

---

Introduction to Databases, Fall 2005  
IT University of Copenhagen

**Lecture 8, part II: XML for data exchange**

October 31, 2005

Lecturer: Rasmus Pagh

---

# — Today's lecture, part II —

---

## XML for data exchange

- What is XML?
- Defining XML formats using XML schemas.

## — Motivation: Integration of databases —

- In many businesses, data from a large number of heterogeneous databases need to be integrated
  - in connection with data warehousing, or
  - in connection with system integration in general
- This is no easy task, due to differences in formats, conventions, systems, assumptions, etc.

This part of today's lecture gives an overview of (one of the things that may go into) transferring data from one system to another.

# XML

---

XML is a *standardized textual notation for so-called “semistructured” data.*

It is (primarily) aimed at semistructured data which is a **tree**. More specifically, an XML tree is a structure consisting of:

- *Leaves* that contain data (in the form of strings)
- *Internal nodes*, each having a number of leaves and/or other internal nodes as *children* (no cycles are allowed).

In XML notation, an internal node with label `1` is represented as `<1>...</1>`. The dots ... abbreviate the XML for the children of the node.

# XML example

---

```
<?xml version="1.0">
<root>
  <star>
    <name>Carrie Fisher</name>
    <address><street>Maple</street><city>H'wood</city></address>
    <address><street>Locust</street><city>Malibu</city></address>
  </star>
  <star>
    <name>Mark Hamill</name>
    <address><street>Oak</street><city>B'wood</city></address>
  </star>
  <movie>
    <title>Star Wars</title>
    <year>1977</year>
  </movie>
</root>
```

## — XML and WWW —

---

Besides data interchange, another use of XML in connection with databases is for sharing information via the World Wide Web.

- Newer web browsers have special facilities for viewing XML documents (e.g., containing the result of a database query).
- There are specialized languages such as XSLT that can be used to specify how XML data is to be presented in a browser (converting it to HTML).

## — Problem session (5 minutes) —

---

Suggest a way of representing the below relation in XML.

<i>accountNo</i>	<i>balance</i>	<i>type</i>
12345	1000.00	savings
67890	2846.92	checking
32178	-3210.00	loan

Does your approach work for arbitrary relations?

What about entire relational databases?

---

**Next: Defining XML formats using XML schemas.**

---

## — Schemas for XML —

---

When doing data interchange it is necessary to have a *common description* of the data format, i.e., we need a specification of what data is allowable.

Several languages for writing such **schemas** for XML are used. The most widespread are **DTD** (old and well-established) and **XML Schema** (new, more powerful standard).

These schema languages work by specifying a **grammar** for the XML documents allowed, i.e., a set of rules that can be used to form any allowable XML document.

Essentially, for each `<1>...</1>` (called an XML **element**) it is specified what can occur between `<1>` and `</1>`.

# DTDs

---

**DTD by example:** [Figure 4.22 shown on slide]

DTDs have these ways of stating what can be in an element:

- Text, written as #PCDATA.
- A sequence of elements, written (ELEM1, ELEM2, ELEM3, ...)
- Zero or more occurrences of the same element, written ELEM\*.
- A choice between elements, written ELEM1 | ELEM2 | ELEM3 | ...
- An optional element, written ELEM?

It is possible to combine the above, and write expressions such as:

((A? | (B|C)\*), D)

# XML Schemas

---

“XML Schema” seems to be the upcoming standard for XML schemas.

Some main features, relative to DTDs:

- Sophisticated type system (in contrast to #PCDATA).
- Large schema definitions can be split into modules.
- Can specify “no content”.
- Supports a mechanism for distinguishing different XML elements with the same name (“namespaces”).

[Murray, Figure 3, shown on slide]

## — Why XML? —

---

The reason for the success of XML and XML Schema is, in fact, *not* that it does something that could not be done before.

The good new thing is **standardization**:

- There is *widespread agreement* that the XML standards will form the basis of information interchange in the future.
- Consequently, the major players in the software industry, many country administrations, etc., support XML.
- There are many tools available for XML and related technologies.

## — Most important points in this lecture —

As a minimum, you should after this week:

- Be able to recognize an XML document.
- Be able to understand a simple DTD or XML Schema.

## — Next lecture —

---

Next time we will look at the basics of relational database efficiency:

- Indexes.
- Update versus query performance.
- Database tuning.