XML pod lupou

Jazyk XML se dostává stále častěji do popředí zájmu. Zmiňovali jsme se o něm například v únorovém Chipu v článku nazvaném "Proč XML?". Nyní vám přinášíme trojdílný seriál, v němž vám přiblížíme některé další možnosti, které XML nabízí. V prvním díle se zaměříme na XML data v databázovém přístupu…

Ve vizi zveřejněné Boswordthem a Brownem z Microsoftu počátkem tohoto roku v publikaci Bulletin of the IEEE CS TC on Data Engineering jsou aplikace na webu viděny jako otevřené a spolupracující. Co to znamená v praxi? Na webu bude jednoduché nalézat zboží a služby a každý zákazník bude například schopen:

objevit všechna místa s informacemi o hledané knize a objednat si ji z jednoho z těchto míst;

před odjezdem na měsíc na dovolenou objevit, kdo mu bude o víkendech sekat trávník, a tuto službu také zajistit;

otevřít spreadsheet nebo své vlastní zákaznické stránky, o něž se bude starat kterékoliv místo, které spravuje portfolio zákazníka, a pak provádět v těchto portfoliích změny.

Stručně řečeno, bude jednoduché objevovat data a aplikace a interaktivně k nim přistupovat pomocí webových služeb. O jaká data jde? Zúžíme-li pohled pouze na podnikové zdroje informací, je zajímavé, že pouze 10 % z nich je reprezentováno pomocí strukturovaných dat v klasických databázích. Zbytek tvoří nestrukturovaná nebo tzv. semistrukturovaná data. Ta jsou definována jako data, která jsou nepravidelná, neuspořádaná či neúplná a jejichž struktura se může měnit, a to dokonce nepredikovatelným způsobem. Patří sem již zmiňovaná data ve webových zdrojích, HTML stránky, bibtexovské soubory, ale i velmi speciální data, jako jsou například data biologická.

Z hlediska řízení a zpracování dat pomocí webových služeb je třeba ještě uvážit, že dnešní podnikové aplikace vyžadují přístup i k externím datům (webovým stránkám partnerů, dalším databázím textů či strukturovaných dat). Je – a hlavně bude – třeba zpracovávat a zajišťovat:

katalogy zboží zahrnující osobní (kastomizované) pohledy na nabídku zboží,

e-obchodování (objednávky, faktury),

e-brokering (co koupit a od koho včetně výběru vhodné databáze dat),

úlohy související s integrací heterogenních informačních zdrojů.

Zdá se, že jazyk XML je vhodným adeptem k dosažení této vize. XML data lze považovat za instanci semistrukturovaných dat. Příkladem využití XML, databází a webu může být zpracování dat v komunitě dodavatelů a překupníků. Vychází se z předpokladu, že dodavatelé mají vlastní relační databáze (s různými schématy) o nabízeném zboží. Dohodou je zajištěno, že oba uživatelé mají společný DTD pro výměnu dat o zboží a definují XML pohled nad svou databází.

Pak se odehrávají následující procesy:

překupníci vyšlou dotaz v XML-orientovaném jazyku,

dotaz se komponuje s XML pohledem na straně dodavatele,

dotaz se převede na dotaz v jazyku SQL,

dojde k vyhodnocení dotazu, materializaci a odeslání odpovědi.

XML data mohou být aplikacemi generována a také aplikacemi zpracovávána. Data mohou být do XML formátu transformována z relačních databází a naopak, XML dokument může být uložen v relační databázi. Mnohdy stačí pouze XML pohled na relační data. Pomocí pohledů se může realizovat i integrované vidění relačních a semistrukturovaných dat. Začínají se realizovat přímo databáze XML dat. V některých aplikacích jsou XML dokumenty dokonce vhodnější pro reprezentaci dat než jakékoliv jiné reprezentace (relační databáze, soubory). Typicky jde o hierarchické struktury (obr. 1), které lze v XML popsat mnohem přirozeněji (obr. 2) než například v relačním modelu dat (RMD).

XML schéma a XML databáze

Jazyk XML, původně chápaný jako nový standard sloužící na webu pro reprezentaci a výměnu elektronických dat (EDI), je tedy možné vidět i jako nový datový model sloužící pro reprezentaci informací. Jako takový může být i implementován, tj. je možné jej pojmout jako formát pro uložení dat.

Jakmile začneme uvažovat databáze XML dat, je třeba řešit řadu základních otázek, kterými se standard XML nezabývá. Patří sem zejména, jak

ukládat XML data,

extrahovat data z velkých XML dokumentů,

vyměňovat data (přenosem XML dokumentů nebo XML dotazů),
vyměňovat data mezi komunitami, které používají různé, avšak související DTD,
integrovat data z více XML zdrojů.

Pro uložení XML dat existuje mnoho přístupů. XML data lze ukládat v souborových systémech. Je možné využít software pro semistrukturovaná data. XML data se ukládají v objektových i relačních databázích, současné verze univerzálních serverů firem, jako je např. Oracle a Informix, ukazují, jak lze XML data organizovat v objektově-relačních databázích.

Existuje-li databáze XML dat, je přirozené vytvářet odpovídající dotazovací jazyky. Výměna dat pak může znamenat zkonstruovat pomocí takového jazyka z dokumentu validního vzhledem k jednomu DTD dokument validní vzhledem k druhému DTD. Integrace informačních zdrojů může být realizována dotazem nad více (dokonce heterogenními) databázemi XML dokumentů.

Z hlediska terminologie obvyklé v databázích se lze dívat na XML jako na jazyk modelování dat. Dobře vytvořený XML dokument (či množina takových dokumentů) je potom XML databáze a DTD její databázové schéma.

Modelování dat v XML

Z hlediska přístupu k XML datům z databázového hlediska má smysl se zabývat hlavně elementy a jejich hierarchickou strukturou, dále pak atributy. Atributy slouží k asociaci jména s hodnotou. Lze jimi popsat blíže obsah elementu, např. množství = “20”, měna = “EUR”. Atribut může obsahovat běžně nejvýše jednu hodnotu, u atributů typu IDREF dokonce více hodnot. Na rozdíl od elementů je množina atributů neuspořádaná. Jednoduchý příklad XML dokumentu je na obr. 3.

```
<adresář>
<osoba rod_č=”111-22-3333”>
   <příjmení> Kopecký </příjmení>
   <jméno> Michal </jméno>
   <s_titulem> Mgr. M. Kopecký </s_titulem>
   <adresa> Malostranské 25 </adresa>
   <adresa> Praha, 100 00 </adresa>
   <tel> 2191 4268 </tel>
   <fax> 2191 4323 <fax>
   <tel> 2191 4323 </tel>
   <email>kopecky@ksi.mff.cuni.cz </email>
  </osoba>
</adresář>
```

Obr. 3. Dokument z adresáře

```
<!DOCTYPE Adresář [
   <!ELEMENT adresář (osoba*)>
   <!ELEMENT osoba
         (příjmení, jméno, s_titulem?, adresa*,
      (fax | tel)*, email*)>
    <!ATTLIST osoba
                  rod_č    ID    #REQUIRED>
    <!ELEMENT jméno          (#PCDATA)>
   <!ELEMENT s_titulem (#PCDATA)>
   <!ELEMENT adresa (#PCDATA)>
   <!ELEMENT tel               (#PCDATA)>
   <!ELEMENT fax       (#PCDATA)>
   <!ELEMENT email    (#PCDATA)>
]>
```

Obr. 4. DTD adresáře

Pomocí atributů typů ID, IDREF a IDREFS lze snadno modelovat vztahy mezi elementy (1:1, 1:N a M:N). Představme si zaměstnance a projekty v relačních tabulkách ZAMĚSTNANCI(JMÉNO, ROD_Č, VĚK) a PROJEKTY(NÁZEV, ROZPOČET, ŘÍZEN). Atribut ŘÍZEN je v tabulce ZAMĚSTNANCI zřejmě cizím klíčem. Jedna z možností, jak reprezentovat či vidět data z relační databáze jako XML data, je na obr. 5.

Atraktivnější možností je chápat ROD_Č zaměstnanců jako atribut typu ID a ten využít v elementech PROJEKT pro elementy ŘÍZEN (obr. 6). Element ŘÍZEN je prázdný, obsahuje (zde nepovinně) odkaz na element toho zaměstnance, který projekt řídí. Protože element odpovídající Kopeckému v XML databázi na obr. 6 není, nelze se o řízení projektu Vyhledávání nic dozvědět.

DTD jako databázové schéma

Uvažovat DTD jako schéma v databázovém smyslu je ovšem pouze východisko z nouze. Z hlediska standardů databází (nebo programovacích jazyků) nahrazují DTD databázové schéma pouze nedostatečně:

K dispozici je pouze jeden základní typ PCDATA.
Neexistují žádné užitečné “abstrakce” jako množiny, multimnožiny, seznamy.
Hodnoty atributů IDREF jsou netypované (ukazuje se na něco, ale neví se, na co!).
Neexistují žádná integritní omezení.
Omezení na pořadí elementů, jak je vyžaduje DTD, mohou být příliš tvrdá.

```
 <db>                                          <zaměstnanci>
<projekty>                                      <zaměstnanec rod_č = 715512/0132 >
   <projekt>                                                <jméno> Mikulová,L.
      </jméno>
        <název> Vyhledávání </název>                        <věk> 38 </věk>
        <rozpočet> 100000 </rozpočet>                 </zaměstnanec>
       </řízen >                                      <zaměstnanec rod_č =
      700321/1423 >
   </projekt>                                         <jméno> Dvorský, J. </jméno>
   <projekt>                                          <věk> 29 < /věk>
      <název> Třídění </název>                                 </zaměstnanec>
        <rozpočet> 700000 </rozpočet>               </zaměstnanci>
        </řízen idref = ”715512/0132”>             </db>
   </projekt>
</projekty>
```

Obr. 6. XML data2, která vznikla z relační databáze

Již první bod může být v praxi problémem. Pro hodnotu atributu množství v objednávce zboží nemůžeme např. pomocí prostředků XML zkontrolovat, že jde o celé číslo z rozsahu 1-100. Co také vadí, je samotný jazyk pro popis DTD. Ten sám totiž není rozšiřitelný. Jak však ukazuje poslední vývoj, pracuje se na vytvoření jazyka pro tvorbu XML schémat. Podstatou těchto návrhů je zavedení typů hodnot a specifikace mohutnosti množiny. Např. pro

DTD: <!ELEMENT článek (titul, autor*, rok, (časopis|konference))>

má odpovídající schéma tvar na obr. 7. Řada dalších pokusů přibližuje XML data klasickým databázím. Explicitní integritní omezení lze využít při optimalizaci dotazů a podobně jako u klasických databázích pro filtrování “dobrých” dat do XML databáze.

```
<elementType name=”článek”>
    <sequence>
          <elementTypeRef name=”titul”/>
          <elementTypeRef name=”autor” minOccurs=”0”/>
          <elementTypeRef name=”rok”/>
          <choice> <elementTypeRef name=”časopis”/>
                        <elementTypeRef
     name=”konference”/>
          </choice>
    </sequence>
</elementType>
```

Obr. 7. XML schéma

Závěr

Přestože jsou XML schémata semistrukturovaná, jejich databázová budoucnost je založena na pojmu schéma podobně jako v klasických databázích. Toto schéma může být dáno implicitně značkami elementů nebo DTD definicemi XML dokumentu, případně pomocí složitější struktury – XML schématu. Teprve pak lze uvažovat o realizaci silných dotazovacích jazyků, o restrukturalizaci XML dokumentů a o konverzích mezi relační databází a XML dokumenty.

Jaroslav Pokorný
pokorny@ksi.ms.mff.cuni.cz