyTime,
- XLink 1.0 – W3C recommendation: 2001,
- XLink 1.1 – current version (made official TR: May 2010).
**Terminology**

**Resources** — any addressable unit of information or a service (file, program, query result).

**Link** – a relation between participating resources, expressed explicitly with a **linking element**.

**Arc** – information about traversal between labelled resources (in defined direction):

- **outbound arc** – from a local resource to some external resource,
- **inbound arc** – from an external resource to some local resource,
- **third party** – between two external resources.

Note: a resource is regarded as **remote** when addressed by URI (even though it resides in the same document or linking element as the link which uses it).
Simple link:
- is outbound,
- binds exactly two resources: a local one with an external one,
- contains exactly one arc between resources.

Extended link:
- binds arbitrary number of local and external resources,
- uses arcs to define methods of traversal between resources,
- defines roles of participating resources,
- defines roles of arcs.
Simple link — an example

<article xmlns:xlink="http://www.w3.org/1999/xlink">
  <author>Diego Amigo</author>
  <title><sponsor xlink:type="simple"
      xlink:href="http://www.example.com/shop.xml#viagra">Viagra</sponsor>
    reduces hamster ’jet lag’</title>
</article>
Extended link — an example

Traversal from any parent to any child:

<family xlink:type="extended"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <person xlink:type="locator" xlink:href="joe.xml"
    xlink:label="parent" xlink:title="Joseph"/>
  <person xlink:type="locator" xlink:href="cathy.xml"
    xlink:label="parent" xlink:title="Katherine"/>
  <person xlink:type="locator" xlink:href="mikey.xml"
    xlink:label="child" xlink:title="Michael"/>
  <person xlink:type="locator" xlink:href="toya.xml"
    xlink:label="child" xlink:title="La Toya"/>
  <person xlink:type="locator" xlink:href="janet.xml"
    xlink:label="child" xlink:title="Janet"/>
  <link xlink:type="arc" xlink:from="parent"
    xlink:to="child"/>
</family>
Attributes in extended links

- **type**: role of the element in a link (simple, extended, locator, arc, resource, title, none)
- **href**: URI of the external resource
- **role**: abstract identifier of the resource role (URI)
- **arcrole**: as above, but for a single arc between resources
- **title**: text label of the resource or arc (optional)
- **show**: presentation info (new, replace, embed, other, none)
- **actuate**: activation info (onLoad, onRequest, other, none)
- **label**: label used as identifier in from and to, not necessarily unique
- **from, to**: pointer (in an arc) for a certain resource label
Simple link as extended

Simple link:

<link xlink:href="...">I. M. Weasel</link>

Extended link:

<link xlink:type="extended">
   <resource xlink:type="resource"
      xlink:label="local">
      I. M. Weasel
   </resource>
   <locator xlink:type="locator" xlink:href="..."
      xlink:label="remote" xlink:role="..."
      xlink:title="...">
   </locator>
   <go xlink:type="arc" xlink:from="local"
      xlink:to="remote" xlink:arcrole="..."
      xlink:show="replace" xlink:actuate="onRequest"/>
</link>
Future of XLink

Applications:
- organization and association of resources even when no writing permission is granted,
- a new type of added value – link sets.

Scope:
- local – link servers, link databases,
- Internet?

Problems:
- visualization of extended links,
- synchronization of links and resources (Internet).
The standard was created in IBM (2001), currently under supervision of OASIS (from 2005).

In short:

- information gathered in **topics** – portions of information on a given subject (title, text, images),
- **ditamaps** – sequences of **topicrefs**, basing on topic metadata,
- **conrefs** – allow to include components into topic contents.

http://www.ditaworld.com
Specialisation: topics can be supplemented hierarchically (according to a specified formalism); the specification introduces 3 „basic” specialisations:

- **task** – procedure describing the method of execution of a given activity, composed of steps,
- **concept** – explanation of processes related to task steps,
- **reference**.

Result:

- specialised documents are of the base type,
- specialised type processing is bound to processing of the base type,
- the processing chain can be extended with new elements.
Database administration is easy.

- If it does not work, restart it.
- If it runs slowly, tell users, to use it economically.