Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Seminarski rad iz predmeta
**„Sustavi za praćenje i vođenje procesa“**

# XML baze podataka

5. lipnja 2005.

Juraj Urbanke
0036383647

# SADRŽAJ

# Uvod

XML postaje podatkovni standard za elektroničko poslovanje, a količina XML podataka u razmjeni se povećava eksponencionalnom brzinom. To dovodi do potrebe za pohranjivanjem XML datoteka u baze podataka zbog brzog i preciznog dohvata.

# XML, XSLT, XPath, Xquery

**XML** (engl. *eXtensible Mark-up Language*) je *proširivi jezik zasnovan na oznakama* (engl. *tags*) koji pruža tekstualni format zapisivanja podataka neovisan o računalnoj platformi. XML je osnovni jezik pomoću kojeg su oblikovani ostali standardni jezici stoga mrežnih usluga. Za razliku od HTML-a gdje oznake određuju način prikazivanja podataka, XML oznake nose informaciju o značenju podataka. Također, XML uvodi dodatna pravila u odnosu na sintaksu HTML-a zbog lakšeg parsiranja i obrade. Nadalje, XML može sadržavati nestrukturirane podatke poput tekstualnih dokumenta i strukturirane podatke kao u bazama podataka. Moguće je i postavljanje upita na sadržaj dokumenta, odnosno moguće je postavljati upite prema strukturi i vrijednostima podataka u XML dokumentu. Suprotno od skupa oznaka jezika HTML koji je unaprijed definiran, skup oznaka XML-a ima svojstvo proširivosti, odnosno skup oznaka moguće je definirati i proširiti ovisno o potrebi. Stoga se XML koristi za razne primjene jer pruža mogućnost definiranja struktura i sintakse ovisno o potrebi. Primjerice, XML se koristi u užim industrijskim primjenama, no također i za izgradnju standardnih jezika.
Skup oznaka koji koristi određeni XML dokument naziva se rječnik (engl. *vocabulary*) XML dokumenta. S obzirom da je razvijeno mnoštvo nezavisnih rječnika XML dokumenata, moguće je da oznake i atributi istih imena posjeduju različito značenje u različitim rječnicima. Da bi se izbjeglo miješanje i zabuna uvodi se pojam prostora imena (engl. *namespace*) koji se zadaje pomoću URI-a (engl. *uniform resource indentifier*). Smještanjem u zadani prostor imena svaki element i svaki atribut jednoznačno su određeni.
Svaki XML dokument sadrži dva svojstva valjanosti: svojstvo dobre oblikovanosti te svojstvo ispravnosti. XML dokument je dobro oblikovan (engl. *well-formed*) ako slijedi određena pravila koja definira XML jezik. Primjerice, XML dokument smije imati samo jedan korijenski element, potrebno je da svaka početna oznaka ima pripadnu završnu oznaku te se zahtijeva da element dijete bude pravilno ugniježđen.
XML dokument sadrži određene podatkovne tipove koji su definirani hijerarhijskom strukturom XML elemenata ovisno o primjeni. *XML Schema* [41] jezik zasnovan na XML-u pruža standardan način definiranja strukture XML dokumenta i podatkovnih tipova. Primjerice, moguće je definirati jednostavni cjelobrojni tip podataka te složeni tip kao što je zapis sastavljen od cjelobrojnog podatka i znakovnog niza. Osim toga, pomoću *XML Schema-e* definiraju se ograničenja nad vrijednostima podatkovnih tipova. Stoga je svojstvo ispravnosti (engl. *validity*) XML dokumenta zadovoljeno ako podatkovni tipovi odgovaraju definiranim tipovima u pridruženom *XML Schema* dokumentu te ako

vrijednosti tipova zadovoljavaju postavljena ograničenja. Programska komponenta koja provjerava ispravnost XML dokumenata obično se naziva XML *validator*.

Primjer XML dokumenta:

```
<?xml version="1.0" encoding="UTF-8"?>
<CONTACT>
<NAME>Ivan Ivković</NAME>
<ADDRESS>Unska 3, Zagreb</ADDRESS>
<COMPANY>FER</COMPANY>
<TEL_NUM>555-444</TEL_NUM>
<EMAIL>ivan.ivkovic@fer.hr</EMAIL>
<NOTE/>
</CONTACT>
```

**XSLT** (engl. eXtensible Stylesheet Language: Transformations) je jezik za transformaciju XML i transformira strukturu XML dokumenta. XSLT sadrži naredbe kao i tradicionalni programski jezici kao što su varijable, funkcije, iteracije i provjere uvjeta. XSLT omogućuje razdvajanje sadržaja od prezentacije (kako bismo mogli informacije prikazati na različitim uređajima, u različitim oblicima,...) te se najčešće koristi za transformaciju XML u HTML. Također, XSLT omogućuje jednostavno transformiranje jednog XML-a u drugi XML, kako bi se uskladili protokoli i integrirala različita rješenja.

**XPath** je jezik za pronalaženje informacija u XML dokumentima. Predstavlja izraze koji mogu sadržavati put ka elementima XML dokumenta, pozive funkcija, reference na varijable, matematičke operacije itd. Koristi se zajedno sa XSLT da bi se odabrao skup elementa iz XML koji se želi transformirati.

**XML Query** (XQuery) omogućuje prilagodljive mehanizme za postavljanje uvjeta (engl. f*lexible query utilities*) za ekstrahiranje podataka iz dokumenata omogućujući potrebnu interakciju između *web*-a i baza podataka. XQuery je baziran na XML strukturi, a može se koristiti i za XML dokumente koji nemaju shemu. Jezik je dizajniran od strane W3C kao jezik s konciznom i lako razumljivom sintaksom.

## Tipovi XML baze podataka

## Flat files

*Flat file* je najjednostavniji oblik XML baze podataka. XML dokumenti se snimaju u datoteku, a njima se rukuje pomoću nekog API-ja (sučelja programske potpore). To je prihvatljiva metoda za mali skup XML dokumenata. Postoje alati za pretraživanje i modifikaciju, indeksiranje te transakcijsku obradu podataka.

Primjer *flat file*-a:

Diets\
Diets\Atkins
Day1.xml
Day2.xml
Day3.xml
Day4.xml
Diets\3DayDiet
Day1.xml
Day2.xml
Day3.xml

## Relacijske baze podataka

Skladištenje u relacijsku bazu podataka automatski nudi niz prednosti koje ove baze imaju nad *flat file* sustavom: višekorisnički pristup, sigurnost, transakcije,...

| XML | Relacijski model |
|---|---|
| ▪ podatci su smješteni u jednu hijerarhijsku strukturu | ▪ podatci su smješteni u više tabela |
| ▪ čvorovi imaju elemente i/ili atribute | ▪ obilježja imaju jednu vrijednost |
| ▪ elementi mogu biti ugniježđeni | ▪ vrijednosti obilježja su nedjeljive |
| ▪ elementi imaju definiran redoslijed | ▪ Redoslijed redova i stupaca na postoji |
| ▪ Shema je opcionalna | ▪ Shema je obavezna |

Tablica: razlike između XML-a i relacijskog modela

Tri su osnovna načina za smještavanje XML-a u relacijsku bazu podataka.

- CLOB (Character large object)
- „Čisto“ relacijski pristup (Pure relational)
- *XML enabled* relacijske baze podataka

Skladištenje XML-a u CLOB:

Kod pohrane na ovaj način pretraživanje i modifikacija se vrše nad dokumentom izvan baze podataka. Baza podataka „nije svjesna“ da smo u nju smjestili XML pa ne nudi nikakve servise za rad s XML-om. XML dokumenti nisu indeksirani što degradira perfomanse, a izgradnja vlastitog alata za indeksiranje se preporuča samo ako se radi s malim skupom XML dokumenata.

„Čisto“ relacijski pristup:

Kod ovog pristupa koriste se prednosti relacijskog modela – XML podatci se mapiraju na redove i stupce. Dvije najznačajnije tehnike mapiranja su *table-based* i *object-relational* mapiranje.

- *table-based* mapiranje

Mapiranje zasnovano na tablicama je korišteno od strane velikog broja proizvoda, a modelira XML dokument kao jednu ili kao skup tablica. *Table-based* mapiranje se koristi najviše za serijalizaciju podataka iz relacijske baze podataka.

Primjer:

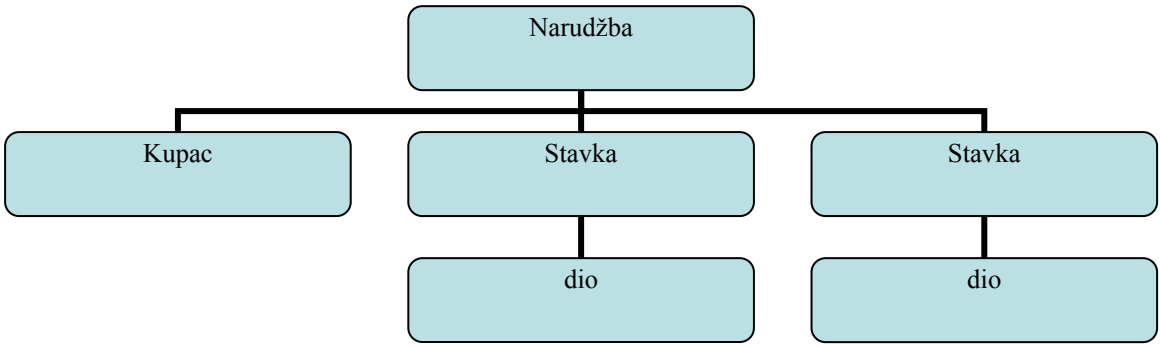
```
<database>
    <table>
        <row>
            <column1>...
            </column1>
            <column2>...
            </column2>…
        </row>
        <row> ...
        </row> ...
    </table>
    <table> ...
    </table> ...
</database>
```

- *object-relational* mapiranje

Koristi se od strane svih *XML enabled* relacijskih baza podataka. XML se modelira kao stablo objekta. Kao klase objekta se modeliraju elementi koji sadrže atribute, mješoviti sadržaj i složeni elementi, a kao skalarna obilježja se modeliraju jednostavni elementi i PCDATA. Zatim se klase mapiraju u tablice, a skalarna obilježja u stupce

Primjer:

```
<?xml version ="1.0"?> <
<Narudžba brojNarudžbe="12345">
    <Kupac brojKupca="543">
        <NazivKupca>FER </NazivKupca>
        <Ulica>Unska 3 </Ulica>
        <Grad>Zagreb </Grad>
        <PoštanskiBroj>10000</PoštanskiBroj>
        <Država>Hrvatska</Država>
    </Kupac>
    <Datum>5.6.2005.</Datum>
    <Stavka brojStavke="1">
        <Dio brojDijela="123">
            <Opis>
                <p><b>Francuski ključ:</b><br/>
                Nehrđajući čelik, doživotna garancija.</p>
            </Opis>
            <Cijena>735.68</Cijena>
        </Dio>
        <Količina>10</Količina>
    </Stavka>
    <Stavka brojStavke="2">
        <Dio brojDijela="456">
            <Opis>
                <p><b>Pištolj za bojanje:</b><br />
                Aluminij, jednogodišnja garancija.</p>
            </Opis>
            <Cijena>1452.78</Cijena>
        </Dio>
        <Količina>5</Količina>
    </Stavka>
</Narudžba>
```



Glavni problem kod „čisto“ relacijskog pristupa slijede iz razlika između XML formata i relacijskog modela. Samo mapiranje može biti složen posao, a manipulacija nastalom strukturom još složenija.

*XML enabled* relacijske baze podataka:

Kao odgovor na ove probleme, u relacijske sustave su ugrađeni mehanizmi koji olakšavaju rad s XML dokumentima. *XML enabled* baza podataka je „svjesna” da radi s XML strukturama i u skladu s time nudi različite servise. Ovakve baze podataka mogu smjestiti XML u CLOB, ali i u više tabela kada se koristi objektno-relacijsko mapiranje. Nad takvim podacima su implementirane mogućnosti za pretraživanje teksta koje su specifične za XML dokumente, a iste omogućuju jednostavniju izradu efikasnih aplikacija.
Nove mogućnosti uglavnom uključuju podršku za upitne jezike (najčešće XPath) i API-je (najčešće DOM = Document Object Model) koji su u skladu s preporukama World Wide Web Consortiuma. Koristeći SQL i njegove dodatke se lociraju slogovi, a baza podataka omogućuje pogled na te podatke u XML formatu.
*XML enabled* baze podataka, dakle, ne uvode novi model za rad s XML-om već proširuju postojeći – relacijski.
Najpoznatije realizacije *XML enabled* baza podataka su:

- Oracle 9i
- Oracle 10g
- Microsoft SQL Server 2000
- Microsoft Access XP
- IBM DB2
- ...

## Native XML

Efikasno skladištenje XML-a u RDBMS (*Relational DataBase Management System*) podrazumijeva implementaciju tehnika za premošćivanje razlika između dva modela. Alternativni pristup je izgradnja novog tipa baze podataka koja bi:

- kao osnovnu logičku jedinicu imala XML dokument
- podacima manipulirala koristeći XML standarde

Takvi proizvodi se nazivaju *Native XML* baze podataka. *Native XML* baze definiraju logički model za dokument, a ne definiraju uvjete za podatke u dokumentu. Stoga se sve komponente XML modela skladište bez izmjene. U bazu se može smjestiti samo XML, a iz baze se može dobiti samo XML. XML format se automatski mapira na odabrani model za fizičko skladištenje podataka. Takvim podacima se upravlja samo posredno; pomoću baze podataka i XML upitnih jezika.
Sve *Native XML* baze podržavaju jedan ili više upitnih jezika. U početku se najviše koristio XPath, ali kako ovaj jezik nije bio kreiran za tu namjenu, W3C je razvio XQuery upitni jezik. *Native XML* baze podržavaju indeksiranje sadržaja što, naravno, bitno utječe na performanse.

Promjena i brisanje se mogu izvesti na 3 načina: izvan baze pomoću nekog API-ja; da se za tu namjenu definira neki jezik (XUpdate inicijativa); sam upitni jezik sadrži i ove mogućnosti.
Sve *Native XML* baze podržavaju upotrebu nekog od API-ja. API-ji nude mogućnost za spajanje na bazu, postavljanje upita, pretraživanje meta-podataka, ... Rezultati se obično vraćaju u formi XML stringa ili DOM stabla. Ako rezultat upita može sadržavati više dokumenata, podržane su i metode za iterativni prolazak kroz skup.

Primjer *Native XML*-a:

```
<?xml version=”1.0”?>
<diet>
<meal mealName=”breakfast”>
<item itemName=“toast” unit=”slice” quantity=“2” />
<item itemName=“bacon” unit=”strip” quantity=“2” />
</meal>
</diet>
```

Najistaknutije implementacije *Native XML* baza podataka su:

- Tamino (SoftwareAG)
- Xindice (Apache)
- oZone
- eXist
- ...

## Izbor načina skladištenja XML dokumenta

Jedan od najbitnijih faktora pri izboru načina skladištenja podataka je oblik onoga što skladištimo.
*Data centric* dokumenti koriste XML za transport podataka, a namijenjeni su prvenstveno za čitanje od strane računala. Osobine su im: a) regularna struktura, b) fino strukturirani podatci, c) malo ili nimalo mješovitog sadržaja. Često potječu iz baze podataka.
*Document centric* dokumenti su obično namijenjeni čitanju od strane čovjeka – knjige, e-mailovi, ... Osobine su im: a) manje regularna struktura, b) slabo strukturirani podatci, c) dosta mješovitog sadržaja. Obično ne potječu iz baze podataka.

Pri izboru načina skladištenja ne postoji opće pravilo, ali se generalno može reći da su relacijske baze podataka prikladnije u onim tipovima XML dokumenata gdje su integritet konzistentnost podataka na prvom mjestu (tj. *data centric* dokumenti). S druge strane, *Native XML* je prikladniji za *document centric* dokumente koji često sadržavaju strukture koje je teško mapirati na relacijski model.

## Literatura i izvori:


1. Ronald Bourrett: „XML and databases“, December 2004
2. Michael Champion: „Storing XML in databases“, eAI Journal, October 2001
3. Beth Bowden: „Clearing the confusion about XML databases“, Builder.com, June 2003
4. Jovana Vidaković: Predavanja iz izbornog seminara „XML baze podataka“ Prirodoslovno-matematičkog fakulteta Sveučilišta u Novom Sadu
5. Davor Čapalija: “Objava/pretplata mehanizmi za ostvarivanje mreža zasnovanih na sadržaju”, diplomski rad u pripremi, Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu