
Querying XML Documents

Paul Cotton, Microsoft

Jonathan Robie, Software AG

Unicode Conference

Jan 30, 2002

Organization of Presentation

- ◆ XML query history
- ◆ XML Query WG history, goals and status
- ◆ XML Query working drafts
- ◆ XQuery overview
- ◆ XQuery issues
- ◆ Questions

XML query history

- ◆ Early queries facilities for SGML
- ◆ 1998: “roll your own query language”
- ◆ Feb 1998: XQL proposal
 - <http://metalab.unc.edu/xql>
- ◆ Aug 1998: XML-QL submission
 - <http://www.w3.org/TR/NOTE-xml-ql/>
- ◆ Dec 1998: W3C QL’98 Workshop
 - <http://www.w3.org/TandS/QL/QL98>
- ◆ Nov 1999: XPath Recommendation
 - <http://www.w3.org/TR/xpath>

QL'98 candidate requirements

- ◆ QL'98 workshop summary
 - Candidate Requirements for XML Query, Paul Cotton and Ashok Malhotra, IBM
 - <http://www.w3.org/TandS/QL/QL98/pp/queryreq.html>
- ◆ See also:
 - Database Desiderata for an XML Query Language, David Maier, Oregon Graduate Institute
 - <http://www.w3.org/TandS/QL/QL98/pp/maier.html>

W3C XML Query WG - History

- ◆ July 1999: Working Group proposed as part of XML Activity Phase 3 rechartering
- ◆ Sept 1999: WG chartered and first F2F
- ◆ Currently 27 W3C member companies
- ◆ 15 F2F meetings and 88+ telcons so far
- ◆ Public WDs every three months
- ◆ Proposed recommendation(s)

W3C XML Query WG - Goals

- ◆ “The goal of the XML Query WG is to produce a data model for XML documents, a set of query operators on that data model, and a query language based on these query operators.”

W3C XML Query WG - Status

- ◆ Jan 2000: Requirements Working Draft
- ◆ May 2000: XML Query Data Model WD
- ◆ May 2000: Feedback on Schema Last Call
- ◆ Aug 2000: Revised Requirements Working Draft with Use Cases
- ◆ Dec 2000: XML Query Algebra WD
- ◆ Feb 2001: Revised Working Drafts
 - XML Query Requirements
<http://www.w3.org/TR/xmlquery-req>

W3C XML Query WG - Status

- ◆ June 2001: Revised Working Drafts
 - XQuery 1.0: An XML Query Language
 - XML Query Use Cases
 - XML Query 1.0 and XPath 2.0 Data Model
 - XML Query 1.0 Formal Semantics
 - XML Syntax for XQuery 1.0: XQueryX
- ◆ August 2001
 - XML Query 1.0 and XPath 2.0 Functions and Operators
<http://www.w3.org/TR/query-operators>

W3C XML Query WG - Status

- ◆ December 2001
 - XQuery 1.0: An XML Query Language
<http://www.w3.org/TR/xquery>
 - XML Query Use Cases
<http://www.w3.org/TR/xmlquery-use-cases>
 - XML Path Language (XPath) 2.0
<http://www.w3.org/TR/xpath20>
 - XML Query 1.0 and XPath 2.0 Data Model
<http://www.w3.org/TR/query-datamodel/>
 - XML Query 1.0 and XPath 2.0 Functions and Operators
<http://www.w3.org/TR/query-operators>
- ◆ Next publication status
- ◆ WG Charter status

XML Query Requirements WD

◆ General Requirements

- Non-procedural query language
- XML syntax for query language but also a readable syntax
- Protocol independent
- Standard error conditions
- Future support for updates

◆ XML Query Data Model Requirements

- Built on XML Infoset and PSVI
- Namespace aware
- Support for XML Schema data types
- Support for inter- and intra- document references

XML Query Requirements WD

◆ XML Query Functionality

- Operators on all data types
- Text operators across element boundaries
- Support for hierarchy and sequence
- Ability to combine data from different locs
- Aggregation and sorting
- Combination of operators including queries as operands
- Support for NULL/empty values
- Structural preservations
- Identity preservation
- Operations on names
- Operations on “schemas”
- Extensibility
- Closure

XML Query Use Cases WD

- ◆ Use Case Organization
 - Description, DTD/Schema, Input Data, Queries and Results
- ◆ Current Use Cases
 - "XMP": Experiences and Exemplars
 - "TREE": Queries that preserve hierarchy
 - "SEQ" - Queries based on Sequence
 - "R" - Access to Relational Data
 - "TEXT": Full-text Search
 - "NS" - Queries Using Namespaces
 - "PARTS" - Recursive Parts Explosion
 - "REF" - Queries based on References

XML Query 1.0 Data Model WD

- ◆ Defines what information is available to an XML Query or XPath 2.0 processor
- ◆ Published jointly with XSL Working Group
- ◆ Infoset plus the following:
 - Support for XML Schema data types (PSVI)
 - Support for document collections
 - Support for references
- ◆ Node-labelled tree constructor model with node identity
- ◆ Mapping from Infoset to Query Data Model uses Infoset terminology and shown by example

XML Query 1.0 Formal Semantics WD

- ◆ XML Query Formal Semantics is used:
 - to define XQuery semantics
 - to support query optimization
- ◆ FS defines both static and dynamic semantics
 - static semantics are presented as type inference rules, which relate XQuery/FS expressions to types
 - dynamic, or operational, semantics are presented as value inference rules, which relate XQuery/FS expressions to values in the XML Query Data Model

XQuery: A Query Language for XML

- ◆ XQuery is a functional language in which a query is represented as an expression
- ◆ XQuery expressions can be nested with full generality
- ◆ The input and output of an XQuery are instances of the XML Query Data Model
- ◆ Based on OQL, SQL, XML-QL, XPath
- ◆ Readable vs. XML syntax

XQueryX

- ◆ XQueryX is an XML representation of an XQuery
- ◆ It was created by mapping the productions of the XQuery abstract syntax directly into XML productions
- ◆ XQueryX useful to enable:
 - Parser reuse
 - Queries on queries
 - Generation of queries
 - Embedding of queries in XML documents

XQuery Expressions

- ◆ XQuery expressions
 - Path expressions
 - Element constructors
 - FLWR expressions
 - Expressions involving operators and functions
 - Conditional expressions
 - Quantified expressions
 - List constructors
 - Expressions that test or modify datatypes

XQuery Path Expressions

- ◆ Based on abbreviated syntax of XPath 1.0
- ◆ **(Q1) In the second chapter of the document named "zoo.xml", find the figure(s) with caption "Tree Frogs".**

```
document("zoo.xml")/chapter[2]//figure[caption =  
"Tree Frogs"]
```

- ◆ Extended with:
 - a new dereference operator
 - a range predicate
- ◆ **(Q3) Find captions of figures that are referenced by <figref> elements in the chapter of "zoo.xml" with title "Frogs".**

```
document("zoo.xml")/chapter[title = "Frogs"]  
//figref/@refid->fig/caption
```

XQuery Element Constructors

- ◆ XQuery element constructor consists of a start tag and an end tag, enclosing an optional list of expressions that provide the content of the element.
- ◆ **(Q8) Generate an <emp>element that has an “empid” attribute. The value of attribute and the content of the are specified by variables that are bound in other parts of the query.**

```
<emp empid = {$id}>
  {$name}
  {$job}
</emp>
```

XQuery FLWR Expressions

- ◆ A FLWR expression binds some expressions, applies a predicate, and constructs a new result.



FOR and LET clauses generate a list of tuples of bound expressions, preserving document order.

WHERE clause applies a predicate, eliminating some of the tuples

RETURN clause is executed for each surviving tuple, generating an ordered list of outputs

XQuery FLWR Expressions

- ◆ **(Q11) List the titles of books published by Morgan Kaufmann in 1998.**

```
FOR $b IN document("bib.xml")//book
WHERE $b/publisher = "Morgan Kaufmann"
      AND $b/year = "1998"
RETURN $b/title
```

- ◆ **(Q12) List each publisher and the average price of its books.**

```
FOR $p IN distinct(document("bib.xml")//publisher)
LET $a := avg(document("bib.xml")
              /book[publisher = $p]/price)
RETURN
  <publisher>
    <name> {$p/text()} </name>
    <avgprice> {$a} </avgprice>
  </publisher>
```

XQuery Operators and Functions

- ◆ Infix and prefix operators
- ◆ Parenthesized expressions
- ◆ Arithmetic and logical operators
- ◆ Sequence operators UNION, INTERSECT and EXCEPT
- ◆ Functions can be defined in XQuery

XQuery Operators and Functions

- ◆ **(Q25) Find the maximum depth of the document named "partlist.xml."**

```
NAMESPACE xsd="http://www.w3.org/2001/XMLSchema-datatypes"
```

```
FUNCTION depth(ELEMENT $e) RETURNS xsd:integer
```

```
{
```

```
  -- An empty element has depth 1
```

```
  -- Otherwise, add 1 to max depth of children
```

```
  IF empty($e/*) THEN 1
```

```
  ELSE max(depth($e/*)) + 1
```

```
}
```

```
depth(document("partlist.xml"))
```

XQuery Conditional Expressions

- ◆ IF THEN ELSE construct
- ◆ **(Q21) Make a list of holdings, ordered by title. For journals, include the editor, and for all other holdings, include the author.**

```
FOR $h IN //holding
RETURN
  <holding>
    {$h/title,
     IF $h/@type = "Journal" THEN
       $h/editor
     ELSE
       $h/author
    }
  </holding> SORTBY (title)
```


XQuery Quantified Expressions

- ◆ Existential and Universal quantifiers
- ◆ **(Q22) Find titles of books in which both sailing and windsurfing are mentioned in the same paragraph.**

```
FOR $b IN //book
WHERE SOME $p IN $b//para SATISFIES
    contains($p, "sailing")
    AND contains($p, "windsurfing")
RETURN $b/title
```

- ◆ **(Q23) Find titles of books in which sailing is mentioned in every paragraph.**

```
FOR $b IN //book
WHERE EVERY $p IN $b//para SATISFIES
    contains($p, "sailing")
RETURN $b/title
```

Sequence-related Operators

- ◆ A sequence may be constructed by enclosing zero or more expressions separated by commas.
- ◆ For example: ($\$x$, $\$y$, $\$z$) denotes a sequence containing three members represented by variables
- ◆ PRECEDES and FOLLOWS boolean functions
- ◆ () denotes an empty sequence.

XQuery Operators on Data Types

- ◆ INSTANCEOF returns True if its first operand is an instance of the type named in its second operand
- ◆ CAST is used to convert a value from one data type to another
- ◆ TREAT causes the query processor to treat an expression as though its data type were a subtype of its static type

XQuery Issues

- ◆ Alignment of XQuery/XPath
- ◆ Revised version of Formal Semantics and XQueryX
- ◆ Update language – now or later?
- ◆ Internationalization (I18N) issues
- ◆ Support for full-text retrieval

Internationalization Issues

- ◆ Internationalization issues
 - string operations
 - comparison and sorting of data
 - specification of collations
 - default collation (user, query or schema?)
 - relationship xml:lang

Full-Text Issues

- ◆ Full-Text issues

- history within WG
- Library of Congress Use Case
http://lcweb.loc.gov/crsinfo/xml/lc_usecases.html
- related to I18N issues
- portability versus interoperability
- cross language definition of characters, words, sentences and paragraphs
- relationship to SQL/MM Part 2: Full-Text

ftp://sqlstandards.org/SC32/WG4/Progression_Documents/CD/cd-fulltext-2001-05.pdf

Early implementations

- ◆ CL-XML http://homepage.mac.com/james_anderson/XML/index.html
- ◆ Enosys Markets <http://www.enosysmarkets.com/products/xq.html>
- ◆ Fatdog <http://www.fatdog.com/>
- ◆ Kawa-Query <http://www.gnu.org/software/kawa/xquery/>
- ◆ IPSI-XQ <http://xml.ipsi.fhg.de/xquerydemo>
- ◆ Lucent <http://db.bell-labs.com/galax/>
- ◆ Kweelt <http://db.cis.upenn.edu/Kweelt/>
- ◆ Microsoft <http://xqueryservices.com>
- ◆ Software AG <http://www.softwareag.com/developer/downloads/default.htm>
- ◆ SourceForge <http://sourceforge.net/projects/xquench/>
- ◆ X-Hive <http://www.x-hive.com/xquery>
- ◆ XML Global <http://www.xmlglobal.com>
- ◆ and more ...

Questions

- ◆ Today
- ◆ Later:
 - pcotton@microsoft.com
 - jonathan.robie@softwareag.com
- ◆ Feedback email list:
 - www-xml-query-comments@w3.org
- ◆ Public email list:
 - www-ql@w3.org