Interacțiunea Om-Calculator

Laborator 1

XML

Cuprins

1 Obiective laborator

- Familiarizarea cu standardul XML si utilitatea acestuia

2 Despre XML – Scurta istorie

XML – (eXtensible Markup Language) este un limbaj de adnotare/structurare a datelor. Istoria aparitiei sale porneste de la SGML (Standard Generalized Markup Language) aparut in anii ‘80. Acest limbaj isi propunea la aparitie sa creeze documente ce pot fi analizate de catre masini prin introducerea de marcaje (sau taguri). SGML nu a fost un succes deoarece era foarte complex si astfel crearea de programe care sa-l analizeze.

In anii 90 la CERN a aparut HTML-ul (HyperText Markup Language). La baza acestui limbaj a fost SGML-ul si scopul sau a fost marcarea documentelor astfel incat sa poata fi transmise prin retea. Unul din primele documente care au stat la baza WWW-ului si respectiv HTML-ului le puteti gasi la [1]. Una din ideile acestui document este ca browserele trebuie sa ignore marcajele si atributele pe care nu le inteleg. Aceasta idee a avut ca efect din cate cred eu aparitia unui limbaj mai simplu care sa poata fi parsat rapid de catre browsere. Si astfel s-a dezvoltat HTML-ul.

Succesul avut de Web a cauzat o dezvoltare rapida a browserelor care analizau marcajele HTML. Astfel au aparut o gama larga de marcaje si atribute care puteau fi scrise fara prea multe constrangeri. Browserele au trebuit sa tina cont de aceste constrangeri si au devenit foarte complexe. Pe de alta parte s-a sesizat utilizarea HTML-ului pentru adnotarea documentelor si o slabiciune de-a sa – faptul ca nu se pot adauga marcaje noi.

In acest context in decembrie 1997 W3C (World Wide Web Consortium [2]) a lansat ca recomandare propusa versiunea 1.0 de XML [3]. In acest document se afirma ca XML este un subset al SGML care pastreaza caracteristici cum ar fi posibilitatea crearii de marcaje, posibilitatea de validare, posibilitatea de a fi citit si inteles de catre oameni. Totodata se afirma ca XML este creat pentru a fi folosit pentru adnotari, schimb de date, publicare documente si prelucreare automata de catre clienti (agenti) inteligenti. Pastrand un subset al SGML impreuna cu un set de constrangeri documentele XML vor putea fi prelucrate foarte rapid de catre parsere neavand limitele date de complexitatea SGML. Vom vedea care sunt cazurile de utilizare actuale pentru XML dupa ce ii vom intelege sintaxa – dupa care ii vom intelege mai clar si avantajele si slabiciunile.

3 Sintaxa

Am vorbit in prima parte a acestei prezentari despre marcaje si atribute. Acum putem sa incercam sa le si definim. Pentru a ilustra sintaxa vom folosi exemplul din Anexa 1. Un marcaj este un text care descrie sensul sau structura unor date. Acest text este inserat intre semnele „<” si „>”. Marcajele pot avea inceput si sfarsit ca de exemplu:

```
<nume>Popescu</nume> (a)
```

sau pot fi vide (fara text in interiorul marcajului)

```
<casatorit/>
```

Marcajele sunt case-sensitive. La intalnirea unui marcaj de tipul <nume>Popescu</NUME> parserul va da o eroare.

Un element este tot ceea ce se afla intre marcajul de inceput si marcajul de sfarsit . (a) este un element. Numele elementului este „nume”. Aceste nume trebuie sa indeplineasca cateva conditii [4].

- Pot sa contina litere, cifre si alte caractere
- Nu pot incepe cu cifre si cu semne de punctuatie
- Nu pot incepe cu „xml”
- Nu pot contine spatii

Un element poate contine mai multe elemente. Acest lucru determina aparitia unor relatii de rudenie intre elemente si a unor constrangeri suplimentare. Cea mai importanta dintre aceste constrangeri este aceea ca intr-un fisier XML trebuie sa existe un element radacina care sa le

contina pe toate celelalte. Elementul radacina in Anexa 1 este <persoana>. O a doua constrangere este ca elementele nu pot fi incrucisate. Astfel urmatoarea constructie este incorecta <tranzactie><data>2005-11-05</tranzactie></data>. Cu alte cuvinte marcajele trebuie inchise invers fata de cum sunt deschise.

Relatiile de rudenie pot fi parinte-copil sau frate-frate. Un exemplu de relatie parinte-copil in cadrul unui element este urmatoarea.

```
<tranzactie>
      <data>20-07-2005</data>
      <ora>13:00:00</ora>
</tranzactie>
```

In acest exemplu elementul <*tranzactie*> este parinte al elementelor <*data*> si <*ora*> care sunt "frati".

Un element poate sa contina unul sau mai multe atribute. Atributele se specifica in cadrul marcajului de inceput al elementului si sunt de tip nume_atribut="valoare". Un exemplu in Anexa 1 este in cadrul marcajului urmator: <*suma moneda="RON"*> *1000* <*/suma*>. Atributul in acest caz este moneda iar valoarea atributului este RON. Valorile atributelor se pun intotdeauna intre ghilimele.

Ar mai fi de mentionat ca intotdeauna un document XML incepe cu un rand in care se mentioneaza versiunea limbajului si codificarea folosita. In exemplul din Anexa 1 se specifica faptul ca se foloseste versiunea 1.0 si ca se foloseste codificarea ISO-8859-2 care este codificarea folosita pentru limba romana.

Respectand aceste simple reguli de sintaxa vom putea obtine ceea ce se numeste un document XML **bine format.** Acest document se valideaza la randul lui cu ajutorul unui alt fisier in care se definesc tagurile folosite. Acest alt fisier poate fi de tip **DTD** sau **XML Schema** si va fi prezentat in laboratoarele viitoare.

Astfel, un document **XML "bine formatat"** este un document ce foloseste o sintaxa corecta si anume:

- Documentele XML trebuie sa aiba un element radacina (root)
- Elementele XML trebuie sa aiba tag-uri de sfarsit
- Tag-urile XML sunt sensibile litere mari/litere mici
- Elementele XML trebuie sa fie corect imbricate
- Atributele XML trebuie sa fie incluse intre ghilimele

Un document **XML valid** este un document "bine formatat" care se supune si regulilor DTD (Document Type Definition) sau XML Schema.

4 Cazuri de utilizare

Din introducere si din scurta trecere in revista a sintaxei am putut observa ca XML-ul este un limbaj care permite structurarea si adnotarea datelor intr-un mod lizibil atat de catre oameni cat si de catre calculatoare. Avand aceste calitati putem oarecum deduce unde se poate folosi in mod practic si unde este mai putin recomandat.

Poate cel mai evident caz de utilizare este cel al fisierelor de configurare. In momentul de fata foarte multe aplicatii isi pastreaza fisierele de configurare in XML. Motivele sunt cele enuntate mai sus – **parsare rapida pentru sisteme automate, lizibilitate pentru oameni si structura logica a datelor**.

Un alt caz in care se foloseste XML si in care se pare ca va deveni un standard este cel al stocarii informatiilor din fisiere tip office. Initiativa a apartinut dezvoltatorilor de solutii office open-source care au standardizat un format numit Open Document[5][6]. Acest format presupune salvarea oricarui document de tip office – document, foaie de calcul (spreadsheet), prezentare – intr-un format XML. Informatia este stocata in mai multe fisiere care sunt arhivate si astfel utilizatorul poate vedea numai un singur fisier de tip *.sxw sau *.odt, etc. Incepand cu Suita Office 2007 si Microsoft a abordat o arhitectura bazata pe XML in reprezentarea interna a fisierelor (in Anexa 2 este prezentat un fisier .docx in care este scris textul „Hello World").

De ce s-a alex XML pentru un astfel de format? Pentru ca ofera avantajele structurarii informatiei si mai ales pentru ca s-au dezvoltat deja foarte multe alte limbaje si standarde pe baza lui. Aceste limbaje (XHTML, SVG, XSL, SMIL, XLink, XForms, MathML, Dublin Core) sunt folosite si ele in acest format si deoarece toate au la baza XML interoperabilitatea este garantata. XML are si un dezavantaj care este dat de faptul ca introduce informatie redundanta (overhead). Informatia redundanta este reprezentata de numele marcajelor care se repeta in document. Aceasta problema a fost rezolvata de cei de la Open Office prin arhivarea fisierelor XML intr-o arhiva zip. Astfel s-a rezolvat problema spatiului suplimentar aparut prin introducerea unei mici pierderi in viteza.

Un alt caz de utilizare este in stocarea efectiva a datelor – mai precis baze de date. In momentul de fata XML poate fi folosit cu succes pentru baze de date mici. In momentul in care bazele de date cresc ca dimensiune overheadul devine inacceptabil. Din cauza spatiului mare ocupat si a necesitatii accesului foarte rapid la date arhivarea fisierelor XML cu date nu poate fi o solutie si in acest caz. Oricum, multe interfete pentru baze de date permit exportarea bazei de date in format XML – atat definitia tabelelor si a relatiilor intre ele cat si datele propriu zise. Doua exemple de aplicatii care au aceasta facilitate sunt Microsoft Access si PHPMyAdmin[7]. Un alt exemplu de caz in care XML-ul nu a avut succes este legat tot de stocarea unor cantitati mari de date si de acces rapid la acestea – mai precis pastrarea fisierelor jurnal ale serverelor. In acest domeniu este folosit printre altele un standard numit CLF – Common Logfile Format – care are avantajul ca nu introduce informatie suplimentara [8].

XML sta la baza multor tehnologii si limbaje noi. Astfel pentru prezentare de continut pe web exista XHTML si WML (pentru dispozitive mobile), pentru descrierea unor fisiere XML exista XSchema, pentru transformarea fisierelor XML pentru a fi reprezentate exista XSL, pentru realizarea unor prezentari exista SMIL, pentru descrierea obiectelor grafice exista SVG, pentru reprezentarea semanticii unor domenii exista RDF si/sau OWL, pentru schimb de informatii intre aplicatii exista SOAP si asa mai departe, lista aceasta fiind departe de a cuprinde toate tehnologiile/limbajele existente bazate pe XML. De ce au aparut atatea limbaje bazate pe XML? Pentru ca are o structura simpla si totusi stricta, pentru ca este foarte usor de analizat si foarte usor de definit si dezvoltat.

5 Tooluri

Există o largă varietate de instrumente pentru editarea fişierelor XML. Din seria celor comerciale cele mai cunoscute sunt Altova (http://www.altova.com/xml-editor/), Oxygen XML (http://www.oxygenxml.com/) şi StylusStudio (http://www.stylusstudio.com/xml/editor/). Ambele oferă versiuni de „trial" pentru a vedea ce facilităţi oferă.

Datorită simplităţii sintaxei o gamă largă de editoare au facilităţi pentru crearea fişierelor XML: Notepad++ (http://notepad-plus.sourceforge.net), din gama editoarelor mai simple, şi Eclipse sau Netbeans din gama editoarelor mai complexe.

6 Şabloane de proiectare a fişierelor XML

Site-ul XMLPatterns.com prezintă un mare număr de şabloane de construcţie a fişierelor XML. Folosirea acestor şabloane permite crearea unor documente mai uşor de citit, folosit şi prelucrat, permite gestionarea mai facilă a informaţiilor oferite.

În această secţiune vom rezuma şabloanele cele mai importante.

6.1 Şabloane pentru folosirea Metadatelor (date despre date)

Când există metadate într-un document pot fi folosite mai multe soluţii pentru integrarea lor:

- **Head + Body** – varianta folosita şi în HTML – datele despre date se pun într-o secţiune specială a documentului numită head şi datele efective se vor plasa în secţiunea body.
- **Metadatele se pun înaintea datelor în document**. Motivul este destul de simplu şi anume în condiţiile în care procesarea XML-ului se face secvenţial s-ar putea ca un proces, după ce interpretează meta-datele să nu mai dorească sa prelucreze şi restul documentului şi implicit să nu-l mai încarce sau parseze
- În cazul în care există un mare număr de **metadate**, poate fi util ca acestea să fie **plasate într-un fişier separat**. Acest lucru este recomandabil mai ales în cazul în care este vorba de metadate comune mai multor fişiere (vezi ontologii şi RDF)

6.2 Abstractizare

Cum se pot alege şi respectiv grupa elementele astfel încât structura XML-ului format să fie cât mai inteligibilă.

- **Containere** – în containere se pot pune mai multe elemente diferite dar care caracterizează un „părinte" comun dintr-un anumit punct de vedere. De exemplu pentru un om putem pune într-un container „date de identificare" elementele „nume", „prenume", „CNP"
- **Colecţii** – când în cadrul unui element pot apărea ca şi copii mai multe elemente de acelaşi tip acestea este recomandabil să se grupeze ca şi copii al unui element separat. Exemplu un om are mai mulţi copii ceea ce putem exprima în mai multe feluri:

```
<om>
      <date_identificare>…</date_identificare>
      <copil></copil>
```

```
        <copil></copil>
        …
</om>
```

Sau

```
<om>
        <date_identificare>…</date_identificare>
        <copii>
                <copil></copil>
                <copil></copil>...
        </copii>
…
</om>
```

A doua variantă presupune adăugarea elementului „copii” care reprezintă o colecţie de elemente de tip „copil”

- **Elementele este bine să aibă o corespondenţă în** cadrul **domeniului descris** (v. Descrierea claselor şi a proprietăţilor la programarea orientată obiect)

6.3 *Organizarea datelor*

- **Declararea elementelor înainte de a fi referite**. Cu ajutorul atributelor de tip IDREF elemente pot fi referite în cadrul altor elemente. Este recomandabil ca întâi să apară în document elementul şi apoi referinţa (tot pentru uşurinţa procesării)
- **Referirea datelor şi nu repetarea lor**. Datele pot fi declarate folosind declaraţii de tip Entity sau pot fi identificate în mod unic folosind atribute de tip ID şi referite prin atribute de tip IDREF
- **Clasificare folosind atribute** – când pot clasifica anumite elemente în mai multe feluri se pot introduce atribute pentru a nu complica ierarhia. În exemplul de mai sus elementului copil i se poate introduce atributul „sex” care practic introduce o clasificare suplimentară fără a adăuga elemente suplimentare.

6.4 *Altele*

- **Reutilizare atribute** – dacă mai multe elemente au aceleaşi atribute nu trebuie definite atributele de mai multe ori
- **Nume scurte dar inteligibile** – „loc_de_munca” este preferat elementului “serv”.

Pentru mai multe detalii şi pentru mai multe şabloane vizitaţi http://xmlpattern.com

7 Concluzii

XML-ul este un limbaj cu o sintaxa simpla si care permite doar structurarea datelor intr-o maniera proprie prin definirea propriilor taguri. Aceasta facilitate de structurare a datelor a permis folosirea sa pentru a dezvolta limbaje noi precum si pentru a fi folosit in noi standarde de stocare a datelor. Nu in ultimul rand XML-ul poate fi folosit pentru a schimba date intre aplicatiile care au nevoie de a comunica intr-un limbaj comun.

Chiar si in ciuda dezavantajelor pe care le are (overhead) calitatile XML mai sus mentionate ii permit sa fie folosit cu succes in multe domenii si sa fie piatra de temelie pentru si mai multe domenii.

8 Referinte

[1] „The first version of HTML" - http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/MarkUp.html

[2] World Wide Web Consortium – http://www.w3c.org

[3] Press Release XML 1.0 - http://www.w3.org/Press/XML-PR

[4] Tutorial XML – http://www.w3schools.com/xml/xml_elements.asp

[5] Standardizarea Open Document – siteul web OASIS - http://www.oasis-open.org/news/oasis_news_05_23_05.php

[6] Oasis OpenDocument Essentials - http://books.evc-cit.info/odbook/book.html

[7] PHPMyAdmin – web site - http://www.phpmyadmin.net/home_page/index.php

[8] CLF – formatul de inregistrare al jurnalelor - http://www.w3.org/Daemon/User/Config/Logging.html#common_logfile_format

[9] Şabloane XML – www.XMLPattern.com

Anexa 1 – Exemplu de document XML

```
<?xml version="1.0" encoding="ISO-8859-2"?>
      <persoana>
      <nume>Popescu</nume>
      <prenume>Ion</prenume>
      <informatii_personale>
            <oras>Bucuresti</oras>
            <strada>Calea Mosilor</strada>
            <nr>100</nr>
            <tel_fix>0213003000</tel_fix>
            <tel_mobil></tel_mobil>
            <e-mail>popescu.ion@gmail.com</e-mail>
            <casatorit />
      </informatii_personale>
      <tranzactii>
            <tranzactie>
                  <data>20-07-2005</data>
                  <ora>13:00:00</ora>
                  <suma moneda="RON">1000</suma>
                  <directia>venit</directia>
                  <descriere>salariu</descriere>
            </tranzactie>
            <tranzactie>
                  <data>20-07-2005</data>
                  <ora>13:05:00</ora>
                  <suma moneda="RON">35</suma>
                  <directia>plata</directia>
                  <descriere>telefon</descriere>
            </tranzactie>
      </tranzactii>
</persoana>
```

Anexa 2 Fisierul document.xml pentru un document Microsoft Office .docx in care apare textul „Hello World”

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<w:document
xmlns:wpc=http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas
xmlns:mo=http://schemas.microsoft.com/office/mac/office/2008/main
xmlns:mc=http://schemas.openxmlformats.org/markup-compatibility/2006
xmlns:mv="urn:schemas-microsoft-com:mac:vml"
xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:r=http://schemas.openxmlformats.org/officeDocument/2006/relationships
xmlns:m=http://schemas.openxmlformats.org/officeDocument/2006/math
xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:wp14=http://schemas.microsoft.com/office/word/2010/wordprocessingDrawing
xmlns:wp=http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:w=http://schemas.openxmlformats.org/wordprocessingml/2006/main
xmlns:w14=http://schemas.microsoft.com/office/word/2010/wordml
xmlns:wpg=http://schemas.microsoft.com/office/word/2010/wordprocessingGroup
xmlns:wpi=http://schemas.microsoft.com/office/word/2010/wordprocessingInk
xmlns:wne=http://schemas.microsoft.com/office/word/2006/wordml
xmlns:wps=http://schemas.microsoft.com/office/word/2010/wordprocessingShape
mc:Ignorable="w14wp14">
      <w:body>
            <w:p w:rsidR="00007A72" w:rsidRDefault="003C523F">
                  <w:r>
                        <w:t>Hello World</w:t>
                  </w:r>
                  <w:bookmarkStart w:id="0" w:name="_GoBack"/>
                  <w:bookmarkEnd w:id="0"/>
            </w:p>
            <w:sectPr w:rsidR="00007A72" w:rsidRDEFAULT="006575D2">
                  <w:pgSz w:w="11900" w:h="16840"/>
                  <w:pgMar  w:top="1440"  w:right="1800"  w:bottom="1440"
                  w:left="1800"       w:header="708"       w:footer="708"
                  w:gutter="0"/>
                  <w:cols w:space="708"/>
                  <w:docGrid w:linePitch="360"/>
            </w:sectPr>
      </w:body>
</w:document>
```