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

November 24, 2017

XML and Databases
— Exercise Sheet 6 —

You only have to submit the parts marked as "Homework Exercises", i.e. Part e). But please think about the questions in Part a) and b) 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 November 30, 10:00 (before the lecture time).

Repetition Questions

a) Answer the following questions about XML schema:

- Suppose that there are several instances of a key that is referenced with a foreign key. How is the matching done in this case? What key values can be referenced?
- What possibilities are there to derive
  - a simple type from a simple type?
  - a complex type from a simple type?
  - a complex type from a complex type?

  Give an example for each case.
- Compare derived types in XML schema with subclasses in Java.
- Suppose you develop an application program that processes data given as XML files. You defined an XML schema for the data files. If other programmers use your schema with `include`, and extend it, what should you expect in your program? The extended schemas cannot redefine your types and elements, but they can add new types and elements, and it particular make use of derived types.

b) Answer the following questions about XDM (the XQuery and XPath Data Model):

- The main data structure used in XPath 2.0 and XQuery is the "sequence". How is this notion defined?
- What are the seven kinds of nodes?
- Why is an extra "document node" needed as the root, and not simply the root element?
- Name some syntactical details that are present in the textual representation of an XML data file, but are lost in the XDM representation (because they are considered "unessential", and not part of the main information content).

- Why is there an InfoSet standard and the XDM standard? Both describe the information content of an XML file.
- What differences are there between an XDM instance that is constructed without validation and an XDM instance that is constructed from a "Post Validation InfoSet"?
- How is white space (spaces, tabs, line breaks) treated between elements?
- Why are namespace nodes "deprecated"? What is the problem?
- Define the "document order" for nodes in an XDM instance.
- Which relations are there between the following types?
  - "`anyType`"
  - "`anySimpleType`"
  - "`anyAtomicType`"
  - "`untypedAtomic`"

  Explain the type hierarchy and give an example for a value that belongs to each supertype, but not to the subtype.
- What is the "string value" of element nodes? And what is the "typed value" if the document was validated? You have to distinguish between elements with simple content, elements with pure element content, and elements with a "mixed content model". (Part of this question will only be treated in the next lecture, so don't worry if you cannot answer it now.)

In-Class Exercises

c) Please download the following files:

- [http://users.informatik.uni-halle.de/~brass/xml17/xsl/ex1_query.xml]
- [http://users.informatik.uni-halle.de/~brass/xml17/xsl/ex2_query.xml]
- [http://users.informatik.uni-halle.de/~brass/xml17/xsl/query.xsl]

You can also open the directory for downloading the files:

[http://users.informatik.uni-halle.de/~brass/xml17/xsl/]

The purpose of the stylesheet is to show information about the XDM nodes in the XML files. If you open the XML files in a web browser, you should see an HTML list of the nodes with types, name, and possibly string value. Compare this with the XML source code in the downloaded files. (Actually, you could see the source code also in the browser with "View Source": `Ctrl+U` in Firefox.)

d) Experiment with the XPath query expression in in the file `query.xsl`. A minimal subset of XPath was introduced in the XML Schema chapter for keys and foreign keys. Write e.g. different XPath expressions to select the last names of all students.

Homework Exercises

e) Please learn yourself how XML can be processed in Java programs with the DOM interface, e.g. with the following tutorial:

[https://docs.oracle.com/javase/tutorial/jaxp/dom/index.html]

Further tutorials are:

- [https://www.tutorialspoint.com/java_xml/java_dom_parser.htm]
- [https://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/]

The DOM specification was developed by the W3C:

- [https://www.w3.org/TR/DOM-Level-3-Core/]
- [https://www.w3.org/DOM/]

DOM is also used for manipulating HTML in JavaScript programs, but for XML, we need only the DOM core.

Now the homework is to write a small Java program that can read the homework grades database and prints the SIDs of all students that have 10 points for Homework 1. This task could be solved with a short XPath expression, but it is only a simple example for accessing parsed XML data in Java.

You can choose between the version with the data in attributes as in Figure 1

[http://users.informatik.uni-halle.de/~brass/xml17/ex1.xml]

and the version with the data in nested elements as in Figure 2.

[http://users.informatik.uni-halle.de/~brass/xml17/ex2.xml]

Note that you might have to explain your program in class. Please be sure you understood the purpose of every statement in your program. If your submitted program is very similar to a program you found in the internet, you must name the source.

Of course, you must test that your program compiles and executes without errors.


```
<?xml version='1.0' encoding='ISO-8859-1'?>
<GRADES-DB>
<STUDENT SID='101' FIRST='Ann' LAST="Smith"
        EMAIL='smith@acm.org'/>
    <STUDENT SID='102' FIRST='Michael' LAST='Jones'/>
    ...
    <EXERCISE CAT='H' ENO='1' TOPIC='Relational Algebra' MAXPT='10'/>
    ...
    <RESULT SID='101' CAT='H' ENO='1' POINTS='10'/>
    <RESULT SID='101' CAT='H' ENO='2' POINTS= '8'/>
    ...
</GRADES-DB>
```

Fig. 1: Grades Database with Data in Attributes (`ex1.xml`)

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<GRADES-DB>
    <STUDENTS>
        <STUDENT>
            <SID>101</SID>
            <FIRST>Ann</FIRST>
            <LAST>Smith</LAST>
            <EMAIL>smith@acm.org</EMAIL>
        </STUDENT>
        ...
    </STUDENTS>
    <EXERCISES>
        <EXERCISE>
            <CAT>H</CAT>
            ...
        </EXERCISE>
        ...
    </EXERCISES>
    <RESULTS>
        <RESULT>
            <SID>101</SID>
            <CAT>H</CAT>
            <ENO>1</ENO>
            <POINTS>10</POINTS>
        </RESULT>
        ...
    </RESULTS>
</GRADES-DB>
```

Fig. 2: Grades Database with Data in Elements (`ex2.xml`)