

## XML. De ce? Cum? Unde?

Cercet.st. Anton-Mugurel ROG  
Institutul pentru Tehnologii Avansate, Bucuresti

*În articol se încearca definirea XML ca limbaj de programare, ca standard de creare a documentelor XML specifice si ca modalitate de transmitere a datelor între aplicatii scrise atât în limbaje diferite, cât si pe platforme de dezvoltare diferite. De asemenea, se descrie structura logica si fizica a documentelor XML, precum si modalitatea de utilizare a acestora la scrierea unor pagini de Web cu limbajul HTML.*

**Cuvinte cheie:** XML, HTML, web, java, script, SGML, document, entitate, analizor, INTERNET, INTRANET.

### Introducere

Extensible Markup Language, abreviat XML, descrie o clasa de obiecte numite documente XML si, partial, comportamentul unor programe de calculator care le proceseaza. XML este o forma restrânsa a SGML (Standard Generalized Markup Language). De asemenea, XML poate fi considerat si un format de stocare a datelor. Prin constructie, documentele XML sunt conforme cu documentele SGML. Documentele XML sunt realizate din unitati de stocare, numite entitati. Un modul software numit *procesor XML* este utilizat pentru a accesa structura si continutul documentelor XML. Procesorul XML ruleaza în spatele aplicatiilor care au ca facilitate analiza si accesarea continutului documentelor XML. De asemenea, XML este caracterizat de o specificatie care descrie comportamentul cerut unui procesor XML despre cum trebuie sa citeasca datele din documentele XML si ce informatii trebuie sa furnizeze aplicatiei în spatele careia ruleaza.

### Origini si deziderate

XML a fost dezvoltat de catre un grup de lucru (XML Working Grup - cunoscut la început ca si SGML Editorial Review Board) format sub auspiciile Consorțiului World Wide Web (W3C), în anul 1996. Scopurile proiectate pentru XML sunt:

- Trebuie sa fie simplu de utilizat pe Internet/Intranet;

- Trebuie sa suporte o mare varietate de aplicatii;
- Trebuie sa fie compatibil cu SGML;
- Trebuie sa fie usor de scris programe care vor procesa documente XML;
- Numarul facilitatilor optionale din XML sunt reduse la minimum, ideal, la zero;
- Documentele XML trebuie sa poata fi citite usor de catre utilizatori;
- Proiectarea XML ar trebui sa fie pregatita rapid;
- Designul XML trebuie sa fie formal si concis;
- Documentele XML trebuie sa fie usor de creat.

### De ce XML ?

În lume exista milioane de formate de stocare a datelor. Fiecare program are propriul format de stocare a datelor. Problemele apar atunci când dorim sa schimbam date între aceste programe. Trebuie sa realizam programe de conversie care sa transmita datele de la program la altul, ceea ce înseamna timp, resurse si rata de erori mai mare. Mai mult, sunt cazuri în care datele nu sunt portabile de pe o platforma pe alta. De aceea, sa pus problema gasirii unui format comun de stocare a datelor care sa poata fi recunoscut de toate programele si care sa fie portabil de pe platforma pe alta. Astfel, a aparut ideea descrierii datelor cu marcate. De exemplu, un cont se descrie astfel:

```
<CONT>
  <NR_CONT>110002222</NR_CONT>
  <TITULAR>POP ION</TITULAR>
  <SOLD>100.000.000</SOLD>
</CONT>
```

Problema era ca programele care prelucrau aceste date trebuiau sa stie sa le citeasca si, deci, sa cunoasca dinainte structura marcatelor. Ideea geniala îi apartine lui Charles Goldfarb de la IBM, si anume introducerea notiunii **tip de document**, care descrie structura pe care trebuia sa o aiba un anumit document XML pentru a fi valid. Programele care prelucraza aceste documente stiu dinainte structura lor si le pot analiza si accesa fara probleme. Deci, raspunsul la întrebarea "De ce XML?" este: "XML, pentru ca reprezinta un format standard international de stocare a datelor, sprijinit de cele mai mari corporatii din domeniul tehnologiei informatiei, care permite schimbul de date între orice pro-grame de pe orice platforme."

Adoptarea unui nou limbaj este mult mai simpla atunci când cunoastem alte limbaje anterioare acestuia, cu care suntem familiarizati. Toti utilizatorii de HTML vor putea aborda mai usor limbajul XML datorita asemanarilor de forma dintre cele doua. Însa, asemanarile se opresc aici. Este gresit sa credem ca XML este un urmas al HTML. În timp ce HTML este utilizat pentru afisarea datelor, XML este utilizat la descrierea datelor. XML este mult mai flexibil decât HTML, utilizatorii putându-si defini noile marcaje pentru o reprezentare cât mai naturala a datelor din documente. XML utilizeaza notiunea *tip de document* cu care se poate valida sau invalida un document, în timp ce la HTML aceasta notiune lipsind cu desavârsire. Cu toate acestea, HTML si XML vor continua sa existe si sa conlucreze, în sensul ca *datele vor fi descrise si stocate utilizând XML si vor fi afisate utilizând HTML*.

### Cum?

Pentru a înțelege cum se utilizeaza XML trebuie studiata structura XML. Una din cele mai importante calitati ale XML este

posibilitatea de a structura documentele. Orice document XML include o structura logica si o structura fizica. Structura logica se aseamana cu un sablon care ne arata ce elemente trebuie sa includa documentul si în ce ordine.

### Structura logica a documentelor XML

**Prologul** este primul element structural într-un document XML, optional. Prologul consta din doua componente de baza, de asemenea optionale: *declaratia XML* si *declaratia tipului de document*.

Declaratia XML identifica versiunea specificatiei XML cu care documentul va fi în conformitate (versiunea analizorului care va parcurge documentul XML). De exemplu, declaratia arata astfel:

```
<?xml version="1.0"?>
```

O declaratie XML poate contine de asemenea, specificarea setului de caractere utilizat în documente.

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

În acest caz se utilizeaza setul de caractere UTF-8.

Declaratia tipului de document este alcatuita din meta-codul care indica regulile gramaticale ale DTD (Document Type Definition) pentru o anumita clasa de documente. Toate documentele din aceasta clasa, pentru a fi valide, trebuie sa respecte regulile impuse în definirea tipului de document. Tipul de document poate, de asemenea, sa fie precizat într-un fisier extern care contine toate partile DTD. Declaratia tipului de document trebuie sa apara dupa declaratia XML si înaintea elementului document. De exemplu:

```
<?xml version="1.0"?>
<!DOCTYPE Cont SYSTEM "cont.dtd">
```

Liniile de cod spun procesorului XML ca documentul este din clasa *Cont* si se conformeaza regulilor stabilite în fisierul *Cont.dtd*.

**Elementul document** contine toate datele dintr-un document XML. Este similar cu drive-ul C: de pe PC. Acest element poate cuprinde oricâte sub-elemente imbricate. De exemplu:

```
<CONT>
  <NR_CONT>110002222</NR_CONT>
  <TITULAR>POP ION</TITULAR>
```

```
<SOLD>100.000.000</SOLD>
</CONT>
```

**Imbricarea** este procesul de includere a unui obiect în interiorul altuia. De exemplu, un document XML poate contine ca elemente imbricate alte documente. Relatia de imbricare este de tip parinte/copil. Structura se poate reprezenta astfel:

```
<DOCUMENT>
  <PARENT>
    <CHILD1></CHILD1>
    <CHILD2></CHILD2>
  </PARENT>
</DOCUMENT>
```

### Structura fizica a documentelor XML

Aceasta este compusa din continutul documentului. Unitatile de stocare din documentul XML se numesc *entitati* si pot fi parte a documentului sau externe acestuia. Fiecare entitate este identificata unic de un nume si poate fi compusa din cel putin un caracter ajungând pâna la un fisier extern documentului. Entitatile sunt declarate în prolog si sunt referite în elementul document. Dupa declarare, o entitate poate fi utilizata oriunde în document. Prin declararea unei entitati se indica procesorului XML, care poate fi stocat în interiorul acesteia. La referirea unei entitati, procesorul XML regaseste continutul entitatii, asa cum a fost declarat, si sa îl substituie în document.

**Entitatea analizabila**, câteodata numita si entitate text, contine date de tip text si devine parte a documentului XML dupa procesarea acestuia. **Entitatea neanalizabila** este un container (fisier) al carui continut poate sau nu sa fie text. Daca este text, atunci acesta nu este analizabil XML. O entitatea analizabila este scrisa astfel încât sa poata fi citita de procesorul XML si sa i se extraga continutul. Toate referintele entitatii vor fi înlocuite cu continutul acesteia. De exemplu, entitatea pentru soldul contului poate fi:

```
<!ENTITY SC "Soldul Contului">
```

**Referinta entitatii** actioneaza ca un marcaj care se va înlocui cu continutul ei. Inserarea unei referinte se face astfel: `&nume_entitate;`. De exemplu, referirea entitatii SC arata astfel:

```
<VSOLD> &SC; 110002222 </VSOLD>
```

**Entitatile parametru** au aceleasi reguli ca si entitatile analizabile. La referire, simbolul `&` se înlocuieste cu simbolul `%`.

De multe ori, o entitate neanalizabila este numita si entitate binara, deoarece continutul sau este de obicei un fisier binar care nu este interpretat direct de procesorul XML. De asemenea, o entitate neanalizabila poate contine text. De exemplu, declararea unei entitati neanalizabile poate fi: `<!ENTITY ImagineaMea SYSTEM "Imagine.gif" NDATA GIF>`.

Declararea entitatii se citește: Entitatea *ImagineaMea* este un fisier binar în notatia GIF. Notatia GIF indica procesorului XML care utilitar sa foloseasca pentru afisarea imaginii. În cazul nostru, notatia este: `<!NOTATION GIF SYSTEM "\"Utils\Gifview.exe">`

În XML, anumite caractere sunt utilizate pentru marcasele din document. De exemplu, parantezele unghiulare (`<>`) si slash-ul (`/`) sunt interpretate ca marcase si nu ca date de tip caracter. Acestea sunt caracterele rezervate. Daca se doreste utilizarea lor ca si date de tip caracter trebuie puse în secventa *escape*. De exemplu pentru a insera textul `<CONT>` în document se scrie codul: `&lt;CONT&lg;`

Lista cu entitatile predefinite contine: `&lt;` - `<` (opening angle bracket); `&lg;` - `>` (closing angle bracket); `&amp;` - `&` (ampersand); `&apos;` - ``` (apostrophe); `&quot;` - `"` (double quotation mark).

O **entitate este interna** daca nu exista unitate de stocare fizica distincta de corpul entitatii. De exemplu: `<!ENTITY SC "Soldul Contului">`

O **entitate este externa** daca unitatea de stocare fizica este distincta fata de corpul acesteia. Identificatorul unei entitati externe poate fi SYSTEM sau PUBLIC. Identificatorul SYSTEM furnizeaza un pointer catre locatia continutului entitatii, cum ar fi un URI (Uniform Resource Identifier). De exemplu:

```
<!ENTITY MyImg SYSTEM "http:\\
www.numa.com\\img\\img.gif" NDATA
GIF>
```

Identificatorul PUBLIC, în plus fata de identificatorul SYSTEM, cauta fisierul pe statia locala si numai daca nu îl gaseste îl cauta si la locatia publica aflata în retea. De exemplu:

```
<!ENTITY MyImg PUBLIC "-//Informa-  
tii\Imagini\standard\EN"  
"http:\www.nume.com\img\image.gif"  
NDATA GIF>
```

Cea mai buna modalitate de a înțelege DTD (Document Type Definition) este printr-un exemplu. Mai jos se regaseste continutul fisierului Cont.xml.

```
<?xml version="1.0"?>  
<!DOCTYPE CONT [  
  <ELEMENT CONT (NR_CONT,  
    TITULAR, SOLD)>  
  <ELEMENT NR_CONT (#PCDATA)>  
  <ELEMENT TITULAR (#PCDATA)>  
  <ELEMENT SOLD (#PCDATA)>  
>  
<CONT>  
  <NR_CONT>110002222</NR_CONT>  
  <TITULAR>POP ION</TITULAR>  
  <SOLD>100.000.000</SOLD>  
</CONT>
```

Se observa ca documentul are doua parti, prima pentru definirea regulilor gramaticale ale DTD, iar a doua pentru implementarea unui element care respecta aceste reguli. Exista doua dezavantaje majore în utilizarea DTD: regulile nu sunt scrise în XML si, de asemenea, nu permite declararea tuturor tipurilor de date (întregi, date calendaristice etc.). De aceea, corporatia Microsoft a propus consorțiului W3C introducerea schemelor. Astfel, DTD-ul de mai înainte se transforma în schema:

```
<Schema ID="CONT">  
  <Element name = "NR_CONT" type = "string"/>  
  <Element name = "TITULAR" type = "string"/>  
  <Element name = "SOLD" type = "string"/>  
</Schema>
```

### Unde?

Pentru a putea vizualiza datele cu un browser de tip Internet Explorer 5.0, într-un format elegant, se utilizeaza limbajul HTML. În exemplul urmator sunt afisate datele despre contul stocat în documentul Cont.xml prezentat anterior. În continuare, este prezentat fisierului Cont.htm, care

contine codul care realizeaza efectiv afisarea datelor în browser.

```
<!DOCTYPE HTML PUBLIC  
"-//W3C//DTD HTML 3.2 Final//EN">  
<HTML>  
  <HEAD>  
    <COMMENT> Functia care trateaza me-  
sajul raspunzator de încarcarea documen-  
tului Cont.xml este loadDoc().  
  </COMMENT>  
  <SCRIPT LANGUAGE = "JavaScript"  
    FOR = window EVENT = onload>  
    loadDoc()  
  </SCRIPT>  
  <COMMENT>
```

În scripul Java de mai jos se declara si se aloca un obiect ActiveX de tip microsoft.xmlDOM. Cu ajutorul obiectului se încarca documentul Cont.xml. Daca s-a reusit deschiderea cu succes se lanseaza functia start() care parcurge arborele care stocheaza elementul document.

```
CONT  
  |-- NR_CONT = 110002222  
  |-- TITULAR = POP ION  
  |-- SOLD = 100.000.000  
</COMMENT>  
<SCRIPT LANGUAGE = "JavaScript">  
  var xmlDoc = new ActiveXObject ("micro-  
soft.xmlDOM");  
  xmlDoc.load("Cont.xml");  
  function loadDoc()  
  {  
    if (xmlDoc.readyState == "4")  
      start();  
    else  
      window.setTimeout("loadDoc()", 4000);  
  }  
  function start()  
  {  
    /*se pastreaza radacina arborelui*/  
    var rootElem = xmlDoc.documentElement;  
    /*se extrage informatia din primul nod copil  
al radacinii - numarul contului*/  
    nr_contdata.innerText = rootElem.childNo-  
des.item(0).text;  
    /*se extrage informatia din al doilea nod copil  
al radacinii - titularul*/  
    titulardata.innerText = rootElem.childNo-  
des.item(1).text;  
    /*se extrage informatia din al treilea nod copil  
al radacinii - soldul*/  
    solddata.innerText = rootElem.childNo-  
des.item(2).text;  
  }  
</SCRIPT>  
<TITLE>Afisare cont</TITLE>  
<H1>Afisare cont</H1>
```

```

</HEAD>
<BODY>
  <DIV ID = "nr_cont" STYLE = "font
weight:bold; font-size:16">
    <COMMENT>Se afiseaza numarul contului
    </COMMENT>
    Numarul contului:
    <SPAN ID = "nr_contdata" STYLE = "font-
weight:normal"> </SPAN>
  </DIV>
  <BR>
  <DIV ID = "titular" STYLE = "font weight:bold;
font-size:16">
    <COMMENT>Se afiseaza titularul contului
    </COMMENT>
    Titularul contului:
    <SPAN ID = "titulardata" STYLE =
    "font-weight:normal"> </SPAN>
  </DIV>
  <BR>
  <DIV ID = "sold" STYLE = "font
weight:bold;font-size:16">

```

```

<COMMENT>Se afiseaza soldul contului
</COMMENT>
Soldul contului:
  <SPAN ID = "solddata" STYLE =
  "font-weight:normal"></SPAN>
</DIV>
<BR>
</BODY>
</HTML>

```

### Bibliografie

- [1] Neil Bradley: *The XML companion* - Addison-Wesley, 1998
- [2] William J. Pardi: *XML in Action. Web Technology* - Microsoft Press, 1999
- [3] Revista *ORACLE* - numerele din februarie/martie si mai/iunie 2000.