Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

January 22, 2018

XML and Databases
— Exercise Sheet 11 —

You only have to submit the parts marked as "Homework Exercises", i.e. Part c). But please think about the questions in Part a) before the meeting! Send your homework solutions to the instructor via EMail: `brass@informatik.uni-halle.de` (with "`xml17`" in the subject line). The official deadline is January 25, 10:00 (before the lecture time).

Repetition Questions

a) Answer the following questions about XSLT:

- One application of XSLT is to translate XML data files to HTML for display in a browser. The transformation can be done in the browser (client) or on the server. What are the advantages/disadvantages of the two alternatives?
- What else can be done with XSLT besides the translation from XML to HTML?
- Explain the structure of a template. What is the most important attribute of `xsl:template`? Which `xsl`-elements are commonly used inside `xsl:template`?
- If the attribute `match` of a template has the value `A/B`, does the template match an `A`-element with `B`-child, or a `B`-element with `A`-parent?
- If several templates match a node in the input XDM tree, which template will be executed? How can one influence the template selection?
- What is the purpose of the built-in templates of XSLT?
- If a template contains `<xsl:apply-templates select="A">` and this template is applied to node n, how is the corresponding result tree fragment determined?
- If a template contains `<xsl:value-of select="A">` and this template is applied to node n, how is the corresponding result tree fragment determined?
- If a template is applied to node n, this is the context node at the beginning. How can you access this node in XPath-expressions if the context node is changed (by navigating in the input tree)?
- If you want to generate HTML, can you write simply `<br>` for a line break, or do you have to write `<br />`?
- How can one link from an XML-file to a stylesheet?

- What is the purpose of `xsl:output`?
- Is it possible that a stylesheet does not terminate? If yes, give an example.
- Name some differences between XPath 1.0 and XPath 2.0.
- Compare XSLT with XQuery. How are new nodes constructed? How can one sort?
- How can one write case distinctions in XSLT? For instance, suppose you want to print the number of tracks on a CD. If the CD has only one track, you want to print "1 track" (singular), otherwise "n tracks" (plural).

In-Class Exercises

b) Consider again the employee-department database (derived from a relational database used by Oracle in SQL tutorials):

- [http://users.informatik.uni-halle.de/~brass/xml17/empdept.xml]
- [http://users.informatik.uni-halle.de/~brass/xml17/empdept.dtd]

The DTD is:

```
<!ELEMENT EMPDEPT (DEPT*, SALGRADE*)>

<!ELEMENT DEPT (EMP*)>
<!ATTLIST DEPT  DEPTNO   NMTOKEN #REQUIRED
                DNAME    CDATA   #REQUIRED
                LOC      CDATA   #REQUIRED>

<!ELEMENT EMP EMPTY>
<!ATTLIST EMP   EMPNO    NMTOKEN #REQUIRED
                ENAME    CDATA   #REQUIRED
                JOB      CDATA   #REQUIRED
                MGR      NMTOKEN #IMPLIED
                HIREDATE CDATA   #REQUIRED
                SAL      NMTOKEN #REQUIRED
                COMM     NMTOKEN #IMPLIED>

<!ELEMENT SALGRADE EMPTY>
<!ATTLIST SALGRADE
                GRADE    NMTOKEN #REQUIRED
                LOSAL    NMTOKEN #REQUIRED
                HISAL    NMTOKEN #REQUIRED>
```

Please write the following stylesheets:

- Create a headline (`h2`) for each non-empty department, followed by an unordered list (`ul`) of all employees of the department with name, job and salary. E.g. the section for the “Accounting” department should look as follows:

```
<h2>ACCOUNTING</h2>
<ul>
<li>KING (PRESIDENT): 5000</li>
<li>CLARK (MANAGER): 2450</li>
<li>MILLER (CLERK): 1300</li>
</ul>
```

- Create for each employee that is a manager of at least one employee an HTML table of all subordinate employees with name, job and salary. (The attribute `MGR` of an employee contains the `EMPNO` of his/her manager.)

Homework Exercises

Consider again the XML file for the classical music CDs:

- Data file:
  [http://www.informatik.uni-halle.de/~brass/xml17/cd.xml]
- XML Schema definition:
  [http://www.informatik.uni-halle.de/~brass/xml17/cd.xsd]
- DTD:
  [http://www.informatik.uni-halle.de/~brass/xml17/cd.dtd]

The document has the following elements:

- `CDDB`: `(composers, cds?, soloists?)`
- `composers`: `(composer*)`
- `composer`: `(pieceOfMusic*)`, attributes: `cno`, `firstName`, `name`, `born`, `died`.
- `pieceOfMusic`: `(recording*)`, attributes: `pno`, `title`, `key`, `opus`.
- `recording`: empty content, attributes: `rno`, `orchestra`, `conductor`.
- `cds`: `(cd*)`.
- `cd`: `(track*)`, attributes: `cdno`, `name`, `producer`, `numDiscs`, `totalTime`.
- `track`: empty content, attribute: `rno`.
- `soloists`: `(soloist*)`.
- `soloist`: `(performance*)`, attribute: `name`.
- `performance`: empty content, attributes: `rno`, `instrument`.

c) Please write an XSLT-stylesheet for translating the `CDDB` XML file to HTML. It should generate an “unordered list” of CDs, where each entry contains:

- the name of the CD,
- after that in brackets “`[...]`” the number of tracks on the CD,
- and then a nested ordered list containing the titles of the music pieces of which there are recordings on the CD.

The output should look as follows (the order can be arbitrary):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
          "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>CD Database</title>
    </head>
    <body>
        <h1>CD Liste</h1>
        <ul>
            <li>Händel: Feuerwerksmusik [4]
                <ol>
                    <li>Concerto grosso op.6 Nr.7</li>
                    <li>Feuerwerks-Musik (Concerto grosso)</li>
                    <li>Concerto grosso op.6 Nr.8</li>
                    <li>Sinfonia e-moll (aus ‘Der Messias’)</li>
                </ol>
            </li>
            ...
        </ul>
    </body>
</html>
```

Note: The function `current()` returns the current node, i.e. the node to which the template is applied. At first, this is also the context node, therefore one does not need this functions for simple one-step XPath expressions. However, if you write a more complex XPath expression, the context node changes (e.g., after “`/`” or within “`[...]`”), whereas `current()` still returns the node to which the template is applied. This can be useful e.g. when you want to navigate from a `track` to the corresponding `pieceOfMusic`.