Structured Document Processing Languages
Spring 2008
Course Review

Repetitio mater studiorum est!

Goals of the Course

- Learn about central models and languages for
  - manipulating
  - representing
  - transforming and
  - querying
  structured documents (or XML)
- "Generic XML processing technology"

Methodological Goals

- Central professional skills
  - consulting technical specifications
  - experimenting with SW implementations
- Ability to think…?
  - to find out relationships
  - to apply knowledge in new situations
- ("Pidgin English" for scientific communication)

XML?

- Extensible Markup Language is not a markup language!
  - does not fix a tag set nor its semantics
    (like markup languages like HTML do)
- XML is
  - A way to use markup to represent information
  - A metalanguage
    » supports definition of specific markup languages through XML DTDs or Schemas
    » E.g. XHTML a reformulation of HTML using XML

XML Encoding of Structure: Example

```xml
<S>
  <W>Hello</W>
  <E A='1'>world!</E>
</S>
```

Basics of XML DTDs

- A Document Type Declaration provides a grammar (document type definition, DTD) for a class of documents
- Syntax (in the prolog of document instance):
  ```xml
  <!DOCTYPE rootElemType SYSTEM "ex.dtd"
  <!-- "external subset" in file ex.dtd -->
  <! <!-- "internal subset" may come here --> 
  DTD = union of the external and internal subset
  ```

How do Declarations Look Like?

```xml
<!ELEMENT invoice (client, item+)>
<!ATTLIST invoice num NMTOKEN #REQUIRED>
<!ELEMENT client (name, email?)>
<!ATTLIST client num NMTOKEN #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT item (#PCDATA)>
<!ATTLIST item
  price NMTOKEN #REQUIRED
  unit (FIM | EUR) "EUR" >
```

Element type declarations

- The general form is
  ```xml
  <!ELEMENT elementType (E)>
  ```
  where E is a content model
  - regular expression of element names
- Content model operators:
  - E | F: alternation
  - E, F: concatenation
  - E?: optional
  - E*: zero or more
  - E+: one or more
  - (E): grouping
XML Namespaces

```xml
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/TR/xhtml1/strict">
  <!-- XHTML is the 'default namespace' -->
  <xsl:template match="doc/title">
    <h1>
      <xsl:apply-templates />
    </h1>
  </xsl:template>
</xsl:stylesheet>
```

3. XML Processor APIs

- How can applications manipulate structured documents?
  - Overview of document parser interfaces
    - 3.1 SAX: an event-based interface
    - 3.2 DOM: an object-based interface
    - 3.3 JAXP: Java API for XML Processing

A SAX-based application

```
Application Main Routine
    startDocument()
    startElement("a")
    characters("Hi!")
    endElement()
```

DOM: What is it?

- Object-based, language-neutral API for XML and HTML documents
  - Allows programs/scripts to
    » build
    » navigate and
    » modify documents
- "Directly Obtainable in Memory" vs "Serial Access XML"

Overview of XSLT Transformation

JAXP (Java API for XML Processing)

- An interface for "plugging-in" and using XML processors in Java applications
  - Includes packages
    > org.xml.sax: SAX 2.0 interface
    > org.w3c.dom: DOM Level 2 interface
    > javax.xml.parsers: initialization and use of parsers
    > javax.xml.transform: initialization and use of transformers (XSLT processors)
- Included in standard Java

JAXP: Using a SAX parser (1)
4. Introduction to Style Sheets

- Specify and produce visual representation for structured documents
- by defining a mapping from document structure+content to formatting tasks, and
  - inserting/generating new text
  - numbering
  - rearranging
- by rules based on contextual conditions

Process of Transformation (muunnos)

Structured document

Transformer

Transformation

Style sheet
- LaTeX style file, CSS, XSLT

Formatter input

TaX, FOT
(XSL formatting object tree)

Process of Formatting (muotoilu)

- Creates a detailed description of presentation

Formatter input
- TaX
- FOT
(XSL formatting object tree)

Formatter

Descr. of presentation

DVI, PS, PDF

CSS - Cascading Style Sheets

- A stylesheet language
  - mainly to specify the representation of web pages by attaching style (fonts, colours, margins, ...) to HTML/XML documents
- Example style rule:

```css
H1 {color: blue; font-weight: bold;}
```
XSL: Transformation & Formatting

Page regions

- A simple page can contain 1-5 regions, specified by child elements of the simple-page-master

<table>
<thead>
<tr>
<th>Region-start</th>
<th>Region-body</th>
<th>Region-end</th>
</tr>
</thead>
</table>

Top-level formatting objects

- Slightly simplified:
  - for:root
  - fo:layout-master-set
  - fo:page-sequence+:
    - (fo:simple-page-master | fo:page-sequence-master)+
  - fo:flow
  - fo:region-
    - body
    - before?
    - after?
  - specifiy masters for page sequences, by referring to simple-page-masters

XQuery in a Nutshell

- Functional expression language
  - A query is a side-effect-free expression
  - Operates on sequences of items
    - XML nodes or atomic values
  - Strongly-typed: (XML Schema) types may be assigned to expressions statically, and results can be validated
  - Extends XPath 2.0 (but not all axes required)
    - common for XQuery 1.0 and XPath 2.0:
      - Functions and Operators, W3C Rec. 01/2007
  - Roughly: XQuery = XPath 2.0 + XSLT' + SQL'

FLWOR ("flower") Expressions

- for, let, where, order by and return clauses (~SQL select-from-where)
- Form: (ForClause | LetClause)+
  - WhereClause?
  - OrderByClause?
  - 'return' Expr

  - binds variables to values, and uses these bindings to construct a result
  - (an ordered sequence of nodes)

XQuery Example

```
for $pn in distinct-values(doc("sp.xml")//pno)
let $sp:=doc("sp.xml")//sp_tuple[pno=$pn]
where count($sp) >= 3
order by $pn
return <well_supplied_item><pno>{$pn}</pno>
      <avgprice>{avg($sp/price)}</avgprice><well_supplied_item>
```

Course Main Message

- XML is a universal way to represent information as tree-like data structures
- Specialized and powerful technologies for processing it
  - hype has settled
  - R&D still active