XI. FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2006. március 24-25.

XML-RELÁCIÓS ADATMODELL

Zörgő Zoltán

Summary

If we wish to achieve high performance and efficient data manipulation using database management systems which implements an existing model, data modeling is a necessary and inevitable task. Hierarchical, network, relational data models and their descendants have their own vision about the world, and their own tools, methods and methodology. Today, the data marked is ruled by systems implementing the relational model. The World Wide Web, and its widely spread application fields, emphasized the document models. As such, XML is mainly treated only as document. It is still raw material or end product. There are few and restricted ways to use them as we work with tables for example. In most cases mapping and transforms are used to treat relational data as XML or XML as relational data. The XML-Relational model is a new approach, aimed to combine the strengths of both the relational and the document data model. This paper presents the ground concepts of the model.

Összefoglalás

Napjaink informatikájának egyik legfontosabb és technológiailag leginkább támogatott területe az adattárolás és adatkezelés. Az információ hatalma nem csak annak birtoklásában, hanem annak hatékony feldolgozásában is rejlik. Az adatmodellezés megkerülhetetlen és szükséges tevékenység, ha hatékony adatkezelést szeretnénk megvalósítani valamilyen létező alapmodellre épülő adatbázis-kezelő segítségével. A klasszikus modellek és leszármazottaik sajátos eszközökkel és szemlélettel rendelkeznek. Napjainkban a relációs modellt implementáló adatbázis-kezelők a piacvezetők. A World Wide Web és annak igen szerteágazó alkalmazási területei előtérbe hozták a dokumentummodelleket. Az XML, mint leíró nyelv egyelőre inkább csak végtermék és nyersanyag. Ritkán, és csak szűkös keretek között épül rá valódi adatkezelés. Legtöbb esetben átjárókat használnak a két világ között. Az XML-relációs adatmodell egy új szemantikus és implementációs modell, mely lehetővé teszi a relációs világ és a strukturált egyedszemlélet erősségeinek ötvözését, egy egységes modellben. Ez a cikk a modell alapjait ismerteti.

Bevezetés

Mint minden új modell javaslata esetén fontos annak szükségességét megindokolni, és elhelyezni azt a többi – hozzá hasonló – modell és megvalósítás között. A DBTG által életre hívott CODASYL lefektette azokat a követelményeket az adatkezelés területén, melyek máig meghatározzák az információtechnológia e vonulatát. Az adatok kezelését jelentő széles feladatkör komoly elméleti alapokra helyeződött, és olyan absztrakciós szintek, mint az adatmodellek, végleg bekerültek az adatbázis-kezelés összetett világát meghatározó tényezők közé. Ez az időszak egy hatalmas ugrás volt,

és meghatározta a ma ismert hálós, hierarchikus és relációs modellek jellemzőit. Ismeretes, hogy napjaink adatbázisait szinte kizárólag a relációs modellt és utódait megvalósító szabványokat implementáló rendszerek kezelik – igen nagy hatékonysággal.

Az információtechnológia fejlődésének következő lépcsőjének a Webes technológiák elterjedését tekinthetjük. Mára már a Web nem csak a szörfölők paradicsoma, hanem például komoly B2B (Business To Business) kommunikáció csatornája is. Az automatizált Web, a WebSzolgáltatások, és a generált tartalmak a World Wide Web-et a HTML-ről az XML felé irányították. Az XML, mint strukturált leíró nyelv természetesen még mindig alkalmas ember általi olvasásra, de minden jel arra utal, hogy a gépek és szoftverek közötti hatékony adatcsere eszközévé válik elsősorban. A Web-dokumentumok leírása (XHTML) mellett számos szabványosított (SVG, MathML, SOAP, stb.) és megszámlálhatatlan egyéb sémával megadott XML fajta létezik. Mára az irodai alkalmazások is áttértek az XML alapú dokumentumkezelésre (például: OpenDocument). Látható, hogy az XML-technológia alkalmas a legáltalánosabb értelemben vett adatmegadásra. Ennek következtében az adatbázisokkal szemben is felmerült az XML-támogatás igénye – hiszen sok esetben kell strukturált adatot adatbázisban tárolni, vagy éppen adatbázisból kinyert adatot strukturált formában megjeleníteni. Ugyanígy megjelent az XML-el szemben az adatbázis-szemlélet igénye. A W3C válasza az XPath és XQuery szabványok kialakítása volt, melyek lehetővé teszik XML dokumentumokban való navigálást és lekérdezés. Ez egyaránt hordoz jegyeket az SQL-ből és a hierarchikus modell navigációs nyelvéből. Olyan kezdeményezés is létezik, mely az XML dokumentumok XSL-T alapú manipulációs nyelvét definiálja és implementálja [1]. Az adatbázis-kezelők fejlesztőinek válasza a feladatra nem egységes. Többféle lehetőség van. Egyfelől „kilapíthatjuk” a strukturált adatokat és az XML entitásokat és attribútumokat egy-egy rekord mezőinek feleltethetjük meg. Ez a folyamat mindkét irányban működik, és megfeleltetésnek (mapping), objektumok esetén sorosításnak (serialisation/deserialisation) nevezik. A tulajdonképpeni adatkezelés az adatbázis-kezelőben zajlik. A strukturáltság azonban elveszhet. Más megoldásként a strukturált adatokat CLOB jellegű mezőkben tárolják, de vagy magának az adatbázis-kezelőnek, vagy külső komponenseknek kell tudni olvasni az ilyen mezőket. Harmadik lehetőségként néhány adatbázis-kezelő lehetővé teszi strukturált adatok tárolását speciális tárolókban az adatbázison belül. Ehhez azonban hiányoznak a szabványok. Mindhárom megközelítésről elmondható, hogy alárendeli a strukturált adatot a relációsnak. Sara Cohen és társai [2] kidolgozták a strukturált adatok relációkká alakításának matematikai alapjait és javaslatot tettek egy vegyes lekérdező nyelvre.

A relációs modell elterjedését talán az magyarázza legjobban, hogy a valóság objektumainak nagy része valóban relációs entitásként viselkedik, de ugyanezek az entitások változatos és strukturált szerkezetűek is. Itt az XML segíthet. Egy olyan modellre és implementációs irányra lenne szükség, mely nem borítja fel a fejlesztők relációs szemlélethez igazodott módszereit, és kielégíti a strukturált adatok kezelésével szemben megjelenő igényeket. Az XML-relációs modell átfogó megoldást nyújthat.

Az XML-relációs modell ismertetése

A modell célja, hogy egyesített elméleti alapokon egységes eszközkészlet létrehozását tegye lehetővé a strukturált egyedek relációinak tárolására és kezelésére. Egy ilyen egyesítés természetesen szükségessé tehet kisebb megszorításokat a kiinduló modellekhez képest, az új lehetőségek azonban kárpótolják ezeket. A modell fejlesztői indíttatású. Implementációját használva egy weboldal generálása egy XSL-transzformáció segítségével akár már közvetlenül az adatbázis-kezelőből történhet. A terjedelmi korlátok miatt a modellnek csak a fontosabb elemeit ismertetjük.

A modell alapja az attribútum. Az attribútum, akárcsak a relációs modell esetén tartomány eleme. Az entitás attribútumok struktúrája. Ilyen értelemben az XML-ből megszokott entitás-attribútumok és az entitás levél-tartalma is tartományba kell, hogy tartozzon. Ennek következménye, hogy adat csak levélben lehet. A bal oldali példában a ennek megfelelő, mellette a nem megfelelő változatot emeltük ki. A modell sémanyelve segítségével az alaptípusokból saját származtatott adattípusok hozhatók létre.

```
<partner id="311254">  
  <név>Kiss Kft.</név>  
  <elérhetőségek>
```

```
<partner id="311254">  
  <név>Kiss Kft.</név>  
  <elérhetőségek>
```

```

<székhely>
  <irszám>1111</irszám>
  <település>Budapest</település>
  <utcahsz>Nagy tér 1.</utcahsz>
  <telefon>+36-1-1234567</telefon>
</székhely>
</elérhetőségek>
</partner>

```

```

Székhely:
<irszám>1111</irszám>
<település>Budapest</település>
<utcahsz>Nagy tér 1.</utcahsz>
Telefon: +36-1-1234567
</elérhetőségek>
</partner>

```

Az egyedek táblákba szerveződnek. A gyökér-entitás a tábla maga, de alá csak egyféle entitások rendelhetők. Ez elméleti korlátot nem jelent, hiszen a tábla egyedeinek sémáját egy XML séma (XML Schema Definition) [3] adja meg, így az elvben bármilyen lehet. Egy mai általános XML dokumentum úgy képzelhető el, mint egy tábla, mi csak egyetlen egyedet tartalmaz – a dokumentumot magát.

A relációs modell egyik leghatékonyabb és legerősebb eleme a megszorítások kezelése, különös tekintettel a hivatkozási megszorításokra. Az XML szabvány nem foglalkozik a szemantikai integritás kérdésével, bár többféle megoldás is van szemantikai megszorítások implementálására. Ezt azonban ki kell terjeszteni akár több attribútumot is egyszerre érintő megszorításnyelvvvel, mint ahogy például a RELAX NG sémanyelvben mintákkal implementálták [4]. Az új modell egyik legfontosabb eleme, hogy strukturált entitások, vagy akár azok alárendelt entitásai is relációban lehetnek más entitásokkal vagy azok alárendelt entitásaival. Látható, hogy a W3C XML Séma nyelv jó alap, de azt tovább kell fejleszteni. Ez a nyelv még nem készült el teljesen, de a következő példával illusztrálható.

A bal oldalon lévő XML-táblában három „partner” egyed található. Ebben a pillanatban az egyiknek egy, a másiknak két címe van, a harmadikról egyáltalán nem tartunk nyilván címet. A relációs modellben halmazértékű és opcionális attribútumok esetén normalizálást kellett volna alkalmaznunk. A jobb oldali XML-tábla a postai irányítószám alapján tartalmazza a településneveket. Természetes elvárás, hogy amennyiben az irányítószám funkcionálisan meghatározza a településnevet, a partnerek címeiben szereplő irányítószámok hivatkozzanak a megfelelő egyedre a jobb oldali táblából, és a környezet az integritás megőrzését biztosítani tudja.

```

<partners>
  <partner id="1">
    <name>Kiss Kft.</name>
    <telefon>+36-46-234567</telefon>
    <address postal_code="3534">Kiss Táborkok
      útja 1</address>
  </partner>
  <partner id="2">
    <name>Nagy Kft.</name>
    <telefon>+36-1-1111111</telefon>
    <address postal_code="3525">Széchenyi út.
      10</address>
    <address postal_code="1111">Vasas köz
      71/B</address>
  </partner>
  <partner id="3">
    <name>Pantom Bt.</name>
    <telefon>+36-30-1112333</telefon>
  </partner>
</partners>

```

```

<codes>
  <code postal_code="1111">Budapest</code>
  <code postal_code="1015">Budapest</code>
  <code postal_code="4032">Debrecen</code>
  <code postal_code="3530">Miskolc</code>
  <code postal_code="3534">Miskolc</code>
</codes>

```

A hivatkozási megszorítást a partnerek XML-tábla sémájában adhatjuk meg. A következő sémarészletben a dőlt-félkövér kiemelés a saját típus, a félkövér a hivatkozási megszorítás használatát illusztrálja.

```

<xslrs:schema id="partners" xmlns:xslrs="urn:schemas-xml-r-com:xml-r" xmlns:ownns="own-xsd">
  <xslrs:element name="partners" xslrs:Locale="hu-HU" xslrs:class="table">
    ...
    <xslrs:attribute name="postal_code" type="ownns:postal-code"
      xslrs:references="/codes/code[@postal_code=.]" />
    ...
  </xslrs:element>
</xslrs:schema>

```

Az XPath alapú hivatkozás a struktúrára egészen bonyolult szerkezetek is lehetővé tesz. Képzeljük el a következő hivatkozást: `xmlrs:references="/partners/*/address[@id=.]"`. Ha például a hivatkozott partnerek tábla különböző mélységeiben is van cím-entitás, a „*” segítségével ezek közül bármelyiknek az ID attribútuma kielégíti a megszorítást. Mindezekhez a jelenlegi indexelési megközelítés nem elegendő. Az entitások attribútumai és tartalmai fákat alkotnak. A fák levelei indexelhetők a ma is használt módon, de valójában nem rekordokat, hanem magukat a csomópontokat kell indexelni. A tábla sémája az egyedtípust strukturáltan adja meg. Az egyed-előfordulások ennek keretein belül bármilyen alakot felvehetnek, és mivel az egyedeket faként is kell tárolni és ellenőrizni, a hatékonyságot befolyásolhatja a séma bonyolultsága a DML utasítások esetén.

A modell a már hivatkozott [1] XML alapú adatkezelő nyelveket és a megszokott UPDATE, DELETE, INSERT utasítások egyaránt támogatja, ugyancsak XPath-al kiegészítve. Az alábbi példák egy ilyen utasításokat mutatnak be:

```
DELETE partner FROM partners WHERE */address@postal_code=1112;
DELETE partner/telefon FROM partners WHERE /name LIKE "Fantom%"
```

A strukturáltság miatt lehetőség van arra, hogy a feltételek az egyed bizonyos szintjeit vizsgálják, vagy hogy az egyedeknek csak bizonyos részét töröljük.

```
UPDATE codes SET code="Budapest XIII" WHERE fn:String(code@postal_code) LIKE "113_"
INSERT <address postal_code="4032">Egyetem tér 2</address> INTO partners/partner WHERE @id=3
```

A modell lekérdező nyelve is hasonló konstrukciókat kell tartalmazzon. A lekérdezések eredménye mindig XML struktúra. Az alábbi példa a korábbi két tábla összekapcsolt lekérdezése.

```
SELECT * FROM partners INNER JOIN codes ON partners/*/postal_code = codes/code@postal_code
```

Strukturált lekérdezés esetén azonban nem csak az fontos, hogy mit kérdezzünk le, hanem az is, hogy milyen struktúrában szeretnénk az eredményt látni. Ehhez XQuery vagy más lekérdező nyelv [5][6] lehetőségeit kell kibővíteni többtáblás eszközökkel. A relációs világban jól bevált „*” nem mindig egyértelmű ebben a helyzetben. Igen ígéretes nyelv a Quilt[7]. A modell jelen állapotában csak az összekapcsolás lehetőségeire tér ki.

Következtetések

Bár az XML-relációs modell még kidolgozás alatt van, a benne rejlő lehetőségek széles felhasználási területet jelenthetnek. Ha átemeljük az XML és köréje csoportosuló szabványokból és a relációs modellhez kapcsolódó szabványból a bevált eszközöket, egy komplex, nagyon hatékonyan és egy jövő elé néző adatkezelő eszközkészletet kaphatunk.

Irodalom

- [1] <http://xmldb-org.sourceforge.net/xupdate/>
- [2] Sara Cohen, Yaron Kanza, Yehoshua Sagiv: Generating Relations from XML Documents, 2003
in Proc 9th International Conference on Database Theory
- [3] <http://www.w3.org/TR/xmlschema-2/>
- [4] <http://www.relaxng.org/>
- [5] Serge Abiteboul et al.: The Lorel Query Language for semistructured data.
International Journal on digital libraries, 1997
- [6] Alin Deutsch et al.: A query language for XML
In Proc. of the int WWW. Conf., vol. 31, 1999
- [7] I. Manolescu et al.: Pushing XML queries inside relational databases
Tech. Report no. 4112, INRIA, 2001

Zörgő Zoltán, PhD hallgató

Debreceni Egyetem, 4030 Debrecen, Egyetem tér 1

Baross Gábor Szakközépiskola, 3532 Miskolc, Rácz Ádám u. 54-56., zorgoz@inf.unideb.hu