Introducere in XML (eXtensible Markup Language)

Mihai Gabroveanu <mihaiug@central.ucv.ro>

Abstract

In cadrul acestui curs sunt prezentate conceptele de baza ale limbajului XML.

Table of Contents

1. Introducere

Prin date structurate intelegem lucruri ca spreadsheets, liste de contacte, parametri de configuratie, tranzactii financiare sau desene tehnice. XML este un set de reguli (poti sa le consideri si conventii) pentru a crea formate text care iti permit sa structurezi datele. XML nu este un limbaj de programare si nu trebuie sa fii programator pentru a-l invata si folosi. Cu XML, unui calculator ii este usor sa genereze si sa citeasca datele, cat si sa se asigure ca structura datelor este corecta. XML evita problemele obisnuite ale limbajelor de programare: este extensibil, independent de platforma si suporta internationalizarea si localizarea. XML este complet compatibil cu Unicode.

2. Scurt Istoric

XML (e**X**tensible **M**arkup **L**anguage), descendent al SGML (Standard Generalized Markup Language) este un meta-limbaj utilizat in activitatea de marcare structurala a documentelor, a c rei specificatie a fost dezvoltata incepand cu 1996 in cadrul Consortiului World Wide Web (W3C), de un grup de cercetare condus de Jon Bosak de la Sun Microsystems, la care au aderat ulterior o serie de grupuri de experti din comunitatile academice (Text Encoding Initiative, NCSA, James Clark) si industriale (SUN, IBM, Netscape, Oracle, Adobe etc.). Prima versiune de XML a fost standardizata in februarie 1998, ulterior acest standard a mai fost revizuit de doua ori in octombrie 200 si respectiv in februarie 2004.

Scopurile proiectate pentru XML sunt::

1. XML trebuie sa fie simplu de utilizat pe Internet.

2. XML trebuie sa suporte o mare verietate de aplicatii.

3. XML trebuie sa fie compatibil cu SGML.
4. Trebuie sa fie usor sa fie scrise programe ce vor procesa documente XML.
5. Numarul facilitatiilor optionale din XML sunt reduse la minimum, ideal, la zero.
6. Documentele XML trebuie sa fie citibile de catre utilizatori si clare intr-un mod rezonabile.
7. Designul XML ar trebui sa fie pregatita rapid.
8. Designul XML trebuie sa fie formal si concis.
9. Documentele XML trebuie sa fie usor de creat.
10. Caracterul lapidar din marcajele XML sa fie de o importanta minima.

3. Ce este XML?

Documentele XML sunt realizate din unitati de stocare numite entitati, ce contin date parsate sau neparsate. Datele parsate sunt realizate din caractere, unele dintre ele formand date caracter iar altele ca marcaje. Marcajele codifica o descriere a schemei de stocare a documentului si structura logica. XML furnizeaza un mecanism pentru a impune constringeri asupra schemei de stocare si a structurii logice.

XML a fost elaborat pentru:

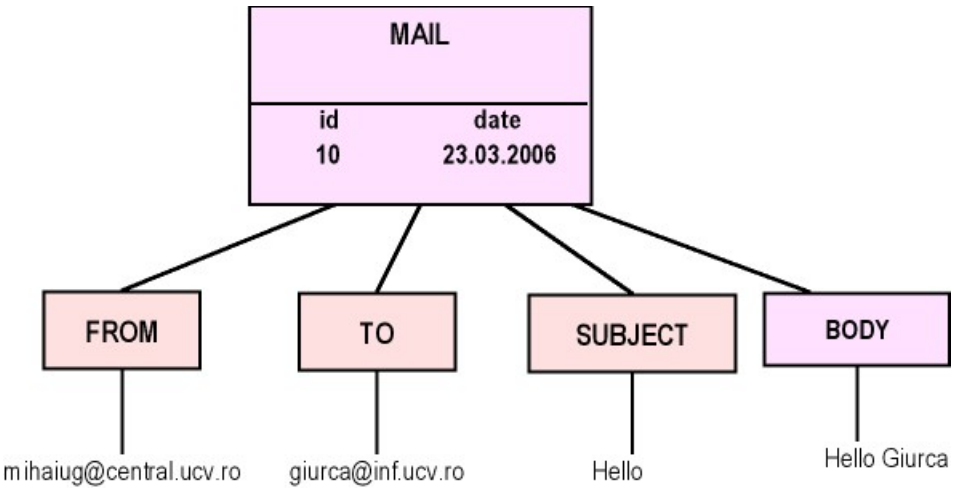
- separarea **sintaxei** de **semantica** pentru a furniza un cadru comun de structurare a informatiei
- **construirea de limbaje de mark-up** pentru aplicatii din orice domeniu
- **structurarea informatiei** in viitor
- **asigurarea independentiei** de platforma si suport pentru **internationalizare**

Un document XML este un **arbore ordonat etichetat**:

- **date caracter** - noduri frunza ce contin datele
- noduri **elemente** etichetate cu
  - un nume (adesea numit si tipul elementului) si
  - o multime de atribute, fiecare din ele avand un nume si o valoare

  acestea pot contine unu sau mai multi copii.

Un exemplu de reprezentare a unui document XML este urmatorul:



4. Structura documentelor XML

Un document XML este format din *marcaje (tag-uri)* si date caracter.

Cuvantul *marcaj (markup)* a fost folosit initial pentru a descrie anumite adnotari, note marginale in cadrul unui text cu intentia de a indica tehnoredactorului cum trebuie listat un anumit pasaj. Generalizand, putem defini marcajul drept orice actiune de a interpreta explicit o portiune de interpreta explicit o portiune de text.

Un *marcaj (tag)* este un sir de caractere delimitat de caracterele "<" si ">". *Datele caracter* reprezinta continutul marcajelor.

În XML, marcajele nu sunt folosite pentru afisarea datelor continute, ci au alte scopuri printre care:

- asigura o sintaxa simpla si standardizata pe care analizoarele XML o pot folosi pentru a utiliza informatia stocata
- asigura o metoda de a descrie structura ierarhica a continutului prin impartirea informatiei (datele caracter) in parti numite *elemente* care mai departe sunt descrise prin *atribute*. Structura ierarhica a intregului document este pusa in evidenta prin utilizarea marcajelor.

Un fisier XML cuprinde urmatoarele sectiuni:

- Prolog
- Definitia tipului de document (optionala)
- Elementul radacina

Example 1. Fisierul `mail.xml`

```
 1   <?xml version="1.0" encoding="UTF-8"?>  ❶
     <!DOCTYPE MAIL SYSTEM "MAIL.DTD">❷
     <MAIL id="10" date="23-03-2006">❸
           <FROM>mihaiug@central.ucv.ro</FROM>
 5         <TO>giurca@inf.ucv.ro</TO>
           <SUBJECT>Hello</SUBJECT>
           <BODY>
             Hello Giurca
           </BODY>
10   </MAIL>
```

❶ Prologul:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Este o instructiune de procesare. Ea informeaza ca urmeaza descrierea unui fisier XML ce respecta versiunea de specificatie 1.0 iar setul de caractere utilizat este encodat **UTF-8**

❷ Definitia tipului de document

```
<!DOCTYPE MAIL SYSTEM "MAIL.DTD">
```

Precizeaza ca fisierul `MAIL.DTD` contine declaratia tipului de document (DTD-ul), document ce are ca radacina tag-ul **MAIL**. Acesta este un set de reguli ce definesc structura unui fisier XML.

❸ Elementul radacina

```
  <MAIL id="10" date="23-03-2006">
       <FROM>mihaiug@central.ucv.ro</FROM>
       <TO>giurca@inf.ucv.ro</TO>
       <SUBJECT>Hello</SUBJECT>
       <BODY>
         Hello Giurca
       </BODY>
 </MAIL>
```

Acesta este elementul radacina al documentului XML.

4.1. Sintaxa unui document XML

Un document XML poate contine urmatoarele tipuri de marcaje:

- Elemente
- Atribute
- Comentarii
- Entitati
- Sectiuni CDATA
- Instructiuni de procesare
- Declaratia tipului de document

4.1.1. Elementele

Elementele sunt blocurile de baza ale unui document XML. Pot fi folosite atat pentru a retine informatii, cat si pentru definirea structurii. Un element incepe cu un tag de start si se termina cu corespunzatorul tag de sfarsit:

```
<nume_tag❶> ........ </nume_tag❷>
```

❶ tagul de start
❷ tagul de sfarsit

Un element poate fi vid, adica nu contine nimic. Sintaxa pentru un element vid este urmatoarea:

```
<nume_tag/>
```

Important

Tag-uri sunt *case sesitive*, adica se face distinctia intre litere mari si litere mici.

De exemplu, urmatoarele exemple de taguri sunt gresite:

```
<Student>Popescu</STUDENT>
```

```
<StudenT>Popescu</student>
```

Numele unui tag este o sucesiune de caractere alfa-numerice ce *incepe obligatoriu cu o litera*. Astfel <7nume> este eronat.

Elementele sunt utilizate nu numai pentru a retine informatii, ele se utilizeaza si pentru a defini structura documentului:

Example 2. biblioteca.xml

```
<?xml version="1.0"?>
<BIBLIOTECA>
    <CARTE>
        <TITLU>XML Bible</TITLU>
        <AUTOR>Elliotte Rusty Harold</AUTOR>
        <EDITURA>IDG Books Worldwide</EDITURA>
        <AN_APARITIE>2002</AN_APARITIE>
    </CARTE>
</BIBLIOTECA>
```

In acest exemplu, observam ca elementele <TITLU>, <AUTOR>, <EDITURA>, <AN_APARITIE> contin informatii, in timp ce elementele <BIBLIOTECA>, <CARTE> sunt folosite doar pentru a defini structura datelor. Prin folosirea lor, datele sunt mai bine organizate, facilitand eventualele operatii de cautare, afisare sau sortare a datelor.

4.1.2. Atribute

Atributele au rolul de a descrie elementele. Putem face o analogie intre atribute – care descriu elemente si adjective – care descriu substantive.

Atributele in XML sunt aceleasi cu atributele din HTML. De exemplu, un atribut al elementului <table> ar fi align="center". Atributele sunt localizate in tag-ul de start al unui element, imediat dupa numele elementului, (acum este evident de ce nu pot apare spatii albe in numele unui element), sunt urmate de semnul ‘=’, care este urmat de valoare atributului intre ghilimele. Daca valoarea unui atribut nu este intre ghilimele va fi semnalata eroare de catre analizorul XML, la fel ca si in cazul in care pentru un atribut nu ar apare si valoarea acestuia.

Astfel, sintaxa generala este urmatoarea:

```
<nume_tag numeAtribut1="valoare1" ... numeAtributN="valoareN"> . . . </nume_tag>
```

Pentru un element pot exista oricate atribute, atat timp cat sunt declarate corect.

Exemplu:

```
<?xml version="1.0"?>
<BIBLIOTECA>
    <CARTE cota="12345">
        <TITLU>XML Bible</TITLU>
        <AUTOR>Elliotte Rusty Harold</AUTOR>
        <EDITURA> IDG Books Worldwide</EDITURA>
        <AN_APARITIE>2002</AN_APARITIE>
    </CARTE>
</BIBLIOTECA>
```

4.1.3. Comentarii

Comentariile sunt secevente de caractere ce pot aparea oriunde in document in afara altor marcaje ce sunt utilizate pentru a descrie anumite detalii legate de continutul si structura documentului programatorului. Ele nu fac parte din datele caracter ale documentului; un procesor XML poate, dar nu este nevoie, sa dea un acces aplicatiilor la comentarii.

Un comentariu incepe cu secventa <!-- si se incheie cu -->.

```
<!-- comentariu -->
```

Comentariile pot fi oricat de lungi, nu exista limite in ceea ce priveste lungimea lor. De asemenea, un cometariu nu poate sa contina secventa de caractere --.

Exemplu:

```
<?xml version="1.0"?>
<!-- Documentul retine cartile dintr-o biblioteca -->
<BIBLIOTECA>
    <CARTE cota="12345">
        <!--  titlul cartii -->
        <TITLU>XML Bible</TITLU>
        <!-- Autorul Cartii -->
        <AUTOR>Elliotte Rusty Harold</AUTOR>
        <!-- Editura in care a aparut cartea -->
        <EDITURA> IDG Books Worldwide</EDITURA>
        <!-- Anul de aparitie a cartii -->
        <AN_APARITIE>2002</AN_APARITIE>
    </CARTE>
</BIBLIOTECA>
```

4.1.4. Referinte la entitati

Referintele la entitati sunt de fapt pointeri catre entitati. În XML, entitatile sunt unitati de text, unde o unitate poate fi orice, de la un singur caracter la un intreg document sau chiar o referinta la un alt document.

Sintaxa referintelor la entitati este:

```
&nume_entitate;
```

‘&’, urmat de numele entitatii, urmat de ‘;’

Una dintre cele mai frecvente utilizari ale referintelor la entitati este atunci cand se doreste folosirea unor caractere care ar duce la aparitia unor confuzii pentru analizorul XML si deci care nu ar trebui sa apara in forma lor normala in text. În acest caz exista cinci entitati predefinite in XML:

Table 1. Entitati definite in XML

Entitate	Referinta la entitate
<	&lt;
>	&gt;
&	&amp;
'	&apos;
"	&quote;

În momentul in care analizorul XML intalneste referinta la o entitate in document, el o va substitui cu datele pe care aceasta le refera si va returna documentul cu inlocuirile facute.

Exemplu:

```
<TITLE>Tom &amp; Jerry</TITLE>
```

dupa analizarea textului de catre analizorul XML, va rezulta:

```
Tom & Jerry
```

O alta utilizare frecventa a referintelor la entitati este in cazul in care avem in documentul XML fragmente de text care se repeta. Pentru a nu scrie aceste parti de text de mai multe ori vom defini o entitate care va avea ca valoare acea parte de text si de fiecare data cand fragmentul respectiv apare in document vom folosi referinta la entitate.

Prin folosirea referintelor la entitati se vor obtine documente mai scurte si se va scurta timpul de redactare.

4.1.5. Instructiuni de prelucrare

Instructiunile de prelucrare sunt un tip special de marcaj care contin informatii despre anumite aplicatii ce urmeaza a fi executate. Sintaxa generala a unei instructiuni de procesare ar fi urmatoarea:

```
<?aplicatie instructiune="valoare" ?>
```

Încep cu <?, urmat de numele aplicatiei si de specificarea unor elemente ce tin de acea aplicatie si se incheie cu ?>. Numele aplicatiei trebuie sa fie diferit de xml sau XML, sau alte moduri de scriere a acestui cuvant, deoarece cuvintele de acest tip sunt rezervate, urmand a fi standardizate intr-o versiune ulterioara.

4.1.6. Sectiuni CDATA

Sectiunile CDATA sunt utilizate pentru a include blocuri de text continand caractere care altfel ar fi recunoscute ca marcaje. Sectiunile CDATA incep cu sirul <![CDATA[si se termina cu sirul]]>.

Sectiunile CDATA sunt folosite in general atunci cand dorim ca datele incluse in interiorul lor sa nu fie interpretate de catre analizor, ci sa fie considerate date caracter. Astfel de situatii se intalnesc cand dorim sa includem caractere ca ‘<’, ‘>’, ‘&’ etc. care ar putea crea confuzii analizorului XML si folosirea lor ar putea duce chiar la generarea de erori sau cand dorim sa includem marcaje care sa fie considerate drept date caracter si sa fie expuse utilizatorului ca atare. Spre exemplificare, consideram un fragment de document XML care contine informatii despre cum se poate crea un tabel in HTML:

```
<?xml version="1.0" encoding="UTF-8"?>
<exemplu>
Un exemplu de creare a unui tabel in HTML:
 <![CDATA[
        <table align=”center”>
                <tr>
                    <td>Coloana 1</td>
                    <td>Coloana 2</td>
                </tr>
        </table>
]]>;
</exemplu>
```

Folosind sectiunea CDATA, analizorul va ignora continutul acesteia si datele vor fi expuse utilizatorului exact in forma in care sunt, si datele nu vor fi interpretate drept marcaje, ci drept date caracter.

O restrictie de sintaxa este faptul ca in interiorul sectiunilor CDATA nu poate sa apara sirul ']]'. Înca un lucru de retinut este ca sectiunile CDATA nu pot fi incluse unele in altele.

4.1.7. Declaratia tipului de document

Declaratia tipului de document este un marcaj special ce poate fi inclus in documentele XML cu rolul de a specifica existenta si locatia *definitiei tipului de document* (DTD –Document Type Definition). Trebuie sa retinem ca declaratia tipului de document si definitia tipului de document sunt notiuni diferite.

DTD-ul este un set de reguli care definesc structura unui document XML, spre deosebire de declaratia tipului de document care are rolul de a “spune” analizorului ce DTD trebuie sa foloseasca pentru verificare si validare.

Sintaxa declaratiei tipului de document difera in functie de tipul DTD-ului: intern sau extern. Considerand ca avem un document XML numit doc.xml, modul de asociere dintre structura sa si setul de reguli specificate in reguli.dtd este inserand in documentul XML, imediat dupa declaratia XML, urmatoarea declaratie a tipului de document:

```
<!DOCTYPE root SYSTEM "reguli.dtd">
```

unde ‘root’ este elementul radacina al documentului XML, iar ‘reguli.dtd’ este numele DTD-ului asociat documentului.

În cazul in care DTD-ul este intern, declaratia tipului de document va avea urmatoarea forma:

```
<!DOCTYPE element_radacina [
<!-- Setul de reguli-->
]>
```

4.2. Documente bine formate (Well-Formed Documents)

Un document XML este un document bine format daca satisface urmatoarele conditii sintactice:

- au exact un singur element radacina (root element)
- fiecare element are un tag de inceput si unul de sfarsit
- tag-urile sunt inchise corect, adica nu sunt de forma:

  ```
  <autor><nume>Elliotte Rusty Harold</autor></nume>
  ```

  Primul tag deschis trebuie sa fie ultimul care este inchis. Tag-urile trebuie inchise exact in ordinea inversa a deschiderii lor, altfel va fi semnalata eroare.
- numele atributelor sunt unice in cadrul unui element

Cu alte cuvinte un document XML este bine format daca respecta regulile sintactice descrise de standardul XML.

5. Bibliografie

- Extensible Markup Language (XML) 1.0 (Third Edition) W3C Recommendation 4th February 2004, François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler
- Elliotte Rusty Harold, XML Bible. IDG Books Worldwide, Inc, 919 E. Hillsdale Blvd., Suite 400, Foster City, CA 94404
- http://www.w3schools.com/xml/