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

November 10, 2017

# XML and Databases
# — Exercise Sheet 4 —

You only have to submit the parts marked as "Homework Exercises", i.e. Part f), g) and h). 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 November 16, 10:00 (before the lecture time).

Not all submitted homeworks will be corrected, but all homework exercises will be discussed in class. If you should have questions about your homework, please ask! A precondition for getting credit for this course is that you submit solutions to two thirds of the homework sheets. Obviously wrong or very incomplete submissions do not count.

If you should miss the tutorial session (Thursday, 12-14), it is recommended that you solve the "in-class exercises" yourself.

## Repetition Questions

a) Answer the following questions about XML schema:

- What does the regular expression "`[ab]*|\d{3}.`" mean?
- Why is it problemantic to use the `pattern`-facet for numeric types?
- Name some numeric types that are predefined in XML schema.
- How does one define a type corresponding to `NUMERIC(5,2)` in XML schema? What is the base type and which facets must be used?
- How does one define an enumeration type in XML schema?
- What is the format for a `dateTime` value in XML schema? How do you write a value for "`November 10, 2017, 9:00 am CET`" (Central European Time: "MEZ", one hour later than UTC)?
- Is a list type a simple type or a complex type in XML schema? Suppose you defined a list of integers as a type. How would a lexical representation look like?
- Give an example where a union-type makes sense.
- What does `xs:all` mean in the content model in XML schema? Name some restrictions for `xs:all` compared with `xs:sequence` and `xs:choice` (or explain positively what can be done with `xs:all`)?



- What is a named model group (`xs:group`)? Why can it be useful? What is the advantage of named model groups when compared with parameter entities in DTDs?
- Why is a global attribute declaration often problematic? What can one do if attributes with the same name and type are used for several different elements?

## In-Class Exercises

b) Define a simple type "`length`" for length measurements with unit. The values of this type should consist of a sequence of digits followed by one of the suffixes "`mm`", "`cm`" and "`m`". Leading zeros should not be allowed, but the value "`0`" with any of the units should be possible. Write a minimal schema and XML document to try your definition.

c) Define a simple type for results of exams that permits values from 0 to 100 percent, plus the value "NE" ("nicht erschienen": the student did not show up to the exam).

d) Define an enumeration type for colors of fireworks articles that has at least the following values: "`gold`", "`silver`", "`red`", "`green`", "`blue`", "`yellow`". Then define a list type for such colors, because articles can also have a mixture of colors, e.g. blue and gold is a nice combination.

e) Suppose that product IDs in your company (for display fireworks) consist of the following components:

- First a capital letter (that encodes the basic article type),
- then a two-digit number,
- then optionally a dot "`.`" followed by 1–3 digits (for caliber variants of the same effect, e.g. `.75` for a 75mm shell and `.100` for the same effect as a 100mm shell),
- finally one of the following letters or letter combinations that encodes color variants: `G` (gold), `S` (silver), `R` (red), `GR` (green), `B` (blue), `Y` (yellow). Only one of these colors can be specified (no combinations as in the previous exercise).

Define a regular expression that describes such product IDs.

## Homework Exercise

f) I want to store benchmark results as listed on the following web page:

[http://users.informatik.uni-halle.de/~brass/push/bench.html]

The requirements are as follows:

- There is a list of systems, each with a short name/ID (e.g. "`XSB`"), an optional long name (e.g. "`XSB 3.6 (Gazpatcho)`"), and an optional web address (e.g. "[`http://xsb.sourceforge.net/`]"). Note that the short name is a valid XML identifier without "`:`".
- There is also a list of benchmarks, each with an identifier (e.g. "`Join1`"), a long name ("`Join-1 Benchmark with query a(X,Y)`"), and one or more versions of program code (e.g. there can be a Datalog version and an SQL version).
- Each version of a benchmark has a name (e.g. "`SQL version`") and a possibly longer text (e.g. the SQL query). Of course, it must be clear to which benchmark the version belongs. If you want, you can assign an ID to a benchmark version (in addition to the name that is a printable text, possibly with spaces).
- Each performance measurement (result of the benchmark) needs the system on which the benchmark was executed, the exact version of the benchmark, the date when the benchmark was run, and the runtime in seconds (with three decimal places, i.e. milliseconds). For simplicity, we store only one total runtime (not load and execute times and the memory requirement as on the above webpage). We also assume that all benchmarks are run on the same machine (that does not need to be described).

Develop an XML data file with at least two systems, and two benchmarks, where one benchmark has two versions. Store at least three performance results in total.

g) Develop a DTD for this application, such that the XML data file from f) can be validated with respect to your DTD.

h) Develop an XML schema for this application, such that the XML data file from f) can be validated with respect to your schema.