Familia XML

-1-

Ștefan Trăușan-Matu

PARTEA I - Documente XML

- Geneza XML: ipostaze, voci ale unui document, limbaje de adnotare, SGML, HTML,familia XML.
- Caracteristicile XML.
- Structura fizica si logica a documentelor XML.
- Documente XML bine formate.
- Tipizarea documentelor. Documente XML valide.
- Declararea tipurilor de elemente XML.
- Declararea listei de atribute a unui element XML.
- Declararea entitatilor XML.
- Declararea notatiilor XML.
- Scheme XML.
- Spatii de nume XML pentru adnotarea documentelor.

Ipostaze ale unui text. Limbaje asociate

- **Textul brut**, succesiunea de semne (cuvinte şi imagini), independente de forma de reprezentare - **limbaj natural**
- **Textul adnotat**, conform unor anumite limbaje, **HTML, LaTex, SGML sau XML**
- **Textul afişat** pe ecranul calculatorului, de exemplu, de un anumit browser

Ipostaze ale unui text.
Limbaje asociate (cont.)

- **Limbajul în care se face adnotarea** - gramatică formală - **DTD, SGML, XML Schema**
- **Arborele** care reprezintă imbricarea fragmentelor de text
- **Stilul** de afişare, implicit sau explicit - **limbaj de stil** , de exemplu, **HTML, CSS, XSL**
- **Textul tipărit pe hârtie**

Ipostaze ale unui text.
Limbaje asociate (cont.)

- **Structura de pagini** a textului, ancorele şi legăturile (structura de hipertext), **limbaje de legare, HTML, XPointer, XLink**
- **Cunoştinţele** din text **- limbaj de reprezentare a cunostintelor, HTML, RDF, RDF Schema**
- **Scopurile urmărite de autor**
- **Istoricul parcurgerii** hipertextului de către un cititor
- **Efectul** pe care textul îl are asupra cititorului

SGML

Standard **G**eneralized **M**arkup **L**anguage - ISO 8879

Principii:

- permite reprezentarea textelor independent de un anumit sistem de operare, program sau maşină;
- este un metalimbaj, adică un limbaj de descriere formală a unui limbaj de adnotare;
- are un caracter declarativ, adică nu se indică ce se face ci doar se marchează zone de text, prelucrarea putând fi făcută de programe scrise de utilizatori;
- se pot defini tipuri de documente (“Document Type Definition” - DTD);
- asigură independenţa datelor, oferind un mecanism general de substituire de şiruri.

HTML

Derivat din SGML - DTD asociat

Probleme:

- Lipsa de extensibilitate
- Mixtura mai multor perspective (continut, structura, stil. cunostinte)
- Personalizare
- Sprijin limitat pentru semantica

XML

- “e**X**tensible **M**arkup **L**anguage”
- Păstrează circa 80% din SGML
- **Destinat:**
  - scrierii de pagini de web;
  - schimbului de informaţii pe Internet (de exemplu, între baze de date construite în standarde diferite, cu o importanţă covârşitoare pentru afaceri - “B2B - Bussiness to Bussiness”)
  - adnotării documentelor stocate electronic.

HTML

```
<table>
  <tr>
    <td>7654</td>
    <td>MARTIN</td>
    <td>SALESMAN</td>
    <td>1250</td>
  </tr>
  <tr>
    <td>7788</td>
    <td>SCOTT</td>
    <td>ANALYST</td>
    <td>3000</td>
  </tr>
</table>
```

XML

```
<?xml version="1.0"?>
<EmployeeList>
  <Employee>
    <ID>7654</ID>
    <Name>MARTIN</Name>
    <Job>SALESMAN</Job>
    <Salary<1250</Salary>
  </Employee>
  <Employee>
    <ID>7788</ID>
    <Name>SCOTT</Name>
    <Job>ANALYST</Job>
    <Salary>3000</Salary>
  </Employee>
</EmployeeList>
```

XML păstrează avantajele HTML:

- este simplu de utilizat pe Internet;
- documentele XML sunt uşor de creat;
- documentele XML sunt uşor de prelucrat;
- textele pot fi citite relativ uşor şi în forma adnotată, fără a folosi un program special de vizualizare;
- este compatibil cu SGML;

Calităţi proprii XML, pe care HTML nu le are

- Extensibilitate - definirea de noi tipuri de documente (DTD-uri sau scheme)
- Limbaj universal de reprezentare în diverse aplicaţii.
- Problema reprezentării conţinutului documentelor, a semanticii textelor.
- Permite vizualizarea diferită a aceluiaşi document pentru mai mulţi utilizatori.
- Permite transferarea unor prelucrări de la server la utilizator.

Perspective asupra XML

- este un limbaj universal de adnotare a documentelor (derivat din SGML);
- permite declararea unei gramatici pentru un limbaj de adnotări:
  - explicit, printr-un DTD sau o schemă,
  - implicit, pe baza structurii de adnotări, chiar dacă nu există un DTD
- este o modalitate foarte comodă de transfer al informaţiilor pe Internet între aplicaţii de categorii foarte diferite;

Perspective asupra XML (cont.)

- este un limbaj adecvat aplicaţiilor de baze de date federate;
- este o modalitate universală de a reprezenta liniar orice structură, oricât de complexă;
- este un limbaj cu tipuri manifeste, spre deosebire de formatul fix al fişierelor sau bazelor de date;
- este un limbaj de declarare formală a unui vocabular de elemente şi atribute;

Perspective asupra XML (cont.)

- permite restricţionarea şi validarea documentelor care pot fi create;
- este o “ontologie de nivel 0”;
- este un prim pas către facilitarea achiziţiei cunoştinţelor din document;

Structura documentelor XML

Fiecare document XML are o structură:

- fizică
- logică

Structura fizică a documentelor XML

Fizic, documentele XML sunt compuse din unităţi numite **entităţi**.

O entitate are un **nume** şi un **conţinut**.

O entitate poate face referire la alte entităţi pentru a le include în document.

Un document începe printr-o **rădăcină** (“root”) sau **entitate document** (“document entity”).

Entităţi

Din punct de vedere al plasării în fişierul documentului:

- **interne** (în acelaşi fişier)
- **externe** (aflate în alt fişier).

Entităţi (cont.)

- **prelucrate** (analizate, “parsed”) de către procesoarele XML; acestea pot fi atât interne cât şi externe; - conţin **text**, adică o secvenţă de caractere, ce pot reprezenta **adnotări** sau **date de tip caracter** (“caracter data”). Caracterele constituie atomii unui text, conform specificaţiilor standardului ISO/IEC 10646. Caracterele legale sunt tab, “enter” (CR), sfârşit de linie LF şi caracterele legale ale Unicode şi ISO/IEC 10646.
- **neprelucrate** (neanalizate, “unparsed”), aceste entităţi sunt întotdeauna externe.

Entităţi (cont.)

- **generale**, utilizabile fără restricţii;
- **parametru**, utilizabile numai în cadrul unui DTD

Structura logică a documentelor XML

elemente, delimitate în text de **adnotări** (“**markup**”) de început şi de sfârşit.

Adnotările sunt incluse între paranteze unghiulare (delimitate de caracterele “<” şi “>”, care nu pot fi folosite în alt scop în documentele XML.

Dacă este necesară utilizarea “<” şi “>”, se folosesc construcţiile: < şi >, adică referinţe la entităţi special constitutite. Adnotările apar în perechi, pentru a indica începutul respectiv sfârşitul elementului. Adnotarea de sfârşit se deosebeşte de cea de început printr-o bară oblică.

Adnotari XML

```
<pagtitlu>

<traducator>Andrei Cornea</traducator> a tradus
  <titlu>Republica</titlu> de <autor>Platon</autor>
  <ingrijit>pentru volumul ingrijit de Constantin
  Noica</ingrijit>.

</pagtitlu>
```

Există o infinitate de posibilităţi de adnotare a unui text:

```
<pagtitlu>

 <traducator><substantivPropriu>Andrei</substantivPropriu>
  <substantivPropriu>Cornea</substantivPropriu></traducator>
  <verb>a tradus</verb>

 <titlu>Republica</titlu> de
  <autor><substantivPropriu>Platon</substantivPropriu></autor>
  <ingrijit>pentru <substantiv>volumul</substantiv>
  <verb>ingrijit</verb> de
  <substantivPropriu>Constantin</substantivPropriu>
  <substantivPropriu>Noica</substantivPropriu>.</ingrijit>

</pagtitlu>
```

Elemente vide, fără conţinut

```
<adnotare/>
```

Adnotările pot fi atribuite

```
<pagtitlu>

<traducator>Andrei Cornea</traducator> a tradus
  <titlu>Republica<titlu> de <autor>Platon</autor>

<ingrijit secret=”da”>

pentru volumul ingrijit de Constantin Noica.

</ingrijit>

</pagtitlu>
```

Structura logică a documentelor XML

- Fiecare document XML are o **structură de arbore**, cu un element rădăcină.

- Pentru fiecare element C care nu este radacina documentului, există un alt element P, **părintele lui C** (“parent”) în document, astfel încât C să fie conţinut direct de acesta. Se spune că C este **copilul** lui P (“child”).

Structura logică a documentelor XML

```
<a><b>  text1  <c>  text2  </c>  <d>  text3  </d>  </b>  text4
  <b>text5 </b></a>
```

```
<a>
     <b>
          text1
          <c> text2 </c>
          <d> text3 </d>
     </b>
     text4
     <b> text5 </b>
</a>
```

Componente ale structurii logice a documentelor XML

- elemente, conform precizărilor anterioare;
- comentarii;
- secţiuni CDATA;
- indicaţii (instructiuni) de procesare;
- referinţe la caractere sau la entităţi;
- declaraţii de prolog, care cuprind declaraţia XML şi, eventual definirea unui DTD.

Comentarii

- Pot apare practic oriunde în documentele XML, în afara adnotărilor

- Sunt delimitate de “<!--“ şi “-->”

```
<!-- Comentariu -->
```

Secţiuni CDATA

- Conţin date caracter (CDATA), care se iau ca atare (nu sunt analizate de procesoarele XML).
- Pot apare oriunde pot apare datele caracter. Sunt utilizate de obicei pentru a include blocuri de text ce conţin caractere ce ar fi altfel recunoscute drept adnotări.
- Încep cu şirul "<![CDATA[" si se termina cu şirul "]]>":

Secţiuni CDATA (exemplu)

```
<![CDATA[

            Text

]]>
```

Indicaţii de procesare (PI)

- Modalitate de a include în documente indicaţii (instrucţiuni) pentru aplicaţiile care prelucrează documentele XML. Ele nu sunt părţi ale datelor caracter, dar sunt trimise aplicaţiei.

- Indicaţiile de procesare sunt delimitate de “<?” şi ?>”, încep cu o ţintă (“PITarget”) utilizată pentru a identifica aplicaţia căreia îi este destinată instrucţiunea, urmată de informaţii suplimentare de trimis aplicaţiei.

Indicaţii de procesare (exemplu)

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
  href="b1.xsl" ?>
<adnotare> Text afisat </adnotare>
```

Referinţe la caractere sau la entităţi interne

- Secvenţe de caractere delimitate de “&#” sau “&#x” şi “;”, pentru caractere,
- Secvenţe de caractere delimitate de “&” şi “;”, pentru entităţi.
- Tin locul, unui caracter, respectiv unui şir de caractere, cu care sunt înlocuite de procesoarele XML

Documente XML:

- Bine formate (pot sa nu aiba un DTD asociat)
- Valide (au un DTD asociat si il respecta)

Reguli pentru documentele XML bine formate

Un **document XML bine format** trebuie să respecte următoarele reguli:

- Documentul trebuie să aibă **adnotările în perechi** (nu pot lipsi adnotări de început sau sfârşit, ca în cazul SGML sau HTML), şi contează dacă o literă este mare sau mică (<adn> este diferit de <ADN>, XML este “case sensitive”).
- Orice document XML trebuie să aibă un singur element superficial, numit **rădăcină** (“root”), sau element document, care nu apare în conţinutul altui element.

Reguli pentru documentele XML bine formate (cont.)

- Dacă sunt mai multe perechi de adnotări, ele trebuie să fie imbricate (să respecte o structură de “paranteze”)
- Numele elementelor trebuie să satisfacă anumite reguli, de exemplu, să înceapă cu o literă sau cu “_”
- Valorile atributelor adnotărilor trebuie să fie puse între ghilimele (de exemplu, <adn val=”2”> şi nu <and val=2>).

Reguli pentru documentele XML bine formate (cont.)

- Nu trebuie ca un atribut să apară de mai multe ori în aceeaşi adnotare.
- Pot exista adnotări care nu includ o zonă de text. Aceste adnotări, denumite **adnotări vide**, au caracterul “/” la sfârşit, de exemplu: <adnotare/>.
- Entităţile amp, lt, gt, apos, quot (pentru caracterelor “&”, “<”, “>”, apostrof şi ghilimea) pot fi utilizate fără a fi declarate.

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

<bib>
<webdoc src="http://concept.cs.uah.edu/CG/cg-standard.html" abr="CG">
Conceptual Graphs </webdoc>
<webdoc abr="CYC" src="http://www.cyc.org"> CYC</webdoc>
<webdoc abr="DOM" src="http://www.w3.org/DOM"> DOM </webdoc>
<webdoc abr="KIF" src="http://logic.stanford.edu/kif/kif.html">
Knowledge Interchange Format </webdoc>
<webdoc abr="KQML" src="http://www.cs.umbc.edu/kqml"> KQML </webdoc>
<webdoc abr="RDF" src="http://www.w3.org/RDF"> RDF Specification
<an>1999</an></webdoc>
<webdoc abr="RDFS" src="http://www.w3.org/TR/WD-rdf-schema"> RDF Schema
Specification</webdoc>
<articol abr="Sho93">
  <autor> Y. Shoham, Agent-oriented programming </autor>
  <revista nr="60">
    <titlu>Artificial Intelligence</titlu>
    <an>1993</an>
    <pp> 51-92</pp>
  </revista>
</articol>
<webdoc abr="XML" src="http://www.w3.org/XML"> XML </webdoc>
<carte abr="XMLc" isbn="1-861003-4-12">
  <autor> Cagle, K. </autor><autor>Gibbons, D.</autor><autor> Hunter,
D.</autor><autor> Ozu, N.</autor><autor>Pinnock, J.
 </autor><autor>Spencer, P.</autor>
 <titlu> Beginning XML </titlu>
 <editura>Wrox Press,</editura>
 <an> 2000</an>
</carte>
</bib>
```

```
- <bib>
    <webdoc src="http://concept.cs.uah.edu/CG/cg-standard.html"
abr="CG">Conceptual Graphs</webdoc>
    <webdoc abr="CYC" src="http://www.cyc.org">CYC</webdoc>
    <webdoc abr="DOM" src="http://www.w3.org/DOM">DOM</webdoc>
    <webdoc abr="KIF"
src="http://logic.stanford.edu/kif/kif.html">Knowledge Interchange
Format</webdoc>
    <webdoc abr="KQML" src="http://www.cs.umbc.edu/kqml">KQML</webdoc>
  - <webdoc abr="RDF" src="http://www.w3.org/RDF">
      RDF Specification
      <an>1999</an>
    </webdoc>
    <webdoc abr="RDFS" src="http://www.w3.org/TR/WD-rdf-schema">RDF
Schema Specification</webdoc>
  - <articol abr="Sho93">
      <autor>Y. Shoham, Agent-oriented programming</autor>
    - <revista nr="60">
        <titlu>Artificial Intelligence</titlu>
        <an>1993</an>
        <pp>51-92</pp>
      </revista>
    </articol>
    <webdoc abr="XML" src="http://www.w3.org/XML">XML</webdoc>
  - <carte abr="XMLc" isbn="1-861003-4-12">
      <autor>Cagle, K.</autor>
      <autor>Gibbons, D.</autor>
      <autor>Hunter, D.</autor>
      <autor>Ozu, N.</autor>
      <autor>Pinnock, J.</autor>
      <autor>Spencer, P.</autor>
      <titlu>Beginning XML</titlu>
      <editura>Wrox Press,</editura>
      <an>2000</an>
    </carte>
  </bib>
```

XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                      "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title>Virtual Library</title>
 </head>
 <body>
  <p>Moved to <a href="http://vlib.org/">vlib.org</a>.
  </p>
 </body>
0</html>
```

Documente XML valide

- Un document XML bine format devine un **document XML valid**, dacă, în plus faţă de restricţiile de bună formare, are asociată şi o definiţie de tip de document (**Document Type Definition** - DTD), pe care o respectă.

Tip de document

Repertoriu de adnotări

+

Modalitate de structurare

Avantaje separare DTD - document

- Acelaşi DTD poate fi folosit de mai multe documente. De exemplu, limbajul (DTD-ul) HTML.
- Acelaşi document poate fi considerat conform mai multor DTD-uri (compatibile).
- Se poate verifica faptul că un document satisface un anumit DTD (**validare**).

Declaraţia tipului de document

- în document (subset intern)
- într-o entitate externă

<!DOCTYPE element_document [sursa_subset_extern] [subset_intern_al_DTD]>

unde [sursa_subset_extern] poate fi :

- SYSTEM locaţie
- PUBLIC locaţie1 locaţie2

```
<?xml version="1.0"?>
<!DOCTYPE lectie
[
. . . continut declaratie DTD
]>

<lectie>
. . .
</lectie>
```

```
<?xml version="1.0"?>
<!DOCTYPE lectie SYSTEM "file:avl.dtd">

<lectie>
. . .
</lectie>
```

Declaraţii de adnotări

<!cuvânt_cheie parametru*>

- tip de element (“ELEMENT”),
- liste de atribute (“ATTLIST”),
- entităţi (“ENTITY”),
- notatie (“NOTATION”)

Declaraţia unui tip de element

- specificarea numelui elementului, nume care va fi folosit pentru adnotări;
- specificarea conţinutului elementului:

<!ELEMENT nume conţinut>

Dacă elementul are drept conţinut un text luat ca atare, fără o structurare în alte elemente, se foloseşte notaţia:

(#PCDATA)

Gramatica

- parantezele sunt folosite pentru grupare sau delimitare;
- “,” separă elemente care apar în secvenţă;
- “|” separă două elemente alternative, din care se poate alege unul drept conţinut;
- “+” indică apariţia o data sau de mai multe ori a unui element;
- “*” indică lipsa sau apariţia de mai multe ori a unui element;
- “?” specifică un element opţional;
- absenţa unui astfel de semn, înseamnă că elementul trebuie să apară exact o dată.

```
<!DOCTYPE carte [
 <!ELEMENT carte (pagtitlu, (capitol)+)>
 <!ELEMENT pagtitlu (titlu, autor, trad, ingrijit?)>
 <!ELEMENT titlu (#PCDATA)>
 <!ELEMENT autor (#PCDATA)>
 <!ELEMENT traducator (#PCDATA)>
 <!ELEMENT ingrijit (#PCDATA)>
 <!ELEMENT capitol (#PCDATA)>
]>
```

```
<!ELEMENT demonstratie (#PCDATA | concept | proprietate | fig)*>
```

Declararea listei de atribute a unui element

- Atributele sunt folosite pentru a asocia elementelor perechi de nume - valoare.

- Declaraţiile listelor de atribute precizează:
  - ce atribute pot apare într-un tip de element;
  - constrângerile legate de tipul acestor atribute;
  - valorile implicite pentru atribute.

Declararea listei de atribute a unui element

<!ATTLIST nume_element

(nume_atribut tip_atribut valoare_implicita)* >

Tipurile de atribute XML

- de tip şir, definit prin CDATA;
- de tip nume: ID, IDREF, IDREFS, NMTOKEN, NMTOKENS, ENTITY, ENTITIES
- de tip enumerat.

Nume

- Un **nume** (“**Name**”) este un cuvânt care începe cu o literă şi continuă cu litere, cifre, cratime, liniuţe de subliniere, două puncte sau cu puncte.
- Un **fragment de nume** (“**Nmtoken**”) este un mixaj dintre mai multe caractere pentru nume.

NameChar ::= Letter | Digit | '.' | '-' | '_' | ':' | CombiningChar | Extender

Name ::= (Letter | '_' | ':') (NameChar)*

Names ::= Name (S Name)*

Nmtoken ::= (NameChar)+

Nmtokens ::= Nmtoken (S Nmtoken)*

unde S este o zonă de spaţiu.

Restricţii lexicale şi semantice

- Tipul ID specifică faptul că atributul respectiv trebuie să aibă o valoare unică. ID nu poate avea ca declaraţie de valori implicite decât #REQUIRED sau #IMPLIED.
- IDREF este un tip care indică faptul că atributul este o referinţă la un element cu un ID.
- IDREFS este o mulţime de IDREF.
- NMTOKEN şi NMTOKENS se conformează producţiilor de mai sus.

Valori implicite

Valoare_implicita ::='#REQUIRED' | '#IMPLIED'
| (('#FIXED' S)? AttValue)

- #REQUIRED semnifică faptul că atributul şi valoarea sa vor trebui întotdeauna sa fie furnizate;
- #IMPLIED precizează că nu este furnizată nici o valoarea implicită;
- #FIXED stabileşte faptul că întotdeauna atributul va trebui să aibă valoarea implicită asociată.

```
<!DOCTYPE lectie [
<!ELEMENT lectie (titlu,(subiect)*) >
<!ATTLIST lectie
    nume ID #IMPLIED>
<!ELEMENT subiect ( concept | proprietate | fig | demonstratie | prog
)*>
<!ATTLIST subiect
     nume ID #REQUIRED>
<!ELEMENT fig EMPTY>
<!ATTLIST fig
     nr ID #REQUIRED
     caption CDATA #REQUIRED>
<!ELEMENT titlu (#PCDATA)>
<!ELEMENT concept (#PCDATA)>
<!ELEMENT proprietate (#PCDATA)>
<!ELEMENT demonstratie (#PCDATA | concept | proprietate | fig)*>
<!ELEMENT prog (#PCDATA)>
<!ATTLIST proprietate
     nr ID #REQUIRED
     nume CDATA #IMPLIED>
]>
```

```
01  <?xml version="1.0"?><!--prod.xml-->
02  <!DOCTYPE sales [
03  <!ELEMENT sales ( products, record )> <!--sales information-->
04  <!ELEMENT products ( product+ )>         <!--product record-->
05  <!ELEMENT product ( #PCDATA )>      <!--product information-->
06  <!ATTLIST product id ID #REQUIRED>
07  <!ELEMENT record ( cust+ )>                <!--sales record-->
08  <!ELEMENT cust ( prodsale+ )>     <!--customer sales record-->
09  <!ATTLIST cust num CDATA #REQUIRED>     <!--customer number-->
10  <!ELEMENT prodsale ( #PCDATA )>     <!--product sale record-->
11  <!ATTLIST prodsale idref IDREF #REQUIRED>
12  ]>
13  <sales>
14    <products><product id="p1">Packing Boxes</product>
15              <product id="p2">Packing Tape</product></products>
16    <record><cust num="C1001">
17              <prodsale idref="p1">100</prodsale>
18              <prodsale idref="p2">200</prodsale></cust>
19            <cust num="C1002">
20              <prodsale idref="p2">50</prodsale></cust>
21            <cust num="C1003">
22              <prodsale idref="p1">75</prodsale>
23              <prodsale idref="p2">15</prodsale></cust></record>
24  </sales>
```

Declaraţia unei entităţi interne

```
<!ENTITY Pub "Universitatea Politehnica Bucuresti">
```

Ori de câte ori procesorul XML va întâlni

```
&Pub;
```

în document, se va rescrie această secvenţă cu

“Universitatea Politehnica Bucuresti”

Declaraţia unei entităţi externe

```
<!ENTITY en SYSTEM
  “http://cs.pub.ro/~ceva/fisier.xml">

<!ENTITY en PUBLIC
  “http://cs.pub.ro/~ceva/fisier.xml"
   “... URI alternativ ...”>
```

Entităţi neanalizate

```
<!ENTITY nume SYSTEM “. . . locatie . . .”
       NDATA notatie>

<!ENTITY nume PUBLIC “. . . locatie1 . . .”
                      “. . . locatie2 . . .”
       NDATA notatie>
```

Entităţi parametrizate

Entităţile parametrizate sunt folosite exclusiv în interiorul unui DTD. Ele se declară prin:

```
<!ENTITY % nume “text inlocuitor”>
```

Referirea la aceste entităţi se face prin:

```
%nume;
```

Declararea notaţiilor

- Notaţiile sunt folosite pentru a specifica ce tipuri de entităţi neanalizate sunt folosite în document, care este URI-ul unde se poate găsi un program care poate prelucra entitatea.
- Declaraţia de notaţie asociază un nume fiecărei notaţii.

```
<!NOTATION png SYSTEM
  “http://www.wrox.com/Programs/PNG_Viewer.exe”>
```

Spaţii de nume

- Fiecare document XML are un vocabular de nume de adnotări

  exemplu : `<set>`

- => conflicte => adnotări prefixate
- adnotări prefixate :

`<svg:set>`

`<math:set>`

Spaţii de nume (cont.)

xmlns:[*prefix*]=*URI*

```
xmlns:svg="http://www.w3.org/2000/svg-20000629"
xmlns:math="http://www.w3.org/1998/Math/MathML”
```

xmlns:xsl=”http://www.w3.org/1999/XSL/Transform”

xmlns:xsl=”http://www.w3.org/1999/XSL/Format”

xmlns:xsl="http://www.w3.org/TR/WD-xsl"

xmlns:xt=”http://www.jclark.com/xt”

xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”

xmlns:rdfp='http://www.w3.org/XML/2000/04/rdf-parse/#”

xmlns:xlink="http://www.w3.org/1999/xlink"

Scheme XML

```
<?xml version="1.0" ?>
 <distributie>
   <personaj>MACBETH</personaj>
   <personaj>BANQUO</personaj>
   <grup>
     generali ai armatei regelui
   </grup>
 </distributie>
```

```
<!DOCTYPE distributie
 [<!ELEMENT distributie (personaj+, grup) >
  <!ELEMENT personaj (#PCDATA) >
  <!ELEMENT grup (#PCDATA) > ]>
```

```
<?xml version="1.0"?>
<Schema name="schema_sample_1"
    xmlns="urn:schemas-microsoft-com:xml-data"
    xmlns:dt="urn:schemas-microsoft-com:datatypes">

  <ElementType name="personaj" content="textOnly" model="closed"/>
  <ElementType name="grup" content="textOnly" model="closed"/>

  <ElementType name="distributie" content="eltOnly" model="closed">
      <element type="personaj" minOccurs="1" maxOccurs="*"/>
      <element type="grup" minOccurs="1" maxOccurs="1"/>
  </ElementType>
</Schema>
```

Adresarea către interiorul documentelor XML

- Limbajul XPath
- Limbajul XPointer

Limbajul XPath

XPath consideră un document XML ca un arbore format din şapte tipuri de noduri:

- **Rădăcina** - copii: nodul element corespunzător documentului şi, eventual noduri pentru instrucţiuni de procesare şi noduri comentariu.
- **Element**. Pentru fiecare element din document există câte un nod element. Copii - alte noduri element, noduri pentru instrucţiuni de procesare, comentarii şi noduri pentru text. Fiecare nod element poate avea un identificator unic, dacă este specificat prin tipul ID.

Limbajul XPath (cont.)

- **Atribut**. Fiecare nod element are o mulţime de noduri atribut, corespunzătoare atributelor sale. Aceste noduri atribut au ca părinte nodul element respectiv dar ele nu sunt copii ai elementului părinte.
- **Spaţiu de nume**.
- **Instrucţiuni de prelucrare**
- **Comentariu**
- **Text** - secvenţe maximale de date de tip caracter.

Limbajul XPath (cont.)

Principala construcţie XPath este **expresia**. În urma evaluării expresiilor XPath, se obţin obiecte de tipurile:

- un nod;
- mulţime de noduri;
- boolean;
- număr (în virgulă mobilă);
- şir de caractere.

Limbajul XPath (cont.)

Evaluarea fiecărei expresii XPath se face într-un context format din:

- un nod context;
- o pereche de numere întregi care reprezintă o poziţie şi o mărime a contextului, prima fiind mai mică decât cea de-a doua;
- legări de variabile;
- o bibliotecă de funcţii;
- mulţimea declaraţiilor de spaţii de nume.

Limbajul XPath (cont.)

Principalul tip de expresie XPath este **calea către o locaţie** (“location path”). Ea include convenţiile uzuale folosite pentru specificarea căilor către fişiere în sistemele de operare.

Căile către locaţii sunt o secvenţă de paşi separaţi de caracterul “/”. Ele pot fi:

- Relative - căi care încep de la nodul context curent.
- Absolute - căi care pleacă din rădăcina documentului, fapt indicat prin faptul că încep cu “/”.

Limbajul XPath (cont.)

axa::nod[predicat*]

Fiecare pas are trei părţi:

- o axă,
- un tip de nod
- zero sau mai multe predicate.

Limbajul XPath (cont.)

Axa specifică direcţia în care se face pasul curent. Ea poate fi:

- child, nodul copil, aceasta fiind axa implicită;
- descendant, adică copii sau copiii copiilor, pe oricâte nivele;
- parent
- ancestor, adică părinte sau părintele părintelui ş.a.m.d.;
- following-sibling
- preceding-sibling
- following
- preceding
- attribute, adică atributele nodului context;
- namespace
- self
- descendent-or-self
- ancestor-or-self

Limbajul XPath (cont.)

Expresiile XPath pot fi folosite diverse funcţii predefinite.

- number last() - ultimul element dintr-o mulţime
- number position() - poziţia unui element
- number count(node-set) - numărul de elemente
- node-set id(object) - nodul cu identificatorul dat ca argument
- string name(node-set?)
- string string(object?)
- boolean starts-with(string, string)
- boolean contains(string, string)
- string substring-before(string, string)
- number string-length(string?)

Limbajul XPath (cont.)

Abrevieri	Ce se consideră faţă de nodul context
*	toţi copii
text()	copiii de tip text
@nume	atributul denumit “nume”
@*	toate atributele
nume[i]	al i-lea copil “nume”
nume[last()]	ultimul copil “nume”
*/nume	nepoţii “nume”
//nume	descendenţii “nume”
.	nodul context
..	Părintele
../@nume	atributul “nume” al părintelui
nume[@atr=”val”]	copii “nume” care au atributul “atr” cu valoarea “val”

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

<bib>
<webdoc src="http://concept.cs.uah.edu/CG/cg-standard.html" abr="CG">
Conceptual Graphs </webdoc>
<webdoc abr="CYC" src="http://www.cyc.org"> CYC</webdoc>
<webdoc abr="DOM" src="http://www.w3.org/DOM"> DOM </webdoc>
<webdoc abr="KIF" src="http://logic.stanford.edu/kif/kif.html">
Knowledge Interchange Format </webdoc>
<webdoc abr="KQML" src="http://www.cs.umbc.edu/kqml"> KQML </webdoc>
<webdoc abr="RDF" src="http://www.w3.org/RDF"> RDF Specification
<an>1999</an></webdoc>
<webdoc abr="RDFS" src="http://www.w3.org/TR/WD-rdf-schema"> RDF Schema
Specification</webdoc>
<articol abr="Sho93">
  <autor> Y. Shoham, Agent-oriented programming </autor>
  <revista nr="60">
    <titlu>Artificial Intelligence</titlu>
    <an>1993</an>
    <pp> 51-92</pp>
  </revista>
</articol>
<webdoc abr="XML" src="http://www.w3.org/XML"> XML </webdoc>
<carte abr="XMLc" isbn="1-861003-4-12">
  <autor> Cagle, K. </autor><autor>Gibbons, D.</autor><autor> Hunter,
D.</autor><autor> Ozu, N.</autor><autor>Pinnock, J.
 </autor><autor>Spencer, P.</autor>
 <titlu> Beginning XML </titlu>
 <editura>Wrox Press,</editura>
 <an> 2000</an>
</carte>
</bib>
```

Expresie XPath	Rezultat obţinut
/bib/webdoc	Conceptual Graphs
//an	1999
/bib/articol/*	Y. Shoham, Agent-oriented programming
/bib/webdoc/following-sibling::webdoc	CYC
/bib/webdoc/attribute::*	http://concept.cs.uah.edu/CG/cg-standard.html
//an/../@abr	RDF
/bib/webdoc[6]	RDF Specification 1999
/bib/webdoc[4]/following-sibling::*	KQML
/bib/webdoc[@*]	Conceptual Graphs
//*[@nr]	Artificial Intelligence 1993 51-92
/bib/webdoc[6][@abr]	RDF Specification 1999
/bib/webdoc/text()	Conceptual Graphs
/bib/webdoc[last()]	XML
//*[count(*)>2]/*[2]	CYC

Limbajul XPointer

În HTML se poate face o referire (de exemplu, o legătură) doar la un întreg document sau la un punct specificat prin #ancora (şi definit în document prin <a name=”ancora”>).

Expresiile XPointer sunt extensii ale XPath şi pot fi adăugate la un URI, după caracterul “#”, ca în următorul exemplu:

http:// . . . /b1.xml#xpointer(/bib/articol)

Limbajul XPointer (cont.)

- XPointer este un limbaj care extinde XPath în sensul că permite adresarea nu numai a unui nod sau a unei mulţimi de noduri (ca în XPath) ci şi a unui punct sau a unui fragment dintr-un nod al unui document.
- Un **punct** dintr-un document este precizat printr-o pereche formată dintr-un nod container şi un index. Dacă nodul container are fii, indexul se referă la al câtelea fiu este considerat. Dacă nodul container nu are fii, de exemplu, este un nod text, indexul este considerat faţă de şirul de caractere care formează nodul.

Conexiuni între documente. Limbajul XLink

Specificarea legăturilor în XLink se face folosind documente XML bine formate, pe baza unui repertoriu de atribute, după cum urmează:

- "type" este un atribut care specifică **tipul** legăturii. El poate avea ca valoare:
  - "simple", pentru legăturile simple, între două resurse.
  - "extended", pentru legături extinse.

Legaturi extinse

Pentru acest tip de legături se pot defini elemente pentru specificări suplimentare, conform următoarelor patru atribute:

- “locator”, pentru resurse externe,
- “arc”, pentru specificarea direcţiilor de traversare a legăturilor,
- “resource”, pentru specificarea de resurse locale,
- “title”, pentru a specifica un element pentru titlu (în locul atributului cu acelaşi nume, de care se va vorbi un pic mai jos).

Atribute referitoare la semantica legăturii

- “role”,
- “title”.

Atribute referitoare la comportarea legăturii

- “actuate”, precizează când va căutată noua resursă. Poate avea ca valoare:
  - “onLoad”, similar cu “<img>” din HTML,
  - “onRequest”, similar cu “<a href=. . .>”,
  - “undefined” - specific aplicaţiei.
- “show”, precizează cum va fi afişată noua resursă în urma parcurgerii unei legături. Poate avea ca valoare modalitatea de afişare:
  - “new”, într-o fereastră separată;
  - “replace”, prin înlocuire, ca la “<a href=. . .>” din HTML;
  - “embed”, ca la “<img>” din HTML;
  - “undefined”

Atribute care descriu direcţionalitatea

- “from”,
- “to”.

Foi de stil

O foaie de stil XSLT tipică este un document de tip "xsl:stylesheet" :

```
<xsl:stylesheet

    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    . . . . . .

</xsl:stylesheet>
```

Reguli XSLT

Regulile XSLT au două părţi:

- O parte care conţine un **şablon** pentru specificarea locului în care se aplică regula. Pentru referirea la anumite noduri din arborele unui document, se foloseşte limbajul XPath pentru indicarea căilor de acces la acele noduri. Nodul astfel referit este **nodul context** pentru elementele din partea a doua a regulii, care descriu fragmentul de arbore rezultat.
- Efectul aplicării regulii, adică fragmentul de arbore rezultat.

Reguli XSLT

Declararea unei reguli se face cu elementul “template”:

<xsl:template match=". . . sablon . . .">

. . . . . fragment de arbore rezultat

</xsl:template>

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="b1.xsl" ?>
 <bib>
<webdoc src="http://concept.cs.uah.edu/CG/cg-standard.html" abr="CG"> Conceptual Graphs </webdoc>
<webdoc abr="CYC" src="http://www.cyc.org"> CYC</webdoc>
<webdoc abr="DOM" src="http://www.w3.org/DOM"> DOM </webdoc>
<webdoc abr="KIF" src="http://logic.stanford.edu/kif/kif.html"> Knowledge Interchange Format </webdoc>
<webdoc abr="KQML" src="http://www.cs.umbc.edu/kqml"> KQML </webdoc>
<webdoc abr="RDF" src="http://www.w3.org/RDF"> RDF Specification  <an>1999</an></webdoc>
<webdoc abr="RDFS" src="http://www.w3.org/TR/WD-rdf-schema"> RDF Schema Specification</webdoc>
<articol abr="Sho93">
  <autor> Y. Shoham, Agent-oriented programming </autor>
  <revista nr="60">
    <titlu>Artificial Intelligence</titlu>
    <an>1993</an>
    <pp> 51-92</pp>
  </revista>
</articol>
<webdoc abr="XML" src="http://www.w3.org/XML"> XML </webdoc>
<carte abr="XMLc" isbn="1-861003-4-12">
  <autor> Cagle, K. </autor><autor>Gibbons, D.</autor><autor> Hunter, D.</autor><autor> Ozu,
      N.</autor><autor>Pinnock, J.
 </autor><autor>Spencer, P.</autor>
 <titlu> Beginning XML </titlu>
 <editura>Wrox Press,</editura>
 <an> 2000</an>
</carte>
</bib>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <html>
     <body>
        <xsl:value-of select="/bib/webdoc[@abr='RDF']"/>
     </body>
    </html>
  </xsl:template>

</xsl:stylesheet>
```

Rezultat:

RDF Specification 1999

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <html>
     <body>
      <h2>Paginile de web sunt :</h2>
       <ol>
        <xsl:for-each select="/bib/webdoc">
           <li>
            <xsl:value-of select="."/>
           </li>
        </xsl:for-each>
       </ol>
     </body>
    </html>
  </xsl:template>

</xsl:stylesheet>
```

```
<html>
<body>
<h2>Paginile de web sunt :</h2>
<ol>
<li> Conceptual Graphs </li>
<li> CYC</li>
<li> DOM </li>
<li> Knowledge Interchange
Format </li>
<li> KQML </li>
<li> RDF Specification  1999</li>
<li> RDF Schema
Specification</li>
<li> XML </li>
</ol>
</body>
</html>
```

Paginile de web sunt :

1. Conceptual Graphs
2. CYC
3. DOM
4. Knowledge Interchange Format
5. KQML
6. RDF Specification 1999
7. RDF Schema Specification
8. XML

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <html>
     <body>
       <h2>Situri de vizitat!</h2>
       <ul>
       <xsl:apply-templates/>
       </ul>
     </body>
    </html>
  </xsl:template>

  <xsl:template match="webdoc">
    <li>
      <xsl:value-of select="@src"/>
    </li>
  </xsl:template>

  <xsl:template match="articol">
  </xsl:template>

  <xsl:template match="carte">
  </xsl:template>

</xsl:stylesheet>
```

```
<html>
<body>
<h2>Situri de vizitat!</h2>
<ul>
<li>http://concept.cs.uah.edu/CG/cg-standard.html</li>
<li>http://www.cyc.org</li>
<li>http://www.w3.org/DOM</li>
<li>http://logic.stanford.edu/kif/kif.html</li>
<li>http://www.cs.umbc.edu/kqml</li>
<li>http://www.w3.org/RDF</li>
<li>http://www.w3.org/TR/WD-rdf-schema</li>

<li>http://www.w3.org/XML</li>

</ul>
</body>
</html>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/bib/webdoc">
    <xsl:value-of select="@src"/>
  </xsl:template>
  <xsl:template match="/bib/articol">articol</xsl:template>
  <xsl:template match="/bib/carte">carte</xsl:template>
  <xsl:template match="/bib">
  document
  </xsl:template>
</xsl:stylesheet>
```

Rezultatul este:

```
  document
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

 <xsl:template match="/">
  <html>
   <body>
    <h2>Situri de vizitat!</h2>
    <ul>
    <xsl:apply-templates/>
    </ul>
   </body>
  </html>
 </xsl:template>

 <xsl:template match="webdoc">
  <li>
   <xsl:element name="a">
    <xsl:attribute name="href">
     <xsl:value-of select="@src"/>
    </xsl:attribute>
    <xsl:value-of select="."/>
   </xsl:element>
  </li>
 </xsl:template>

 <xsl:template match="articol">
 </xsl:template>

 <xsl:template match="carte">
 </xsl:template>

</xsl:stylesheet>
```


D:\trausan\ca\bbb.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address D:\trausan\ca\bbb.html
Go Links
Siturи de vizitat!
Conceptual Graphs
CYC
DOM
Knowledge Interchange Format
KQML
RDF Specification 1999
RDF Schema Specification
XML
My Computer