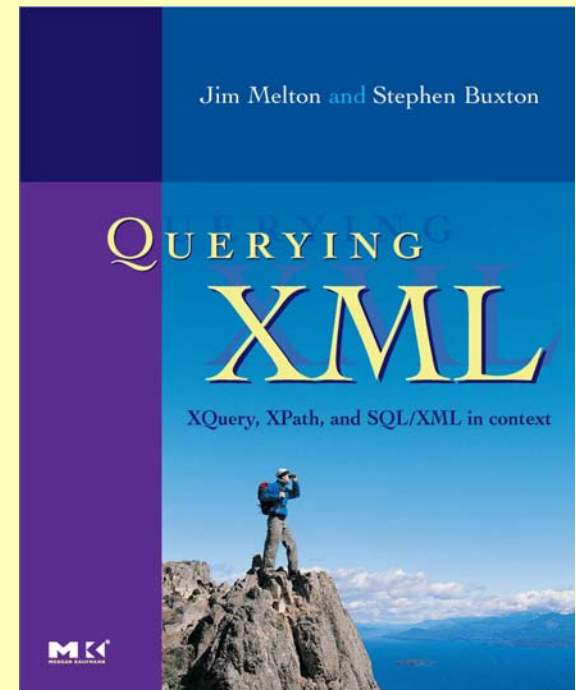


# XQuery Update

## An Update

# Your Humble Presenter

- Editor, all part of SQL standard, > 20 years
- Co-Chair W3C XML Query WG
- [Co-]Editor of 1/2-dozen XQuery-related docs
- Author: five SQL books, one XPath/XQuery book
- *SV Dream SeQueL*
- *Sheltie Rescue of Utah*



# History of XQuery Development

- 1998-12: Query Language '98 Workshop
- 1999-09: XML Query WG established
- 2000-11: Quilt chosen as starting point (syntax)
- 2001-08: Publication of first public WD
- 2003-11, 2005-04: Last Calls 1, 2
- 2005-11: Candidate Recommendation
- 2006-11: Proposed Recommendation
- 2007-01: Recommendation

# XQuery Update Timeline

- 2002-10: First proposed to XML Query WG
- By Chamberlin, Florescu, Lehti, Melton, Robie, Rys, Siméon
- 2006-01: FPWD
- 2007-08: LC
- 2008-03: CR
- PR, REC?

# What is XQuery Update?

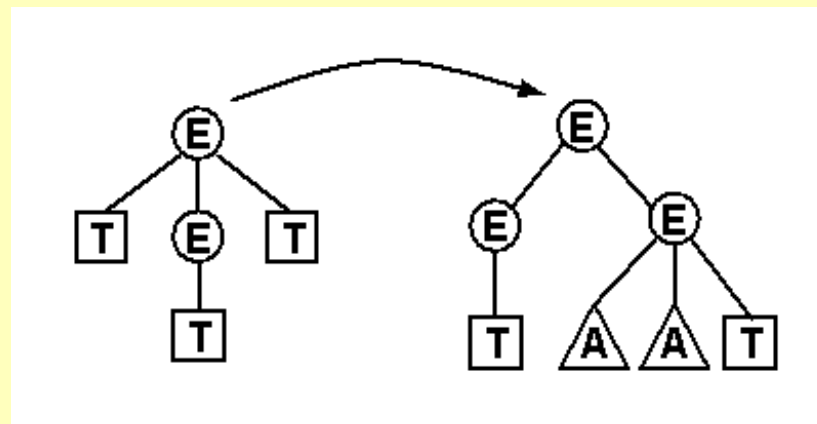
- Extension to XQuery 1.0
- Extension of processing model
- Syntax, semantics, formal semantics, XML syntax
- “Functional language with side effects”

# Outline of the spec

- Extensions to processing model, prolog, static context
- New expressions
- Extensions to existing expressions, built-in functions
- Update primitives, internal routines

# A Vital Conceptual Decision

- Update expressions make changes *only* to XDM instances (“no angle brackets”)
- Effect, if any, on backing store (persistent data) is implementation-defined



# Two Kinds of Expressions

- Expressions that return a value: XQuery 1.0 expressions (“simple expressions”)
- Expressions that return a pending update list: XQuery Update 1.0 expressions
- Implications of that design decision (debated)
  - Heads: Simpler language, easier optimization, no important use cases impossible
  - Tails: Actually simpler to allow both value & PUL, no optimization problems, important use cases require it

# Processing Model Extensions

- Values *vs* Pending Update Lists
  - Simple expressions, vacuous expressions
  - Updating expressions, basic updating expressions
- Definitional method
  - Update primitives – *e.g.*, upd:insertBefore(), upd:delete(), upd:replaceValue(), upd:rename() – contents of PULs
  - Update routines – *e.g.*, upd:mergeUpdates(), upd:applyUpdates(), upd:revalidate(), upd:removeType()
- Snapshot semantics: entire query = snapshot; terminated by upd:applyUpdates()

# Prolog Extensions

- Revalidation mode

```
declare revalidation ( strict | lax | skip )
```

- Updating operations cause loss of type information on updated nodes

```
<part num="12"><name>robot</name></part>
```

```
➔ insert node <qty>25</qty> into //part
```

```
➔ <part num="partno"><qty>25</qty>  
    <name>robot</name></part>
```

*...but what is its XML schema type?*

- Revalidation discovers (new) type...or not

# New Kinds of Expressions

- Insert
- Delete
- Replace
- Rename
- Transform
- *Note: not all of these are updating expressions 😊*

# Insert Expression

- **insert node *source location?* target**
- ***source* and *target* are simple expressions**
- ***location* is one of:**
  - **into or as first into or as last into**
  - **after**
  - **before**
- ***target* must be a single element node if any form of **into ...** is specified, and must be single element, text, comment, or PI node if **after** or **before** is specified**

# Insert Expression Examples

- `insert node <user_tuple>  
<userid>U07</userid><name>Annabel  
Lee</name></user_tuple> into  
doc( "users.xml" ) / users`
- `insert nodes <comment>This is a  
bargain !</comment> as last into  
doc( "items.xml" ) / items / item_tuple  
[ itemno=1002 ]`

# Delete Expression

- **delete node *target***
- *target* is a simple expression
- Causes deleted node(s) to be “disconnected” from its/their parent(s) at end of snapshot

# Delete Expression Examples

- `delete nodes`  
`doc("part-tree.xml")//part`
- `for $pt in`  
`doc("part.xml")//parts[@name="car"]`  
`//part,`  
`$pl in doc("part-list.xml")//part`  
`where $pt/@partid eq $pl/@partid`  
`return`  
`delete nodes $pl`

# Replace Expression

- **replace node *target* with *source***
- *target* and *source* are simple expressions
- *target* must be single element, attribute, text, comment, or PI node...that has a parent
- Element, text, comment, or PI nodes can be replaced only by zero or more of those
- Attribute nodes can be replaced only by zero or more attribute nodes

# Replace Value Of Expression

- replace value of node *target* with *source*
- *target* and *source* are simple expressions
- Retains node identity of *target*
- *target* must be single element, attribute, text, comment, or PI node
- *source* must evaluate to single text node (or empty sequence)
- Content of *target* node is replaced by that text node

# Replace Expression Examples

- **replace node**  
`fn:doc("bib.xml")/books/book`  
`[1]/publisher` with  
`fn:doc("bib.xml")/books/book`  
`[2]/publisher`
- **let \$bp :=**  
`fn:doc("bib.xml")//book[1]/price`  
**return** replace value of node `$bp`  
with `$bp/value() * 1.1`

# Rename Expression

- **rename node *target* as *newname***
- ***target*** and ***newname*** are simple expressions
- ***target*** must be a single element, attribute, or PI node
- ***newname*** must evaluate to a QName
- Effects limited to ***target***; descendants and attributes not affected

# Rename Expression Example

- ```
for $node in $root//abc:*
let $localName := fn:local-name($node),
$newQName :=
  fn:concat("xyz:", $localName)
return ( rename node $node as
  fn:QName("http://xyz/ns", $newQName),
  for $attr in $node/@abc:*
  let $attrLocalName :=
    fn:local-name($attr),
    $attrNewQName :=
      fn:concat("xyz:", $attrLocalName)
  return rename node $attr as
  fn:QName("http://xyz/ns", $attrNewQName)
```

# Transform Expression (not an updating expression)

- **copy** *\$varname* := *source* ...  
    **modify** *update-expr* return *result*
  - *\$varname* is name of variable holding *copy*
  - *source* is simple expression, must be single node
  - *update-expr* is an updating expression (or empty sequence, or call to error() function)
  - *result* is a simple expression, result of transform
- New copy made of *source* (with no type), bound to *\$varname*, *update-expr* is applied to copy, revalidation performed, *result* is result of transform
- Does *not* preserve node id of *source*

# Transform Expression Examples

- `for $e in  
 //employee[skill = "Java"]  
return  
 copy $je := $e modify  
 delete node $je/salary  
 return $je`
- `let $oldx := /a/b/x  
return copy $newx := $oldx  
 modify (rename node $newx as "newx",  
 replace value of node $newx by  
 $newx * 2)  
return ($oldx, $newx)`

# Compatibility of Update Expressions

- If any node is affected by more than one **rename** expression within a snapshot: dynamic error
- If any node is affected by more than one **replace** expression within a snapshot: dynamic error
- Within a given snapshot, if an element node **E** is the target of a **replace value of** expression, and the children of **E** are also modified by other expressions, the final children of **E** are determined by the **replace value of** expression

# Extensions to Existing Expressions

- To account for updating expressions used within XQuery 1.0 expressions
  - FLWOR
  - Typeswitch
  - Conditional
  - Comma
  - Parenthesized
  - Function Declaration (new type: updating function)
  - Function Call

# Extensions to Function Library

- `fn:put($node as node(),  
$uri as xs:string)  
as empty-sequence()`
- Semantics are implementation-defined, but presumed to “store a document or element to the location specified by \$uri”
- Results not visible until after completion of containing query

# Conformance

- Minimal conformance to XQuery 1.0
- Support for everything in XQuery Update 1.0 *except* optional feature
  - Update Facility Static Typing Feature
- Conformance to the update facility static typing feature requires conformance to XQuery 1.0 static typing

# Test Suite

- In development at:  
<http://dev.w3.org/2007/xquery-update-10-test-suite/>
- Please contribute tests and/or run the test suite on your XQuery Update implementations (you *do* have one, don't you?)

# Future

- XQuery Scripting Extension 1.0
- Merge into XQuery 1.1?
- Relationship with Full Text?

# The Documents

- Requirements:  
<http://www.w3.org/TR/xquery-update-10-requirements/>
- Use Cases:  
<http://www.w3.org/TR/xquery-update-10-use-cases/>
- The spec:  
<http://www.w3.org/TR/xquery-update-10/>

# Questions?