
Introduction to XPath

Web Engineering, SS 2007

Tomáš Pitner

XPath – Why?

- **Part of the core XML standards family**
 - XML Path Language (XPath) Version 1.0
 - W3C Recommendation 16 November 1999

 - **Mean to navigate in the document tree**
-

Example of an XML Document

```
<?xml version='1.0' encoding='UTF-8'?>
<staff organization="Bundesregierung">
  <person id="agu">
    <name>Alfred Gusenbauer</name>
    <party url="http://www.spoe.at">SPÖ</party>
  </person>
  <person id="wmo">
    <name>Wilhelm Molterer</name>
    <party url="http://www.oevp.at">ÖVP</party>
  </person>
</staff>
```

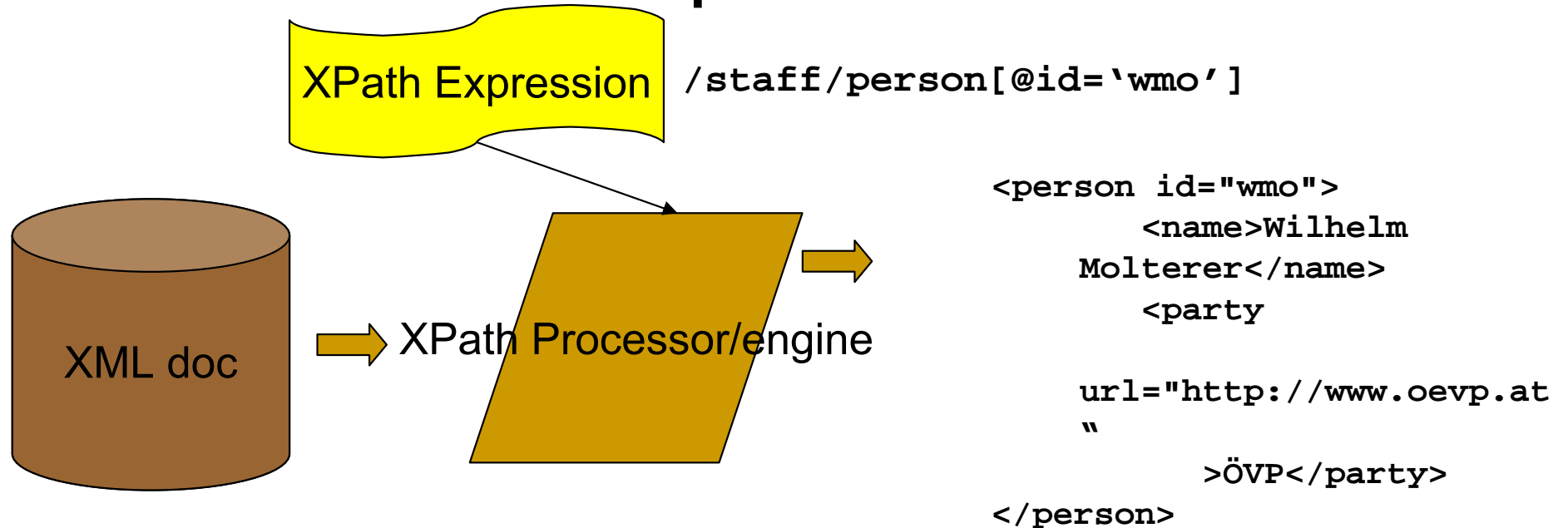
XPath – The Goal

- **Navigation in XML documents**
 - Similar to file addressing in a FS
 - Selects nodes (not just elements) in the tree
 - Further node processing often takes place:
 - XSLT, XQuery – both heavily depend on XPath
 - **Works on Logical Structure (XML tree)**
 - **Example: *an XPath expression against the last document***
 - `/staff/person[@id='wmo']`
-

XPath Processing

■ XPath Processor/Engine

- ❑ Takes XPath expression
- ❑ Takes context node (position in the doc)
- ❑ Accesses the source XML tree
- ❑ Produces node-set/sequence as result



XPath – Anatomy of an Expression

■ XPath Expression

- Consists of **steps** separated by slash /
 - example – two steps
 - `/staff/person`
 - semantics: Select all person elements under staff (which is the root element).
 - resembles *traversing directories* in a FS
 - applicable to both **elements** and **other nodes!**
 - may be **absolute** (starting with /) or **relative**
 - i.e. evaluated without or against context nodes
-

XPath – Predicates

- **XPath Expression with Predicates**

- each step may contain a **predicate**

- It filters the nodes selected at the step

- `/staff/person[@id='wmo']`

- **Select just the person under staff with `id` attribute = 'wmo'**

- **Predicates may be compound**

- `//person[@id='wmo' or @id='agu']`

XPath – Functions

- **To select non-element nodes**
 - **comment()**
 - **text()**
 - **processing-instruction() ...**
 - **To compute a value**
 - **count()**
 - **... much more**
 - **Example – select just the text of the name of the ‘agu’ person:**
 - **`/staff/person[@id='agu']/name/text()`**
-

XPath Processing – Context, Axes

- **Context nodes and Axes**

- **Context nodes**

- “central point” for evaluating relative expressions

- **Axes**

- “directions” leading from the context node out
- node-sets relative to the context node(s)

- **Steps in the expression mean traversing along axes**

XPath Expression Examples

- All against the doc on the first slide... **absolute expressions**
 - **//name** ... selects all name elements
 - **//person[party]** ... selects all person elements having (any) party child element
 - **/staff/person[party[@url='xxx']]**
 - selects all person elements having party with url xxx
 - **/person** ... does not select anything from this doc
-

XPath Expression Examples

- All against the doc on the first slide... **relative expressions**
 - **Context node: first person**
 - **name** ... selects this person's name element
 - **../staff** ... selects the staff element
 - **../*** ... selects also the staff element
 - **/** ... also the staff

 - **./** ... selects all descendant elements
 - **staff/name** ... *does not select anything from this doc*
-

XPath Axes Example (ancestor)

ancestor

ancestor-or-self

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<staff organization="Bundesregierung">
```

```
<person id="agu">
```

```
<name>Alfred Gusenbauer</name>
```

parent

```
<party url="http://www.spoe.at">SPÖ</party>
```

```
</person>
```

```
<person id="wmo">
```

```
<name>Wilhelm Molterer</name>
```

```
<party url="http://www.oevp.at">ÖVP</party>
```

```
</person>
```

```
</staff>
```

context node

XPath Axes Example (descendant)

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<staff organization="Bundesregierung">
```

```
<person id="agu">
```

```
<name>Alfred Gusenbauer</name>
```

```
<party url="http://www.spoe.at">SPÖ</party>
```

```
</person>
```

```
<person id="wmo">
```

```
<name>Wilhelm Molterer</name>
```

```
<party url="http://www.oevp.at">ÖVP</party>
```

```
</person>
```

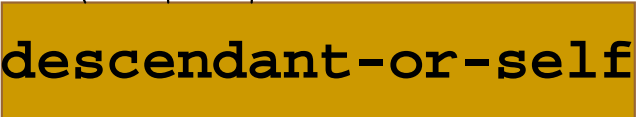
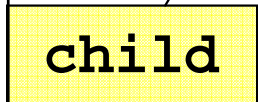
```
</staff>
```

context node

descendant

child

descendant-or-self



XPath Axis Example (attribute)

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<staff organization="Bundesregierung">
```

```
  <person id="agu">
```

```
    <name>Alfred Gusenbauer</name>
```

```
    <party url="http://www.spoe.at">SPÖ</party>
```

```
  </person>
```

```
  <person id="wmo">
```

```
    <name>Wilhelm Molterer</name>
```

```
    <party url="http://www.oevp.at">ÖVP</party>
```

```
  </person>
```

```
</staff>
```

context node

attribute

XPath Axis Example (following-sibling)

context node

```
<?xml version='1.0' encoding='UTF-8'?>
<staff organization="Bundesregierung">
  <person id="agu">
    <name>Alfred Gusenbauer</name>
    <party url="http://www.spoe.at">SPÖ</party>
  </person>
  <person id="wmo">
    <name>Wilhelm Molterer</name>
    <party url="http://www.oevp.at">ÖVP</party>
  </person>
  <person id="abc">
    <name>ABC</name>
    <party url="http://www.abc.at">ABC</party>
  </person>
</staff>
```

following-sibling

XPath Axis Example (preceding-sibling)

```
<?xml version='1.0' encoding='UTF-8'?>
<staff organization="Bundesregierung">
  <person id="agu">
    <name>Alfred Gusenbauer</name>
    <party url="http://www.spoe.at">SPÖ</party>
  </person>
  <person id="wmo">
    <name>Wilhelm Molterer</name>
    <party url="http://www.oevp.at">ÖVP</party>
  </person>
  <person id="abc">
    <name>ABC</name>
    <party url="http://www.abc.at">ABC</party>
  </person>
</staff>
```

preceding-sibling

context node

XPath Axis Example (preceding)

```
<?xml version='1.0' encoding='UTF-8'?>
<staff organization="Bundesregierung">
  <person id="agu">
    <name>Alfred Gusenbauer</name>
    <party url="http://www.spoe.at">SPÖ</party>
  </person>
  <person id="wmo">
    <name>Wilhelm Molterer</name>
    <party url="http://www.oenb.at">ÖVP</party>
  </person>
  <person id="abc">
    <name>ABC</name>
    <party url="http://www.abc.at">ABC</party>
  </person>
</staff>
```

preceding

context node

XPath Axis Example (following)

context node

```
<?xml version='1.0' encoding='UTF-8'?>
<staff organization="Bundesregierung">
  <person id="agu">
    <name>Alfred Gusenbauer</name>
    <party url="http://www.spoe.at">SPÖ</party>
  </person>
  <person id="wmo">
    <name>Wilhelm Molterer</name>
    <party url="http://www.oevp.at">ÖVP</party>
  </person>
  <person id="abc">
    <name>ABC</name>
    <party url="http://www.abc.at">ABC</party>
  </person>
</staff>
```

following

XPath Axis Example (self)

context node

```
<?xml version='1.0' encoding='UTF-8'?>
<staff organization="Bundesregierung">
  <person id="agu">
    <name>Alfred Gusenbauer</name>
    <party url="http://www.spoe.at">SPÖ</party>
  </person>
  <person id="wmo">
    <name>Wilhelm Molterer</name>
    <party url="http://www.oevp.at">ÖVP</party>
  </person>
  <person id="abc">
    <name>ABC</name>
    <party url="http://www.abc.at">ABC</party>
  </person>
</staff>
```

self

XPath Axis Example (namespace)

```
<?xml version='1.0' encoding='UTF-8'?>
<staff organization="Bundesregierung
  xmlns:breg="http://www.austria.gv.at/ns/1.0"
  xmlns="http://generic-ns.org/ns/1.0">
  <person id="agu">
    <name>Alfred Gusenbauer</name>
    <breg:party url="http://www.spoe.at">SPÖ</breg:party>
  </person>
  <person id="wmo">
    <name>Wilhelm Molterer</name>
    <breg:party url="http://www.oevp.at">ÖVP</breg:party>
  </person>
</staff>
```

NS axis for 'party' contains:

<http://generic-ns.org/ns/1.0>

breg = <http://www.austria.gv.at/ns/1.0>

XPath Processing – Summary of axes

- ❑ **ancestor**
 - ❑ **ancestor-or-self**
 - ❑ **attribute**
 - ❑ **child**
 - ❑ **descendant**
 - ❑ **descendant-or-self**
 - ❑ **following**
 - ❑ **following-sibling**
 - ❑ **namespace**
 - ❑ **parent**
 - ❑ **preceding**
 - ❑ **preceding-sibling**
 - ❑ **self**
-

XPath – Long Syntax

- The expression
 - **/staff/person** can be rewritten as
 - **/child::staff/child::person**
 - Similar for other axes
 - there are axes expressible in the long form ONLY
 - Example: *preceding-sibling, ancestor-or-self...*
-

XPath – Resources

- E. R. Harold – **XML in a Nutshell**
 - Chapter on **XPath free** on the web:
<http://www.oreilly.com/catalog/xmlnut/chapter/ch09.html>
 - **XPath (and XSLT) Quick Reference Card**
 - http://www.mulberrytech.com/quickref/XSLT_1quickref-v2.pdf
-