

# Network Services, VU 2.0

## XML Technologies / 2

Dipl.-Ing. Johann Oberleiter  
Institute for Information Systems, Distributed  
Systems Group

# Agenda

- XSLT Best Practices
- XQuery
- XLink & XPointer
- CSS

# XSLT Best Practices

- Source Code generation with XSLT
  - In theory good idea
    - Eg. Transform UML class (in XMI) to Java class
- Problems:
  - Mixing of XML & prog language elements in the same stylesheet
    - Readability
    - Maintainability
    - Debugging (XSLT+Source Code)
  - Only good for small (tiny) programs
  - Better solutions
    - Code transformers (MDA community)
    - Code template engines (eg. Velocity)
      - small set of specific commands

# XSLT Best Practices

- Primary goal of XSLT
  - Generation of XML
  - Generation of simple Text
- Pipelining of XSLT
  - Sometimes its easier to use multiple XSLTs instead of 1 huge stylesheet
    - One stylesheet per task
      - Eg. One adds footer & header, one adds table-layout
    - Easier to debug

# XML Query

- XQuery
- Query language for XML fragments, XML documents, and XML databases
- NOT represented in XML
  - More like a programming language
- Uses XPath expressions to navigate in a document
- Not yet as accepted similar to XPath/XSLT

# XML Query - Sample

```
for $x in doc("studs.xml")/student/nachname  
  order by $x  
return $x
```

Selects all "nachname" elements of all student elements in the "studs.xml" file, and returns the elements sorted alphabetically

# XML Linking Language

- XLink link is a relationship between resources (or portions of resources)
- Simple Link
  - Outbound link with 2 participants
  - HTML A, IMG links
- Extended Link
  - Supports inbound links, third-party arcs
  - Arbitrary number of resources

# XLink - Sample

```
<my:crossRef  
  xlink:href="students.xml"  
  xlink:role=http://www.example.com/studs  
  xlink:title="Student list">Current List of  
  Students</my:crossRef>
```

```
<go xlink:type="arc" xlink:from="A" xlink:to="B"/>  
<go xlink:type="arc" xlink:from="C" xlink:to="B"/>
```



# XLink Application

- XLinkIt (commercial tool)
  - [www.systemwire.com](http://www.systemwire.com)
- Verifies XML document
  - against rules with semantic constraints
  - based on a first-order logic language called CLIX
    - Eg. "No students may have two different valid degrees on the same course"
- Invalid XML elements represented as XLinks
  - Connects Invalid XML element with conflicting rule
  - XLinkIt may also apply repair actions based on these rules

# XPointer

- Supports addressing into internals of a document
  - Combines URLs and XPath
  - Augments XPath expressions
  - XLink may refer to XPointer
- Difference to XPath
  - XPath says nothing about URIs
  - XPointer provides context required in XPath
- Example
  - `abc.xml#xpointer(//students[position()<=2])`

# Cascading Style Sheets - CSS

- Allows attachment of style information to HTML
- Modifies Layout of Input elements
- Nesting / Cascading of stylesheets
- External vs. Internal
- May also be applied to XML files!
  - Eg. Automatic rendering of XML files in tabular form (instead of tree)