Vizualizarea documentelor xml

- **Fără un fişier de stil asociat**: browserul vizualizează conţinutul documentului xml, cu posibilitatea de a vedea/ascunde descendenţii unui nod din structura arborescentă

  Exemplu: **fisiere/xml-xsl/bib.xml**
- **Cu un fişier de stiluri** (construit asemănător ca pentru un document html): se foloseşte acest fişier la vizualizarea documentului. Pentru fiecare tag din documentul xml **se poate** defini un **stil de apariţie**. La vizualizarea documentului tagurile vor apare în ordinea în care sunt trecute în document, fără posibilitatea de a schimba ordinea acestora sau de a însera caractere suplimentare. Utilizarea acestei facilităţi se face prin includerea următoarei declaraţii în documentul xml (se precizează fişierul de stiluri):

  ```
  <?xml-stylesheet type="text/css" href="fisier.css" ?>
  ```

  Exemplu: **bib.xml** cu fişierul de stiluri **bib.css** din directorul: /**xml-xsl**/**0**/

Observaţii:
- La vizualizare nu se pot folosi atributele definite în taguri.
- Un document xml se poate transforma în alt document (xml, html, etc.) cu ajutorul limbajul XSLT.

XSL - eXtensible Stylesheet Language

Pentru a prezenta (vizualiza, transforma) documentele xml s-a creat **limbajul declarativ XSL**. Acesta este compus din trei componente:
- **XPath** (limbajul XML Path)
- **XSLT** (eXtensible Stylesheet Language Transformation),
- **XSL/FO** (eXtensible Stylesheet Language Formatting Objects).

XPath

- XPath este un **standard** W3C
- XPath permite selectarea unei părţi dintr-un document xml cu ajutorul unor **expresii** (expresii de cale)
- XPath conţine mai multe **funcţii standard** (descrise la **http://www.w3.org/2005/xpath-functions/ şi http://www.w3schools.com/xpath/xpath_functions.asp**)
- In XPath se pot folosi **7 tipuri de informaţii** (elemente):
  1. *element*,
  2. *atribut*,
  3. *text*,
  4. *namespace*,
  5. *processing-instruction*,
  6. *comment*,
  7. *document (root) node*.
- **Valoarea atomică** este valoarea unui atribut sau a unui nod care nu se mai divide.
- Un element poate avea **0, 1, 2, ... fii** (descendenţi pe nivelul imediat următor în structura de arbore).
- Un element poate avea un nod **părinte**
- Precizarea unui element sau a unei mulţimi de elemente se poate face printr-o "**expresie**", care este scrisă ca o **succesiune de paşi** pentru a ajunge **dintr-un nod curent la alt nod** sau la altă mulţime de noduri.

- Intr-un pas (din această succesiune de selectare) poate apare una din următoarele construcţii. In funcţie de context, **există un nod curent**.

nume_element	Selectează toţi descendenţii direcţi ai unui element cu numele precizat.
/	Selectează nivelul următor din structura de arbore. Dacă apare ca prim caracter în expresie, atunci se precizează nodul rădăcină din document
//	Selectează un nod, plecând de la nodul curent, făcând salturi (spre interior) peste 0, 1, 2, ... nivele.
.	Selectează nodul curent
..	Selectează nodul părinte al nodului curent
@	Selectează un atribut din nodul curent

Exemple pentru documentul:

```
<?xml version="1.0" ?>
<bibliografie>
  <carte limba="ro">
    <titlu>PHP, MySQL si Apache</titlu>
    <autor>Julie C. Meloni</autor>
    <editura>Teora</editura>
    <pret moneda="RON">45.87</pret>
    <anap>2005</anap>
  </carte>
  <carte limba="en">
    <autor>Richard Anderson</autor>
    <autor>Brian Francis</autor>
    <autor>Alex Homer</autor>
    <autor>Rob Howard</autor>
    <autor>David Sussman</autor>
    <autor>Karli Watson</autor>
    <titlu>Professional ASP.NET 1.0 Special Edition</titlu>
    <editura>Wrox Press Ltd.</editura>
    <anap>2002</anap>
  </carte>
</bibliografie>
```

carte	Selectează toţi descendenţii direcţi (copiii nodului curent) cu numele *carte*
/bibliografie	Selectează elementul *bibliografie* din document
carte/titlu	Selectează toate elementele *titlu*, care sunt descendenţi direcţi pentru *carte*
//editura	Selectează elementele *editura* incluse în nodul curent, indiferent de poziţia lor în structura arborescentă
bibliografie//anap	Selectează elementele *anap* incluse, indiferent de distanţă, într-un element *bibliografie*
//@limba	Selectează toate atributele cu numele *limba*

- Prin construcţiile precedente se poate selecta o **mulţime** de noduri (o colecţie, o listă). Pentru a specifica un anumit nod (dintr-o colecţie de noduri, sau care îndeplineşte o condiţie), se foloseşte un **predicat**, precizat între două paranteze drepte. Predicatul poate fi o ***expresie numerică*** (precizează printr-un index un anumit element din colecţie), sau poate fi o ***condiţie*** (se poate preciza o submulţime de elemente din colecţie, care îndeplinesc o condiţie). In condiţia din predicat (dacă apare) se pot folosi următorii operatori (aritmetici, relaţionali, logici): +, -, *, div, =, !=, <, <=, >, >=, or, and, mod.
- Pentru a specifica un număr neprecizat de elemente (în expresie, în predicat) se pot folosi construcţiile următoare:

*	pentru orice element
@*	pentru orice atribut
node()	pentru orice nod

Exemple:

Expresie	**Rezultat**
/bibliografie/carte[1]	Selectează primul element *carte* din elementul *bibliografie*. **Observaţie:** IE foloseste pentru primul element indexul 0, iar în standard primul index este 1.
/bibliografie/carte[last()]	Selectează *ultimul* element *carte* din elementul *bibliografie*
/bibliografie/carte[last()-1]	Selectează *penultimul* element *carte* din elementul *bibliografie*
/bibliografie/carte[position()<3]	Selectează primele două elemente *carte* din elementul *bibliografie*
//carte[@limba]	Selectează toate elementele *carte* care au atributul *limba*
//carte[@limba='en']	Selectează toate elementele *carte* care pentru care atributul *limba* are valoarea 'en'
/bibliografie/carte[@pret>50.00]	Selectează toate elementele *carte* din elementul *bibliografie* pentru care atributul *pret* are valoarea > 50
/bibliografie/carte[@pret>50.00]/titlu	Pentru toate elementele *carte* din elementul *bibliografie* la care atributul *pret* are valoarea > 50, selecteaza titlul
/bibliografie/*	Selectează toate elementele fiu din elementul *bibliografie*
//*	Selectează toate elementele din nodul curent
//titlu[@*]	Selectează toate elementele *titlu* care au cel puţin un atribut

- Cu operatorul "|" se pot preciza mai multe expresii. **Exemple**:

//carte/titlu | //carte/pret	Selectează toate elementele *titlu* şi toate elementele *pret* din elementele *carte*
// titlu | // pret	Selectează toate elementele *titlu* şi toate elementele *pret* din document
/bibliografie/carte/titlu | //pret	Selectează toate elementele *titlu* din toate elementele *carte* din *bibliografie* şi toate elementele *pret*

- In procesul de selectare (prin expresia de cale) se ajunge la un **nod curent**, din care se poate merge într-o anumită "direcţie" (pe o anumită axă) pentru a determina alte noduri. In

tabelul următor se precizează denumirea unor astfel de **axe** pentru un nod curent. In expresia de selectare se poate folosi construcţia: "**DenumireAxa::denumirenod**" în loc de "denumirenod". De aici se deduce că un "pas" din selectarea nodurilor poate avea forma: **DenumireAxa::denumirenod [predicat]**, unde **DenumireAxa** şi **predicat** pot lipsi.

DenumireAxa	Rezultat
ancestor	Selectează toţi *strămoşii* (părinţi, bunici, ...) nodului curent
ancestor-or-self	Selectează toţi *strămoşii* (părinţi, bunici, ...) nodului curent *şi nodul curent*
attribute	Selectează toate *atributele* nodului curent
child	Selectează toţi *copiii* nodului curent
descendant	Selectează toţi *descendenţii* (copii, nepoţi, ...) nodului curent
descendant-or-self	Selectează toţi *descendenţii* (fii, nepoţi, ...) nodului curent *şi nodul curent*
following	Toate nodurile care sunt *succesoare* ale nodului curent, indiferent de nivelul în structura arborescentă
following-sibling	Toate nodurile care sunt *succesoare* ale nodului curent şi care au acelaşi părinte
namespace	Selectează toate nodurile *namespace* din nodul curent
parent	Selectează *părintele* nodului curent
preceding	Toate nodurile care sunt *predecesoare* ale nodului curent, indiferent de nivelul în structura arborescentă
preceding-sibling	Toate nodurile care sunt *predecesoare* ale nodului curent şi care au acelaşi părinte
self	Selectează *nodul curent*

Exemple de expresii pentru un pas din procesul de selectare (considerăm că se lucrează într-un anumit context, deci există un nod curent):

child::*	Selectează toţi copiii nodului curent
child::carte	Pentru nodul curent selectează toţi copiii *carte*
attribute::*	Selectează toate atributele nodului curent
attribute::limba	Selectează atributul *limba* pentru nodul curent
descendent::carte	Selectează toţi descendenţii *carte* pentru nodul curent
ancestor::carte	Selectează toţi ascendenţii *carte* pentru nodul curent
ancestor-or-self::carte	Selectează toţi ascendenţii nodului *carte* şi nodul însuşi pentru nodul curent

XSLT (eXtensible Stylesheet Language Transformation)

Scopul **limbajului XSLT** este de a **transforma un document xml în alt document xml**. O descriere simplă a semnificaţiei unui document XSLT pentru XML este ceva asemănător ca CSS pentru HTML, dar cu mai multe facilităţi. In procesul de transformare se foloseşte **limbajul XPath** pentru **selectarea unui element, sau a unei mulţimi de elemente**, din documentul xml ce se transformă.

Un fişier (foaie de stiluri) XSLT este un fişier XML (bine format) care defineşte un **template** (model) pentru utilizarea datelor (taguri, atribute) dintr-un document xml. Printr-un

astfel de şablon se poate face o **reorganizare**, o gestiune mai complexă a componentelor unui document xml (se pot căuta anumite elemente, se pot efectua transformări asupra elementelor, se pot folosi stiluri de apariţie), deci se poate **crea un nou document xml** cu taguri complet diferite.

- Un **document xslt** este document xml, deci prima linie este de forma:
  <?xml version="1.0" ?>
- **Elementul rădăcină** pentru un **document xlst** este **<xsl:stylesheet>** sau **<xsl:transform>** (care sunt sinonime) şi se declară astfel (pentru a putea fi folosite):

  <xsl:stylesheet xmlns:xsl="http://www.w3c.org/1999/XSL/Transform" version="1.0">

  sau:

  <xsl:transform xmlns:xsl="http://www.w3c.org/1999/XSL/Transform" version="1.0">

  iar **la final** aceste taguri trebuie închise.
  In acest mod se precizează că toate elementele care vor începe cu **xsl:** sunt elementele documentului XSLT, iar acestea sunt conforme cu recomandarea XSLT 1.0 la care se face referire prin spaţiul de nume. In documentul xslt toate elementele care nu au acest prefix (adică "**xsl:**") sunt extrase (copiate) în documentul transformat, ***fără nici o analiză***. De aici derivă semnificaţia de "şablon": tot ce se află în cadrul tagurilor xsl face parte din şablon, restul se extrage după cum apare în document.
- Pentru a asocia un document XSLT (o foaie de stiluri XSLT) la un document XML este necesară următoarea declaraţie la începutul documentului XML:
  <?xml-stylesheet type="text/xsl" href="numefişier.xsl"?>
- Un **şablon** descrie o modalitate de transformare a elementelor din documentul xml şi este definit prin tagul:

  <xsl:template ...> ... </ xsl:template>

  Un document xsl poate avea definite unul sau mai multe şabloane.
- Un şablon trebuie să poată selecta unul sau mai multe elemente din structura de arbore a documentului xml. O astfel de selectare se poate face prin limbajul **XPath** (aşa cum e precizat mai sus, se poate face o deplasare în arborelele corespunzător documentului xml, prin selecţia nodurilor XML care satisfac diverse criterii).

Din cele precizate mai sus se poate trage concluzia că un **document xslt** are o structură de forma:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

.... taguri xsl (şabloane care se analizează), sau de alte taguri (care se extrag fără analiză)

</xsl:stylesheet>
```

Precizarea şabloanelor:

- **<xsl:template match="expresie"> ... </ xsl:template>**
  Se defineşte un **model (şablon) pentru transformare**. Expresia din **match** (precizată în limbajul XPath) defineşte un nod în structura de arbore a documentului xml. In interiorul

acestui tag nodul astfel precizat va fi un **nod curent**. Acest nod se va transforma conform modelului.
Acest tag se poate repeta (pentru fiecare model de transformare).

- **<xsl:value-of select="expresie" />** - folosind nodul curent, se poate determina valoarea unei expresii, iar această valoare **se adaugă în documentul rezultat prin transformare**. Pentru "expresie" se foloseşte o construcţie XPath.
- **<xsl:for-each select="expresie ">... </ xsl:for-each >** - permite parcurgerea tuturor elementelor precizate de o expresie (deci o colecţie de elemente, iar pentru fiecare element se folosesc tagurile incluse în **<xsl:for-each>.**
- Colectia de elemente dintr-un tag **< xsl:for-each>** se poate sorta după valoarea unei expresii precizată prin **<xsl:sort select="expresie"/>.** Pot apare atribute suplimentare pentru a preciza tipul valorilor (ex. data-type="number") şi ordinea de sortare (ex. order="ascending").
- **<xsl:apply-templates [select="expresie"] />** - aplică un şablon la:
  - descendenţii nodului curent dacă nu apare select
  - elementele fiu care sunt precizate prin **expresie** (dacă apare select).

  Pentru fiecare nod din această mulţime se aplică un şablon individual definit cu:
  <xsl:templates select="nume" match="element"> ... **</xsl:templates>**
- Intr-un şablon poate apare tagul:

```
<xsl:if test="expresie">
  ...
  ... extrageri dacă valoarea condiţiei este true...
  ...
</xsl:if>
```

sau un tag de forma următoare:

```
<xsl:choose>
  <xsl:when test="expresie">
    ... extrageri ...
  </xsl:when>
 .....
  <xsl:otherwise>
    ... extrageri....
  </xsl:otherwise>
</xsl:choose>
```