Navigation in XML data; XPath

March 1, 2013

1 XPath

1.1 XPath - core principles

- XPath is a special language (non a XML markup) to specify *parts of* XML documents (nodes, node sets, node sequences);
- Parts of text nodes cannot be specified using XPath.
- XPath uses a syntax resembling the one used for specifying (file) path *in a file system.*
- XPath uses a library of standard functions
- in XPath 2.0 and some XPath 1.x processors may use a user-defined set of functions
- XPath 1.0 is a base for XSLT, XPath 2.0 also for XQuery
- XPath syntax is NOT *XML* (it would be too verbose)
- XPath 1.0 and 2.0 are W3C Recommendations - `http://www.w3.org/TR/xpath`

1.2 XPath - application areas

- advanced navigation in XML data¡?xml version="1.0"?¿ ¡a¿ ¡b/¿ ¡b¿ ¡c/¿ ¡/b¿ ¡b¿ ¡c/¿ ¡/b¿ ¡/a¿

1.3 XPath - application areas

- Select third node b://b[3]
- Select the node b, having an ancestor c://b[./c]
- Select empty node b://b[count(./*)=0]

1.4 XPath - application areas /1

- Transformation (XSLT (`http://www.w3.org/TR/xslt`))
  - used to eg. selection of nodes to process: ¡xsl:apply-templates select="./c"/¿

1.5 XPath - application areas /2

- In "selection part" of XML query languages (XQuery (`http://www.w3.org/XML/Query/`))
- In some modeling languages (Schematron (`http://www.schematron.com/`), XML Schema (`http://www.w3.org/XML/Schema`))
- ...

1.6 XPath - pojem cesty (paths) a lokace (locations)

Path defines (ie. navigates to) a location in a document. Paths are constructed similarly as in a file system, ie.

relative evaluated from the so-called *context node* (CN), see later, or

absolute from the document root, but predicates (expressions) also in relation to the CN

1.7 XPath - syntax

[20] PathExpr ::= AbsolutePathExpr — RelativePathExpr [22] AbsolutePathExpr ::= ("/" RelativePathExpr?) — ("//" RelativePathExpr) [23] RelativePathExpr ::= StepExpr (("/" — "//") StepExpr)* [24] StepExpr ::= AxisStep — GeneralStep [25] AxisStep ::= (Axis? NodeTest StepQualifiers) — AbbreviatedStep

1.8 XPath - osy (axes)

Osy (singular *axis*, plural *axes*) are sets (sequences) of document nodes, usually but not exclusively, outgoing from the *context.*__Context__*is composed of document* and the *current (context) node* (CN).Axes:

child all child nodes of the CN

descendant all descendants of the CN. No attributes.

parent parent node to the CN

ancestor all ancestor (parent, parent of parent, etc.) nodes

following-sibling all following siblings of the CN (for NS node and attributes this is empty)

preceding-sibling dtto, but *preceding* siblings

following all nodes located *after CN* (no attributes, descendants and CN)

preceding similarly, but before

attribute all attributes of the CN (must be an element)

namespace contains all NS nodes of the CN (just for elements)

self the CN itself

descendant-or-self union of `descendant` and `self`

ancestor-or-self union of os `ancestor` and `self`

1.9 Example: child axis

//b/child::*¡?xml version=”1.0”?¿ ¡a¿ ¡b/¿ ¡b¿ **¡c/¿** ¡/b¿ ¡b¿ **¡c/¿** ¡/b¿ ¡/a¿

1.10 Example: descendant axis

//b/descendant::*¡?xml version=”1.0”?¿ ¡a¿ ¡b/¿ ¡b¿ **¡c¿ ¡d/¿ ¡/c¿** ¡/b¿ ¡b¿ **¡c/¿** ¡/b¿ ¡/a¿

1.11 Example: parent axis

¡?xml version=”1.0”?¿ ¡a¿ ¡b/¿ ¡b¿ **¡c¿** ¡d/¿ **¡/c¿** ¡/b¿ ¡b¿ ¡c/¿ ¡/b¿ ¡/a¿

1.12 Example: ancestor axis

¡?xml version=”1.0”?¿ **¡a¿ ¡b/¿ ¡b¿ ¡c¿** ¡d/¿ **¡/c¿ ¡/b¿** ¡b¿ ¡c/¿ ¡/b¿ **¡/a¿**

1.13 Example: axis following-sibling

¡?xml version=”1.0”?¿ ¡a¿ ¡b/¿ **¡b¿** ¡c¿ ¡d/¿ ¡/c¿ **¡/b¿ ¡b¿** ¡c/¿ **¡/b¿** ¡/a¿

1.14 Example: axis preceding-sibling

¡?xml version=”1.0”?¿ ¡a¿ **¡b/¿ ¡b¿** ¡c¿ ¡d/¿ ¡/c¿ **¡/b¿** ¡b¿ ¡c/¿ ¡/b¿ ¡/a¿

1.15 Example: axis following

¡?xml version=”1.0”?¿ ¡a¿ ¡b/¿ ¡b¿ ¡c¿ ¡d/¿ ¡/c¿ **¡e/¿** ¡/b¿ **¡b¿ ¡c/¿ ¡/b¿** ¡/a¿

1.16 Example: axis preceding

¡?xml version=”1.0”?¿ ¡a¿ **¡b/¿ ¡b¿ ¡c¿ ¡d/¿ ¡/c¿ ¡/b¿** ¡b¿ **¡d/¿** ¡e/¿ ¡/b¿ ¡/a¿

1.17 XPath predicates

Condition used to select (filter) nodes specified eg. by pathex.: `/article/para[3]` - selects the third para of the article The simplest is (proximity position) - see above

- Attention by reverse axes (`ancestor`, `preceding`...) - the position is calculated always (outwards) from the CN

3 could equally be replaced by `position()=3`

1.18 XPath expressions

Used in predicates, calculation (aggregation), etc. Might contain XPath functions.Expressions can be:

- string (characters)
- numeric (floating-point numbers)

- logic (boolean)
- nodes
- sequences

1.19 XPath - Examples of shortened notation

- `para`*select all "para" child elements of the CN*
- `*`*selects all element children of the context node*
- `text()`*selects all text node children of the context node*
- `@name`*selects the name attribute of the context node*
- `@*` *selects all the attributes of the context node*
- `para[1]`*selects the first para child of the context node*
- `para[last()]`*selects the last para child of the context node*
- `*/para`*selects all para grandchildren of the context node*
- `/doc/chapter[5]/section[2]`*selects the second section of the fifth chapter of the doc*
- `chapter//para`*elect all "para" descendant elements of "chapter"*
- `//para`*all "para" elements from the document*
- `//olist/item`*all item elements, having a parent "olist"*
- . selects the CN
- `.//para`*select all "para" descendants of the CN*
- `..` *vybere parent element of the CN*
- `../@lang`*selects the attribute "lang" from the parent node of CN*

1.20 XPath - shortened notation (2)

Most frequently used is the shortening of *child axis*:

- like `article/para` instead of `child::article/child::para`.
- and *attributes*: we write `para[@type="warning"]` instead of `child::para[attribute::type="warnin`
- use of `//` instead of `/descendant-or-self::node()/`
- and shorthands dot `.` and double-dot `..`

Sometimes it is good to preserve the full (long) form. So, please, learn it!

1.21 Infosources on XPath

- XPath / W3C: `http://www.w3.org/TR/xpath`
- Zvon XPath Tutorial: `http://zvon.org/xxl/XPathTutorial/Output/index.html`
- XPath Tutorial / W3Schools: `http://www.w3schools.com/xpath/xpath\_intro.asp`

1.22 XPath 2.0

- The Recommendation - `http://www.w3.org/TR/xpath20/`
- The return value of an XPath expression: **all are sequences** (even if one item)
- so they define ORDER on the returned nodes
- **Introduce conditional expressions and loops**
- User functions (in fact, dynamically evaluated expressions in XPath)
- One can use general and existence quantifier, eg. `exist student/name="Fred"` or `all student/@id`
- See further eg. `http://www.saxonica.com/`, where also the XPath/XSLT/XQuery processor *Saxon is located.*

1.23 XPath 2.0 - samples

- String functions (`http://www.fi.muni.cz/~tomp/xml03/xpath20/string.html`)
- Numerical functions (`http://www.fi.muni.cz/~tomp/xml03/xpath20/numeric.html`)
- Sequence functions (`http://www.fi.muni.cz/~tomp/xml03/xpath20/sequence.html`)
- Boolean functions (`http://www.fi.muni.cz/~tomp/xml03/xpath20/boolean.html`)