Storing and Querying XML data; XQuery

March 22, 2013

1 XQuery

1.1 Characteristics

- Query language for searching and extraction of XML nodes (elements, attributes) from a document and for an output XML document construction.

1.2 Characteristics (2nd)

- The XQuery is a basic XML query language at present time (and it seems in the future as well).
- The W3C specification since March 2011, see `http://www.w3.org/XML/Query`.
- Base on XPath 2.0 data model, operators and functions.
- Supported by main database engines producers (IBM, MS, Oracle, etc)

1.3 Where to use XQuery (and where not)

XQuery domain are:

- queries, where extraction (selection) part is more complicated than the construction part.
- Use the XSLT in the opposite case
- or using the more general API (using DOM manipulation for example).

1.4 Source code example

Example of source document, XML Queries on it and their results.

```
<?xml version="1.0" encoding="Windows-1250"?>
 <addressbook>
  <person category="friends">
   <firstname>Petr</firstname>
   <lastname>Novák</lastname>
   <date-of-birth>1969-05-14</date-of-birth>
```

```
          <email>novak@myfriends.com</email>
          <characteristics lang="en">Very good friend</characteristics>
         </person>
         <person category="friends">
          <firstname>Jaroslav</firstname>
          <lastname>Nováèek</lastname>
          <date-of-birth>1968-06-14</date-of-birth>
          <email>novacek@myfriends.com</email>
          <characteristics lang="en">Another good friend</characteristics>
         </person>
         <person category="staff">
          <firstname>Jan</firstname>
          <lastname>Horák</lastname>
          <date-of-birth>1970-02-01</date-of-birth>
          <email>horak@mycompany.com</email>
          <characteristics lang="en">Just colleague</characteristics>
         </person>
         <person category="friends">
          <firstname>Erich</firstname>
          <lastname>Polák</lastname>
          <date-of-birth>1980-02-28</date-of-birth>
          <email>erich@myfriends.com</email>
          <characteristics lang="en">Good friend</characteristics>
         </person>
</addressbook>
```

1.5 Example - Simple Query XPath

Figure: XQuery to the source documents. Task: ”extract all surnames in the addressbook”.Query is XPath expression - selects all `lastname` elements.

```
doc(’myaddresses.xml’)/addressbook/person/lastname
```

1.6 Figure - Running XQuery using Saxon 9.0j

XSLT processor Saxon contains the XQuery processor since version 8.x as well.
To process XQuery you need:

- to install Saxon 9.0.0.4j for example (”j” means implementation in java, there is a .NET implementation as well) by unpacking to folder `c:/devel/saxon9-0-0-4j` for example.
- Change working directory to the folder: cd `c:/devel/saxon9-0-0-4j`
- put the above mentioned query into a file (`lastnames.xq`).
- store the above mentioned XML document containing ”addressbook” into the file `myaddresses.xml` in the same directory.
- Run: java -classpath saxon9.jar net.sf.saxon.Query -o result.xml lastnames.xq from command line.

1.7 Figure - result

The query to above mentioned document will create the file `result.xml` its content follows:

```
<lastname>Novák</lastname>
<lastname>Nováček</lastname>
<lastname>Horák</lastname>
<lastname>Polák</lastname>
```

2 XQuery structure

2.1 FLWOR

FLWOR is an acronym of an XQuery structure.It means:

(F)or Initial query part that specifies query cycle including control variable. Results of XPath expression behind the keyword ”in” are assigned to the variable.

(L)et You can assign values of next variable that can be used later in this section.

(W)here specifies selection condition ie. which nodes (values) selected by for section will be used.The condition can utilize the variables defined in the ”let” section.

(O)rder Defines how the nodes should be oredered.

(R)eturn Defines what is returned, constructed from extracted nodes (values).

2.2 FLWOR - simple example

Condition used to select requested nodes can be specified either in an XPath expression in ”for” clause or in the ”where” clause.”Return Mr. Polaks birthdate.”

```
for $person in
        doc('myaddresses.xml')/addressbook/person where
 $person/lastname='Polák' return $person/date-of-birth
```

Query returns:¡?xml version=”1.0” encoding=”UTF-8”?¿ ¡date-of-birth¿1980-02-28¡/date-of-birth¿

3 XQuery Implementation

3.1 SAXON since versions 7.x

- install (extract) Saxon with version 7.0 at least (8.x, 9.x) into some directory

- change working directory to the Saxon directory and
- run: java -classpath saxon9.jar net.sf.saxon.Query -o result.xml queryfile.xq from command line.
- There is a .NET Saxon implementation (means .DLL and .EXE files)

3.2 Native XML databases

Native XML database systems support XQuery as a query language often.Native XML Databases are for example:

- Berkeley DB XML (`http://www.sleepycat.com/products/index.shtml`)
- eXist (`http://exist.sourceforge.net/`)