

# XQuery zet zijn eerste schuchtere stappen

## XQuery groeit slechts heel langzaam uit tot een volwaardige query taal voor XML. Toch wordt het nu al in een aantal XML databank producten uitgetoetst.

Hans C. Arents

*De XML query taal XQuery is na méér dan 2 jaar werk nog steeds niet vastgelegd in een definitieve standaard. Erger nog, zelfs in de meest recente werkversie van de standaard wordt nog altijd niets gezegd over mogelijke insert/update/delete functionaliteiten, toch wel cruciaal om XQuery te laten uitgroeien tot een volwaardige data manipulatie taal. Wenst u toch al wat ervaring met XQuery op te doen, dan kan u wel bij een aantal XML databank producten terecht waarin reeds succesvol een proefversie van XQuery werd geïmplementeerd.*

### Hoever staan we met XQuery?

In een vorig artikel "Voorproefje XQuery stemt gematigd optimistisch" (zie Database Magazine, nummer 2, april 2001) hebben we besproken hoe het W3C (**World Wide Web Consortium**), de non-profit organisatie die instaat voor het vastleggen van alle cruciale Web standaarden, er eindelijk in geslaagd was om een query data model en een query algebra voor XQuery te definiëren, en begonnen was met het vastleggen van de syntax voor XQuery. Bijna een jaar later, op 20 december 2001, is er een nieuwe werkversie van de XQuery standaard verschenen. Deze werkversie verschilt niet wezenlijk van de vorige werkversie van de XQuery standaard, op één belangrijk aspect na: XQuery maakt nu gebruik van XPath 2.0. XPath 2.0 is de nieuwste versie van XPath, de zoeksyntax die toelaat van bepaalde knoppen in een XML boomstructuur terug te vinden en te selecteren. XPath 2.0 speelt ook een sleutelrol in XSLT, de XML stylesheet taal, waarvan tegelijkertijd met de nieuwe werkversie van XQuery een eerste werkdocument voor de nieuwste versie 2.0 verschenen is. De XML Query Working Group en de XSL Working Group werken momenteel dan ook nauw samen om er voor te zorgen dat hun inspanningen op een gelijkaardige manier gebruik maken van de XPath 2.0 syntax. De proliferatie aan nieuwe, goedgekeurde of nog in ontwikkeling zijnde XML standaarden het afgelopen jaar heeft er immers voor gezorgd dat er momenteel een acute nood bestaat aan coördinatie en samenwerking tussen de verschillende XML werkgroepen van het W3C.

De nieuwe werkversie van XQuery heeft echter al heel wat kritiek losgemaakt. Waar XPath vroeger een vrij eenvoudige syntax en semantiek had, evolueert XPath 2.0 meer en meer in de richting van een echte programmeertaal, wat de taal heel wat moeilijker om aan te leren en te gebruiken maakt. De voornaamste kritiek is echter gericht op de XQuery taal zelf, omdat deze nog steeds niet de nodige functionaliteiten bevat om XML elementen aan te maken, aan te passen of te verwijderen. Het W3C verdedigt zich tegen deze kritiek door er op te wijzen dat voor haar de prioriteit nu ligt bij het zo snel mogelijk vastleggen van een stabiele XQuery 1.0 standaard. De gevraagde insert/update/delete functionaliteiten zouden pas aan bod kunnen komen in een versie 2.0 van XQuery.

### Hoe XQuery updates er zouden kunnen uitzien

Op de recente XML 2001 conferentie in Orlando heeft Jonathan Robie, een van de grondleggers van de huidige versie van XQuery, toch al een eerste voorstel gedaan ("Updates in XQuery") met betrekking tot de mogelijke implementatie van insert/update/delete functionaliteiten in XQuery. Om u een idee te geven van hoe XQuery updates er zouden kunnen uitzien volgen hier een aantal (eenvoudige) voorbeelden. In deze voorbeelden gaan we er van uit dat u reeds vertrouwd bent met XQuery en de XPath zoeksyntax die door XQuery gebruikt wordt (zie het vorig artikel). In de voorbeelden worden de volgende XPath symbolen gebruikt:

<b>Symbool</b>	<b>Betekenis</b>
.	De inhoud van de knop zelf
/	De wortel knop, of de separator tussen de verschillende stappen in het zoekpad
//	De afstammelingen van de knop (op een willekeurige diepte in de boomstructuur)
[...]	Een Booleanse expressie om te filteren in de resultaatknoppen van een bepaalde stap

In de voorbeelden werken we net zoals in het vorige artikel met een XML document met als naam "bib.xml" dat een lijst van <boek> elementen bevat. Ieder <boek> element bevat op zijn beurt een <titel> element, één of meerdere <auteur> elementen, een <uitgever> element, een <jaar> element, en een <prijs> element. Een grafische voorstelling van het document model (in XML Schema syntax) voor dit "bib.xml" document vindt u in figuur 1.

### ***Toevoegen (insert)***

De INSERT instructie laat toe van een element toe te voegen vóór of na een bestaand element.

Voorbeeld: voeg een nieuw <auteur> element toe dat de waarde "An Janssens" heeft, in alle boeken die een <auteur> element bevatten met de waarde "Jan Janssens". Deze nieuwe auteur An Janssens dient toegevoegd te worden vóór de bestaande auteur Jan Janssens.

```
UPDATE
INSERT <auteur>An Janssens</auteur>
PRECEDING //auteur[.="Jan Janssens"]
```

Voorbeeld: voeg een nieuw <auteur> element toe dat de waarde "An Janssens" heeft, in alle boeken die een <auteur> element bevatten met de waarde "Jan Janssens". Deze nieuwe auteur An Janssens dient toegevoegd te worden na de bestaande auteur Jan Janssens.

```
UPDATE
INSERT <auteur>An Janssens</auteur>
FOLLOWING //auteur[.="Jan Janssens"]
```

De INSERT instructie dient niet enkel om een element toe te voegen vóór of na een bestaand element, maar laat ook toe van een attribuut toe te voegen aan een bestaand element.

Voorbeeld: voeg een aanspreektitel attribuut toe dat de waarde "Dr." heeft, in alle <auteur> elementen met de waarde "Jan Janssens".

```
UPDATE
INSERT ATTRIBUTE aanspreektitel { "Dr." }
INTO //auteur[.="Jan Janssens"]
```

De INSERT INTO instructie laat ook toe van een element toe te voegen in een ander element. Dit element wordt dan toegevoegd aan het einde van de lijst met kind elementen van het geselecteerde element.

Voorbeeld: voeg een nieuw <prijs> element toe dat de waarde "16" heeft, in alle boeken die een <auteur> element bevatten met de waarde "Jan Janssens". Dit nieuw <prijs> element wordt toegevoegd als het laatste element van de kind elementen van het <boek> element.

```
UPDATE
INSERT <prijs>16</prijs>
INTO //boek[auteur="Jan Janssens"]
```

### ***Aanpassen (update)***

De REPLACE instructie laat toe van een element in zijn geheel te vervangen.

Voorbeeld: vervang in alle boeken die een <auteur> element bevatten met de waarde "Jan Janssens" dit <auteur> element door een nieuw <auteur> element met de waarde "An Janssens".

```
UPDATE
REPLACE //auteur[.="Jan Janssens"]
WITH <auteur>An Janssens</auteur>
```

### ***Verwijderen (delete)***

De DELETE instructie laat toe van een element te verwijderen.

Voorbeeld: verwijder alle boeken die een <auteur> element bevatten met de waarde "Jan Janssens".

```
UPDATE
DELETE //boek[auteur="Jan Janssens"]
```

### **Gebruik binnen een FLWR expressie**

De INSERT/REPLACE/DELETE instructies kunnen gecombineerd worden binnen een XQuery FLWR expressie (een expressie opgebouwd aan de hand van FOR, LET, WHERE en RETURN clauses), zodat een aantal toevoegen/aanpassen/verwijderen instructies samen kunnen uitgevoerd worden.

Voorbeeld: loop doorheen al de boeken, kijk of het boek een <auteur> element bevat met de waarde "Jan Janssens", en als dit het geval is, voeg een nieuw <auteur> element met de waarde "An Janssens" toe vóór dit element, en vervang het <auteur> element met de waarde "Jan Janssens" door een <auteur> element met de waarde "Jan Pieters".

```
UPDATE
FOR $b IN document("bib.xml")//boek
LET $a := $b/auteur[.="Jan Janssens"]
WHERE not (empty($a))
INSERT <auteur>An Janssens</auteur>
PRECEDING $a
REPLACE $a
WITH <auteur>Jan Pieters</auteur>
```

### **Producten die reeds XQuery ondersteunen**

Bovenstaande voorbeelden geven u slechts een eerste voersmaakje van hoe XQuery updates er kunnen gaan uitzien. Wil u liever zelf uitproberen hoe XQuery in de praktijk aanvoelt, dan heeft u nu al de keuze tussen een aantal XML databank producten waarvoor een proof-of-concept XQuery processor geïmplementeerd werd. De bekendste is momenteel QuiP (<http://www.softwareag.com/developer/quip/>) van Software AG, de makers van de goed verkopende Tamino XML databank. QuiP is een prototype Windows visuele XQuery builder, die u toelaat om XQuery queries te formuleren en de resultaten van die queries meteen te bekijken. De software komt met 76 voorbeeld queries, en een aantal XML bestanden waarop u die queries kan loslaten. Als u reeds Tamino gebruikt kan u vanuit QuiP ook uw XML databank aanspreken. De ondersteuning voor de huidige versie van de XQuery standaard is vrij volledig, en de meest recente versie van QuiP ondersteunt ook al het "Updates in XML" voorstel.

Een andere uitstekende prototype XQuery processor is de XQuery demo (<http://www.x-hive.com/xquery>) van X-Hive Corporation, het Nederlandse bedrijf bekend van X-Hive/DB, momenteel één van de meest gesofisticeerde native XML databank producten beschikbaar op de markt. De X-Hive XQuery demo draait rechtstreeks bovenop de X-Hive/DB databank, maar is wel enkel via een on-line web interface toegankelijk. De demo ondersteunt het merendeel van de XQuery Use Cases uit de XQuery standaard, en u kan deze queries op voorbeeld XML documenten in de databank uitproberen. U kan ook uw eigen queries formuleren en uitproberen. De X-Hive XQuery demo ondersteunt nog geen updates, maar ondersteunt wel al XQuery queries die over meerdere documenten heen lopen, zoals vastgelegd in de experimentele XMach-1 XML databank benchmarking specificatie.

Ook Microsoft heeft een gelijkaardige XQuery demo (<http://131.107.228.20/xquerydemo/>) waar u via een on-line web interface enkele voorbeeld XQuery queries kan uitproberen. U vindt hier ook al voorbeelden van XQuery updates, maar wel in een syntax die lichtjes verschilt van de voorgestelde syntax. Microsoft heeft al laten uitschijnen dat het in de volgende versie van zijn SQL Server database (codenaam Yukon, voorzien voor 2003) native XML ondersteuning zal aanbieden, met inbegrip van het gebruik van XQuery als query taal. Daarnaast is ook IBM druk bezig met de ontwikkeling van een native XML databank (codenaam Xperanto), die zal dienstdoen als een onderdeel van DB2. Een eerste pre-beta versie van Xperanto zal in de eerste helft van dit jaar publiek gemaakt worden via de IBM developer web site (<http://www.ibm.com/developerworks/>). Tenslotte zijn er nog een aantal andere commerciële XML databanken zoals GoXML (<http://www.xmlglobal.com/prod/db/>) van XML Global of Virtuoso Server (<http://demo.openlinksw.com:8891/xquery/demo.vsp>) van Openlink die XQuery beginnen te ondersteunen, weliswaar elk in een nét iets verschillende, lichtjes van de voorlopige standaard afwijkende vorm.

Speelt u liever met Open Source software, dan heeft u hier ook al de keuze uit een aantal XQuery implementaties. Zo vindt u bij SourceForge XQuench (<http://sourceforge.net/projects/xquench/>), een in Java geschreven XQuery processor die echter wel nog maar in een eerste prille alpha release is. Gelijkaardig programma's zijn XQEngine (<http://www.fatdog.com/>) en Kawa-Query (<http://www.gnu.org/software/kawa/xquery>). Tenslotte zijn er nog een aantal onderzoeksprojecten waarin de XQuery taal verder verfijnd wordt, zoals Galax (<http://db.bell-labs.com/galax/>) van Bell Labs en IPSI-XQ (<http://xml.darmstadt.gmd.de/xquerydemo/>) van GMD-IPSI.

## Conclusies

XQuery heeft duidelijk nog een lange weg af te leggen alvorens het een geldige aanspraak kan maken op de titel van volwaardige XML data manipulatie taal. Hiërarchische datastructuren zoals XML die kent lijken op het eerste zicht eenvoudiger dan relationele datastructuren, maar XQuery is toch snel aan het uitgroeien tot een taal die veel complexer is dan SQL en toch veel minder mogelijkheden biedt. Bovendien is de taal zo ingewikkeld aan het worden dat te vrezen valt dat enkel de grote databank vendors nog in staat zullen zijn om hier een efficiënte implementatie van te bouwen. Gelukkig is er nu toch al een eerste werkversie van XQuery die binnenkort wel in de meeste native XML databank producten zal ondersteund worden. Een hele verbetering vergeleken met de wirwar aan vendor-specifieke query talen en API's die momenteel door deze producten gebruikt worden. Maar zolang er geen echte insert/update/delete functionaliteiten in XQuery voorzien worden lopen deze XML databanken wel het risico van hetzelfde lot beschoren te zijn als object-georiënteerde databanken. Technisch superieur voor een aantal gespecialiseerde toepassingen, maar qua bruikbaarheid en betrouwbaarheid duidelijk inferieur aan traditionele relationele databanken waaraan nu in snel tempo goede XML ondersteuning wordt toegevoegd.

---

Hans C. Arents ([hca@itworks.be](mailto:hca@itworks.be)) is een onafhankelijk XML technologie adviseur bij I.T. Works, de leidende Belgische organisator van produkt- en leveranciersonafhankelijke IT seminars. U kan meer over hem te weten komen op <http://www.arents.be/>.

### Informatie op het Internet:

<http://www.w3.org/XML/Activity#query-wg> (XML Query Working Group)

<http://www.w3.org/Style/XSL/> (XSL Working Group)

<http://www.w3.org/TR/query-datamodel/> (XQuery 1.0 and XPath 2.0 Data Model)

<http://www.w3.org/TR/query-algebra/> (XQuery 1.0 Formal Semantics)

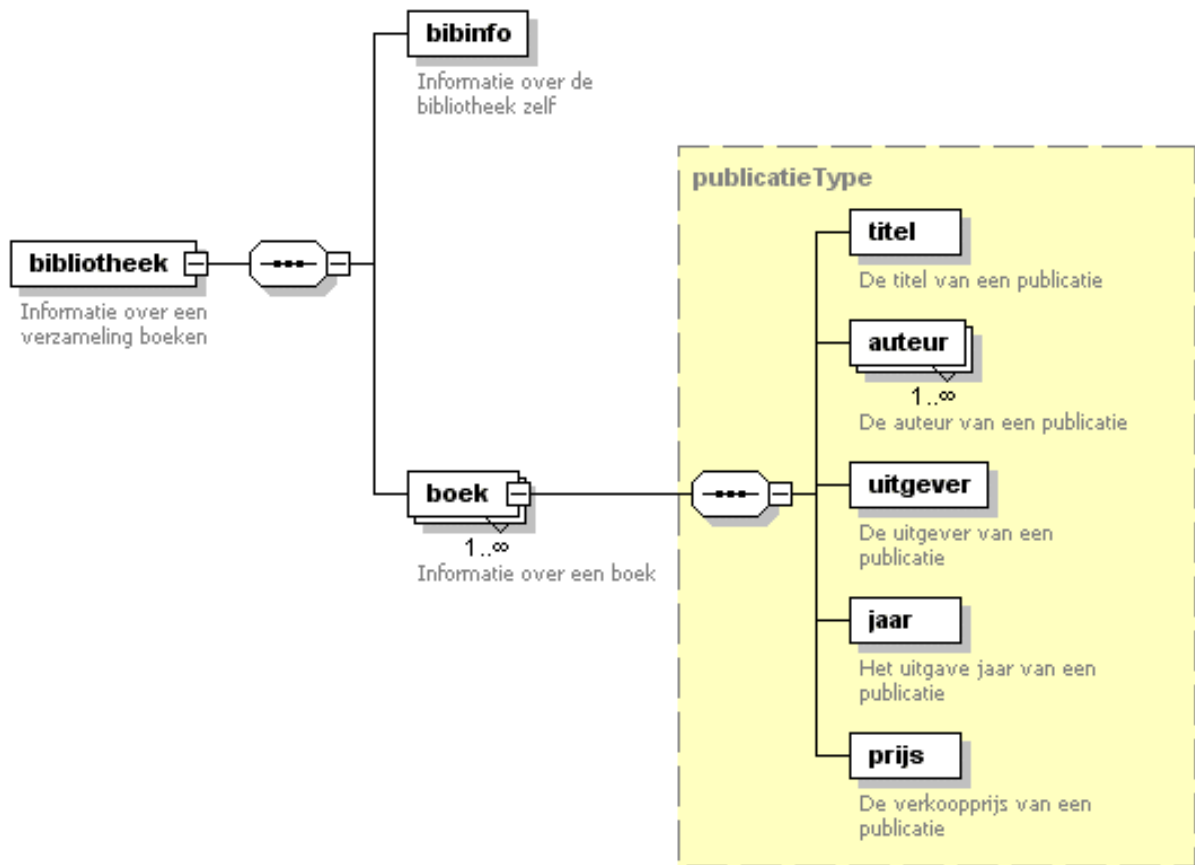
<http://www.w3.org/TR/xquery> (de XQuery 1.0 taal)

<http://www.lehti.de/beruf/diplomarbeit.pdf> (thesis waarop "Updates in XQuery" voorstel gebaseerd is)

<http://dbs.uni-leipzig.de/en/projekte/XML/XmiBenchmarking.html> (XMach-1 XML databank benchmark)

<http://www.brics.dk/~amoeller/XML/querying/index.html> (een korte XQuery tutorial)

<http://www-106.ibm.com/developerworks/xml/library/x-xquery.html> (artikel "An introduction to XQuery")



Figuur 1. Een grafische voorstelling van het document model voor het voorbeelddocument "bib.xml" (gebruikmakend van XML Schema syntax).