Databasesystemer, forår 2005
IT Universitetet i København

Forelæsning 12, del 1: XML i databaser

28. april 2005

Forelæser: Rasmus Pagh

Today's lecture, part I

XML for data exchange

- Semistructured data and XML.
- Defining XML formats using XML schemas.
- (Web) services and service oriented architectures (SOA).

XML in a relational databases

- XML data types.
- Ad-hoc information.

Next: XML for data exchange.

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
- Communication among databases is no easy task, due to differences in formats, conventions, systems, etc.

Also, modern service oriented architectures (SOA) consist of many components (services) that interact through communication protocols that transmit data.

The semistructured data model

The **semistructured data model** is a *flexible* way of describing data.

The flexibility makes it a good data model for data exchange:

- It is (mostly) easy to convert a given data set to a semistructured representation.
- It is (often) easy to perform transformation from one semistructured representation to another.

Semistructured data

Semistructured data can be represented as a **graph** with **nodes**, and **arcs** with labels between the nodes.

There are three kinds of nodes:

- A single **root node**, with no arcs entering, represents the entire database.
- Leaf nodes, with no arcs leaving, have associated data (e.g. strings).
- Interior nodes have arcs entering and leaving, but no data.

XML

XML is a *standardized textual notation for semistructured data*.

It is (primarily) aimed at semistructured data which is a **tree**, i.e., where all nodes (except the root) have exactly one arc entering.

- An arc in the tree with label `l` pointing at a node n in the semistructured data is represented in the XML document as a pair of **tags**:

  `<l>...</l>`

  Here `...` is the XML description of the part of the semistructured data for which n is the root.

- Leaf nodes are represented by the data they contain.

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).

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 `<l>...</l>` (called an XML **element**) it is specified what can occur between `<l>` and `</l>`.

XML and service architectures

Service oriented architectures (SOA) is a current trend in IT systems.

XML is the basis in an on-going standardization of:

- Definition language for (web) *services* (WSDL).
- "Yellow pages" for (web) services (UDDI).

These, or similar, industry standards are likely to play a major role in future systems (web based and within organizations).

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 (cf. EDI, SGML).

The good new thing is **standardization**:

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

Next: XML in relational databases.

XML data types

The major DBMSs now offer XML as a native data type, similar to strings, integers, etc.

XML data can be processed using (a subset of) the XQuery language.

XQuery generalizes in many ways SQL. However, in general one should not expect as good performance as delivered by SQL.

Ad-hoc information

In many databases, the need to store new kinds of information arises continually.

Adding new attributes to a relation may be a bad idea if the new kind of information is only relevant for few rows (many NULLS).

A possible solution is to store such ad-hoc information in an XML attribute. This is particularly attractive if the new data is not used in searches (i.e., does not need to be indexed).