

Netscape 6: reîntoarcerea...

O varianta extinsa e disponibila in cartea 'Tehnologii Web', Matrix Rom, 2001: www.infoiasi.ro/~busaco/books/web.html

[Sabin-Corneliu Buraga](#)

Articol publicat in *PC Report*, vol.9, nr.99 - decembrie 2000

În luna martie 1998, compania Netscape, încă independentă, ia una dintre cele mai radicale decizii publicând sursele de la popularul browser care a dus la dezvoltarea explozivă a Web-ului. Astfel, apare *Mozilla*, varianta *open-source* a navigatorului Netscape.

De atunci, spațiul WWW s-a maturizat, fiind guvernat de standarde precum XML, RDF, DOM, pentru a da numai câteva exemple. Fanii Netscape așteptau de mult noua versiune a browserului, rămas cam în urmă față de concurentul său Internet Explorer.

Dar revenirea, precedată de o serie de versiuni de test Mozilla, se pare că va fi (re)aduce Netscape în prim-planul Web-ului. Deși nu mai oferă, în varianta standard, suport pentru LDAP (Lightweight Directory Access Protocol), Netscape 6 implementează fidel toate standardele de succes ale Consorțiului Web: HTML 4.01, CSS - nivelele 1 și 2, XML 1.0, DOM 1.0 și RDF. Mai mult, întreg browserul a fost rescris complet, fiind conceput modularizat, iar interfețele componentelor Netscape sunt scrise într-un limbaj bazat pe XML, denumit *XUL* (*Extensible User-interface Language*). Oricine poate să compună propriile sale teme pentru interfețele Netscape.

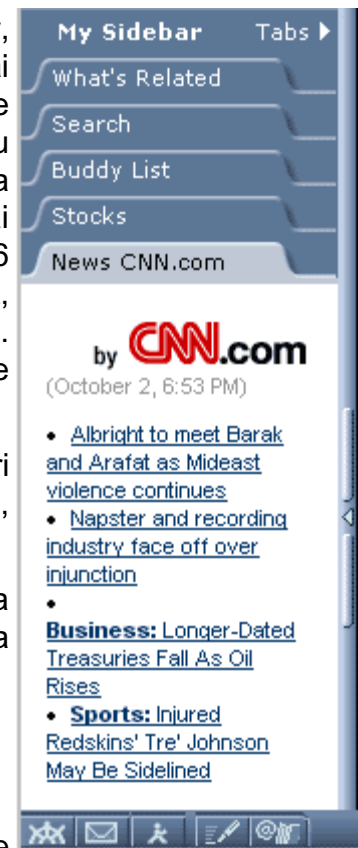
Probabil că în prezent Netscape 6 este cel mai flexibil browser, oferind mai mult decât Internet Explorer în ceea ce privește DOM. Mai mult, Netscape 6 implementează standardul DOM la fel pentru toate platformele pe care rulează (UNIX, Windows, Macintosh), ceea ce nu se poate afirma despre Internet Explorer. Dacă pentru formatarea documentelor XML, Internet Explorer 5 poate utiliza o varianta mai veche de XSL (limbaj care încă nu a fost standardizat), în Netscape 6 acest lucru se realizează via CSS. Nici JavaScript nu a fost uitat, Netscape 6 oferind dezvoltatorilor noua versiune JavaScript 1.5. Suportul pentru limbajul Java se bazează pe Open JVM Interface (OJI).

Din păcate, Netscape 6 nu oferă nici un suport pentru prezentări hipermedia sincronizate scrise în SMIL, față de Internet Explorer 5.5, iar implementarea proprietăților CSS2 este încă precară.

În continuare vom vedea câteva detalii referitoare la implementarea RDF în Netscape 6 și despre noul limbaj XUL de specificare a interfețelor-utilizator.

Suportul pentru RDF

Netscape 6 și varianta sa open-source Mozilla sunt unicele browsere care, în prezent, implementează RDF (Resource Description Framework), scopul fiind menținerea unei baze de date de mici dimensiuni conținând anumite informații interne și, mai important, utilizarea modelului RDF împreună cu sabloanele XUL ca interfață abstractă de programare pentru afișarea informațiilor. Dar despre XUL vom discuta în detaliu puțin mai încolo.



Sursele de date în Netscape 6 sunt manipulate via RDF prin intermediul managerului de componente (*Component Manager*), utilizându-se un URI special de forma `component://netscape/rdf/datasource?name=valoare`. Aceste surse de date pot fi lista de URI-uri stocate de *bookmark*-ul browserului, istoricul navigării, lista serverelor interogate pentru obținerea poștei electronice (prin intermediul protocolului IMAP) etc.

Tot prin intermediul RDF, sunt organizate și informațiile din Netscape *side-bar*, o componentă utilă pentru gruparea legăturilor favorite, a persoanelor de contact și altele (vezi figura alăturată).

Astfel, navigatorul furnizează diverse servicii interne RDF instanțiate și interogate prin intermediul metodelor JavaScript:

```
var RDF =
  Components
  .classes["component://netscape/rdf/rdf-service"]
  .getService(Components.interfaces.nsIRDFService);

// preia în mod asincron un fișier RDF
var ds = RDF.GetDataSource("http://www.mozilla.org/rdf_file.rdf");

// se verifică dacă a fost încărcat sau nu
var remote =
  ds.QueryInterface(Components.interfaces.nsIRDFRemoteDataSource);

if (remote.loaded) {
  alert("Sursa de date a fost încărcată!");
}
else {
  alert("Sursa de date încă n-a fost încărcată complet...");
}
```

Pentru accesarea datelor din *bookmark*-ul navigatorului putem utiliza următorul cod:

```
// utilizează managerul de componente
var bookmarks =
  Components
  .classes["component://netscape/rdf/datasource?name=bookmarks"]
  .getService(Components.interfaces.nsIRDFDataSource);

// prelucrează intrările din bookmark
if (bookmarks.HasAssertion(
  RDF.GetResource("http://home.netscape.com/NC-rdf#BookmarksRoot"),
  RDF.GetResource("http://home.netscape.com/NC-rdf#child"),
  RDF.GetResource("http://home.netscape.com/NC-rdf#PersonalToolbarFolder"),
  true) {
  /* prelucrează datele */
}
```

Fișierul de *bookmark*-uri este un document XML (în particular XUL) de forma următoare:

```
<tree datasources="rdf:bookmarks">
  ...
</tree>
```

Manipularea colecțiilor de date RDF (e.g. `<rdf:Seq>`) se realizează prin intermediul URI-urilor speciale `component://netscape/rdf/container-utils`, funcțiile de interogare a existenței unei colecții fiind `IsSeq()`, `IsBag()` și `IsAlt()`. Pentru crearea unei colecții poate fi folosită una dintre funcțiile JavaScript `MakeSeq()`, `MakeBag()` ori `MakeAlt()`.

Pentru verificarea validității unui document RDF/XML pot fi utilizate programele `rdfpoll`, incluse în distribuția Mozilla. Componenta de interfață beneficiind de facilitățile de reprezentare a informațiilor în RDF poartă numele *Aurora* (cunoscută și sub denumirea de *NavCenter*), folosită la afișarea, explorarea și integrarea datelor în cadrul navigatorului Mozilla (e.g. structuri de directoare de pe discul local, tabele de conținut ale site-urilor Web, indici de căutare, servicii taxonomice precum Yahoo sau Open Directory, istoricul documentelor parcurse etc.).

În acest ultim exemplu, putem remarca utilizarea unei secvențe RDF conținând descrierile componentelor de bază ale browserului:

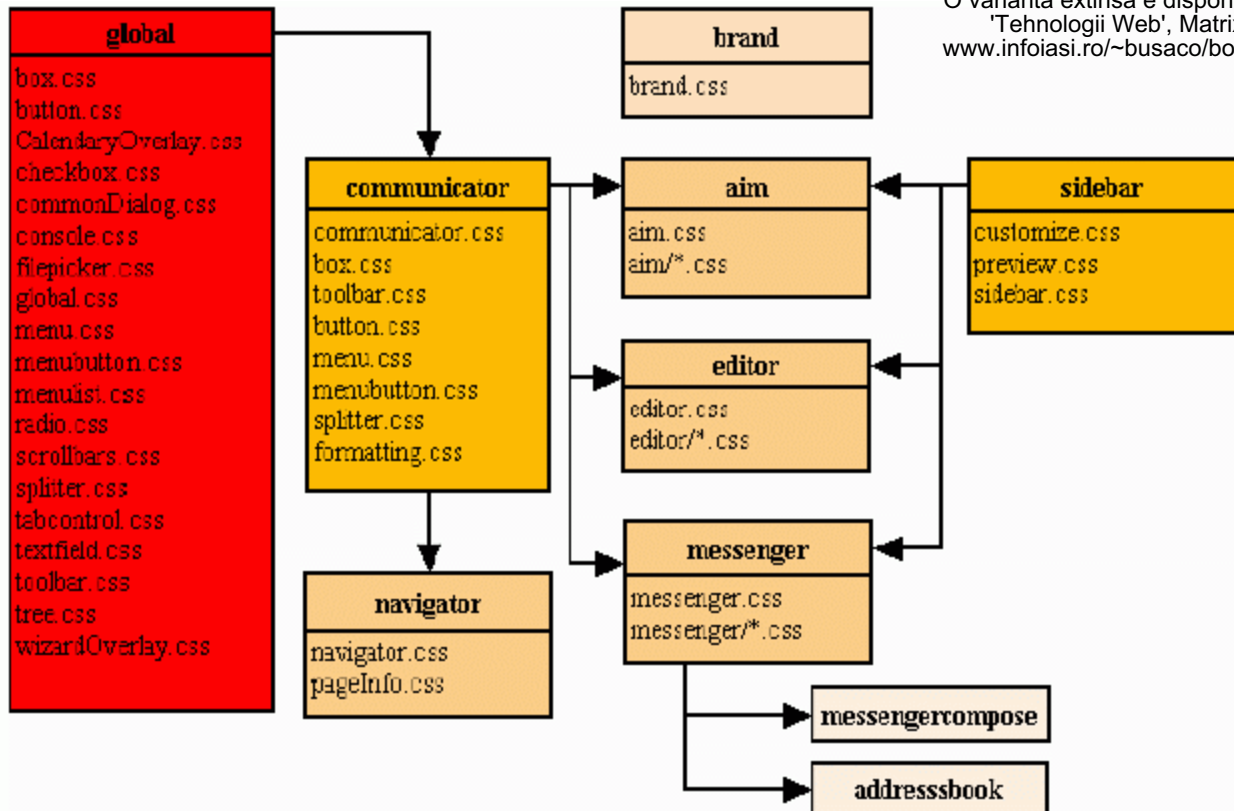
```
<?xml version="1.0"?>
<RDF:RDF xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:chrome="http://www.mozilla.org/rdf/chrome#">

  <!-- list all the packages being supplied by this jar -->
  <RDF:Seq about="urn:mozilla:package:root">
    <RDF:li resource="urn:mozilla:package:editor"/>
    <RDF:li resource="urn:mozilla:package:communicator"/>
    <RDF:li resource="urn:mozilla:package:navigator"/>
  </RDF:Seq>

  <!-- package information -->
  <RDF:Description about="urn:mozilla:package:editor"
    chrome:displayName="Editor"
    chrome:author="mozilla.org"
    chrome:name="editor">
  </RDF:Description>
  <RDF:Description about="urn:mozilla:package:communicator"
    chrome:displayName="Mozilla Core"
    chrome:author="mozilla.org"
    chrome:name="communicator">
  </RDF:Description>
  <RDF:Description about="urn:mozilla:package:navigator"
    chrome:displayName="Navigator"
    chrome:author="mozilla.org"
    chrome:name="navigator">
  </RDF:Description>
</RDF:RDF>
```

XUL - limbaj de descriere a interfețelor cu utilizatorul

Vom prezenta în continuare un limbaj bazat pe XML utilizat pentru specificarea elementelor de interfață cu utilizatorul în cadrul navigatoarelor Netscape din generația a patra (*Netscape Navigator*), a cincea (*Mozilla*) și a șasea (*Gecko*). În fapt, **XUL (Extensible User Interface Language)** reprezintă o aplicație independentă de platformă bazate pe standarde precum HTML 4, XML 1.0, spațiile de nume XML, CSS - nivele 1 și 2, DOM - nivele 1 și 2, destinată reprezentării interfețelor cu utilizatorul în cadrul navigatoarelor produse de Netscape Communications.



Structura de elemente de interfața în Netscape 6

XUL aduce în prim-plan metaforele utilizate de proiectanții interfețelor utilizator clasice (boxe de dialog, butoane, ferestre), fiind un mijloc facil de specificare și de implementare a acestora pe diferite platforme, reducând costul de proiectare și programare a interfețelor grafice pentru Macintosh, Motif, Windows, GTK etc. Un alt avantaj îl constituie faptul că indiferent de platformă, interfețele construite cu XUL vor arăta și vor reacționa similar la comenzile utilizatorului.

Structura unui document XUL

Elementul rădăcină al unui fișier XUL este `<window>`, fiecare document trebuind să conțină măcar un element de tip `window`. O interfață XUL este definită de elementul rădăcină. Acest element poate conține diferite *widget*-uri (controale) facilitând interacțiunea cu utilizatorul:

```

<window title="O interfață XUL" id="main_win">
  <menubar id="main_menu">
    <menu value="File" id="file_menu"/>
    <popup>
      <menuitem value="New" onclick="CreateNewDoc()" />
      <menuitem value="Open" onclick="OpenDoc()" />
      <menuitem value="Close" onclick="CloseDoc()" />
    </popup>
    <menu value="Edit" />
    <menu value="Help" />
  </menubar>
  <box style="background-color:lightgreen;" id="main_box">
    <browser />
  </box>
</window>
    
```

După cum se remarcă, o interfață poate fi compusă din diferite tipuri de *widget*-uri grupate în

următoarele categorii:

O varianta extinsa e disponibila in cartea
'Tehnologii Web', Matrix Rom, 2001:
www.infoiasi.ro/~busaco/books/web.html

- ferestre (definesc locul și modul de afișare a informațiilor, plus evenimentele la care trebuie să reacționeze)

`<window>`

- cutii de vizualizare

`<box>`, `<titledbox>`, `<text>`, `<textfield>`

- meniuri și bare de meniu

`<menubar>`, `<menu>`, `<menubutton>`, `<menulist>`, `<menupopup>`,
`<menuitem>`

- cutii și bare de unelte

`<toolbox>`, `<toolbar>`

- tabulatoare

`<tabcontrol>`, `<tabbox>`, `<tabpanel>`, `<tab>`

- butoane

`<radiogroup>`, `<radio>`, `<button>`, `<checkbox>`

- bare de defilare

`<scrollbar>`, `<scrollbarbutton>`, `<slider>`

- indicatoare de evoluție a unei acțiuni

`<progressmeter>`

- organizatoare (pentru afișarea altor informații, într-o formă structurată)

`<grid>`, `<tree>`, `<treehead>`, `<treecol>`, `<treerow>`, `<treeitem>`,
`<deck>`, `<stack>`

- complexe (utilizate pentru includerea de componente Mozilla, precum editorul HTML, navigatorul Web etc.)

`<editor>`, `<browser>`, `<iframe>`

În ansamblu, *widget*-urile formează așa-numitul *XPToolkit* (*XP* provenind de la *cross-platform*).

Un document XUL trebuie să conțină următoarele instrucțiuni de procesare care asociază interfeței foile de stiluri utilizate pentru afișarea *widget*-urilor pe ecran și spațiile de nume ale elementelor XUL (opțional HTML):

```
<?xml-stylesheet href="chrome://global/skin/xul.css" type="text/css"?>
<window
  xmlns:html="http://www.w3.org/TR/REC-html40"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
...
</window>
```

Exemple

- a. Generarea unei cutii de vizualizare avind titlu, grupând în interiorul ei două butoane:

```
<!-- încarcă un stil de afișare a interfeței (aici 'chrome') -->
<?xml-stylesheet href="chrome://global/skin/" type="text/css" />
<!-- defineste fereastra de afișare -->
<window id="new-menu"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <titledbox style="background-color: lightblue;" orient="vertical">
    <title>Nice Buttons</title>
    <button value="Click Me" />
    <button value="Push Me" />
  </titledbox>
</window>
```

Acest document XUL va avea următoarea reprezentare grafică:



O cutie de vizualizare cu doua butoane

- b. Pentru fiecare buton, se pot preciza proprietăți și acțiuni pentru diferite pseudo-clase de stiluri CSS (*hover* și *active*) și evenimente (e.g. *onCommand*), utile pentru exploatarea interfeței. Astfel, putem avea:

```
<style>
button.bookmarkbutton {
  margin: 0px;
  max-width: 8em;
  font: inherit;
  font-size: smaller;
  color: #99CCCC;
  border: none;
  padding: 1px;
  background-image: none;
}

button.bookmarkbutton:hover {
  color: #CCFFFF;
}

button.bookmarkbutton:active {
  color: #CCFFFF;
  padding-left: 2px;
  padding-top: 2px;
  padding-right: 0px;
  padding-bottom: 0px;
}
</style>
<button class="bookmarkbutton"
  value="&TOC.label;"
  onCommand="GetDoc('bookmarks.html')" />
```

- c. Folosirea unui control de alegere a culorii pentru modificarea conținutului unui document HTML:

```
<html:script language="JavaScript">
  function ChangeColor(cp)
  {
    var color = cp.getAttribute('color');
    document.getElementById("input").value = color;
    document.getElementById("coldiv").style.backgroundColor = color;
  }
</html:script>
<colorpicker id="cp" onclick="ChangeColor(this);"/>
<html:p />
<html:input id="input" />
<html:p />
<html:div id="coldiv"
  style="width:100px; height:100px; background-color:white" />
<html:hr />
```

Remărcăm utilizarea tag-urilor HTML, aparținând spațiului de nume definit de specificația HTML 4.0, în cadrul unei interfețe XUL. În acest exemplu, putem observa de asemeni folosirea unei funcții JavaScript pentru modificarea în mod dinamic a proprietăților unui element `<div>`.



Utilizarea elementului XUL `<colorpicker>` în cadrul unui document HTML

Prin intermediul limbajului XUL pot fi definite așa-numitele *teme* (foarte asemănătoare *skin*-urilor), Netscape 6 venind cu două teme predefinite: *Modern* (pe tonuri de albastru) și *Clasic* (emulând vechea interfață Netscape).

Compania *Alphanumeric* actualmente pregătește o aplicație destinată editării de teme pentru Netscape 6.

XUL Chrome

O serie dintre facilitățile puternice ale browser-ului provin din utilizarea unui *chrome*, acest termen avind diferite semantici în funcție de contextul în care apare. În general, termenul se referă la o interfața XUL și toate fișierele ei adiționale. De asemeni, *chrome* reprezintă structura și conținutul unui document XUL, inclusiv definițiile de proprietăți de stiluri CSS și localizarea interfeței pe o anumită platformă particulară. Un *chrome* este specificat în cadrul interfeței XUL printr-o instrucțiune de procesare `<?xml-stylesheet href="chrome://global/skin/"`

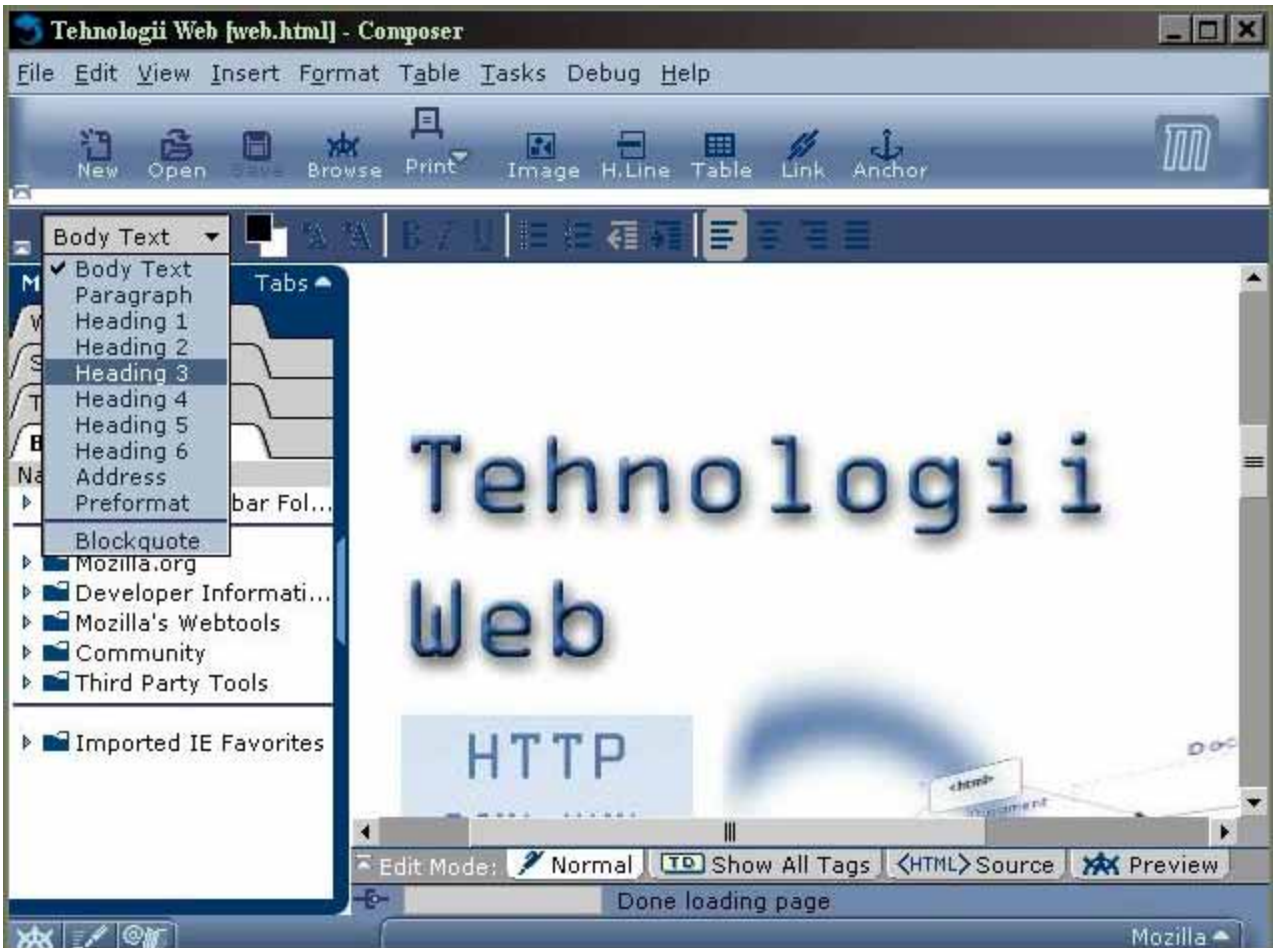
`type="text/css"?>`, lucru care semnaleză motorului de procesare HTML că toate fişierele asociate acestui *chrome*.

În exemplul următor, *chrome* indică un fişier utilizat pentru reprezentarea grafică a unui control grafic, încărcat din cadrul ierarhiei de fişiere care compun un pachet dintr-o aplicaţie Netscape:

```
<menuitem value="Help"
  oncommand="window.openDialog('chrome://help/content/help.xul',
    '_blank', 'chrome,all,dialog=no')" />
```

Dar ce înseamnă un pachet? Vom răspunde la această întrebare în cele ce urmează.

Un *pachet* reprezintă un *chrome* intern, în contextul arhitecturii navigatorului (altfel spus, un pachet este un *chrome* încorporat în cadrul aplicaţiilor Mozilla). Astfel, un pachet poate fi văzut ca un fragment din codul unei interfeţe, fiind înglobat într-un anumit loc în ierarhia de aplicaţii Mozilla sau Netscape. Ca şi un *chrome*, un pachet poate conţine fişiere de stil CSS, definiţii de interfeţe XUL, fişiere grafice şi şiruri de localizare, plus cod specific platformei particulare pe care rulează aplicaţia. Pachetul *navigator* este un pachet definit în `mozilla/bin/chrome/navigator`, iar componenta *mail/news* aparţine ierarhiei `mozilla/bin/chrome/mailnews/`.



Interfaţa editorului HTML încorporat în *Mozilla* (versiunea Milestone 18) folosind *chrome*-ul Modern

Ca şi la sistemele de fişiere, ierarhia este una arborescentă:

```
navigator/  
content/
```

```
default/  
  navigator.xul  
  ...  
skin/  
  default/  
    navigator.css  
    nav-icon.gif  
    ...  
locale/  
  US-en/  
    navigator.dtd
```

Această modularizare asigură flexibilitatea navigatorului Web.

Putem referi un fișier particular dintr-un pachet, prin intermediul unui URI de genul: `chrome://help/content/help.xul`. La crearea unui *chrome* diferit pentru un pachet, structura acestuia se schimbă, inserind încă o intrare în ierarhia de componente ale pachetului.

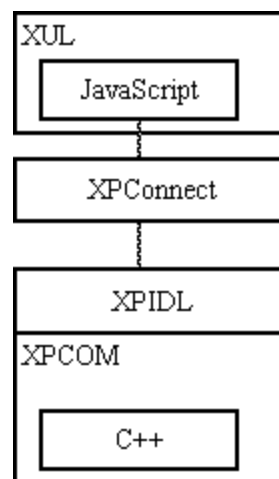
Pentru accesarea diferitelor elemente de interfață, se pot folosi modelele *DOM* (*Document Object Model*) sau extensia acestuia *AOM* (*Application Object Model*), model destinat special pentru prelucrarea documentelor XUL. Astfel, prin intermediul AOM putem referi diferite *widget-uri* ca ierarhii de colecții de controale ale interfețelor.

Integrarea XUL în navigatorul Mozilla

Desigur, un navigator Web reprezintă mai mult decit o interfață. Orice browser trebuie să fie independent de platformă, bazat pe standarde WWW, ușor de configurat și de exploatat și rapid. Mai mult, fiecare navigator Web actual oferă mai mult decit posibilitatea explorării structurii hipermedia a spațiului WWW, integrând clienți pentru poștă electronică și grupuri de știri, transferul de fișiere și altele (managementul *cookie*-urilor). Orice navigator evoluat dă posibilitatea programatorilor să includă în cadrul paginilor scripturi concepute în limbaje standardizate (e.g. JavaScript).

Arhitecții și implementatorii browserului Mozilla utilizează trei tehnologii independente de platformă pentru a conecta nucleul navigatorului (denumit *Gecko*) cu interfața modelată în XUL.

Structura modulelor care întră în componenta navigatorului este ilustrată mai jos:



Interacțiunea componentelor interne ale navigatorului Netscape 6

XPCOM reprezintă o metodologie de programare a aplicațiilor distribuite, inspirată de COM (Component Object Model), independentă de platforma hardware și software: fragmente de cod oferă interfețe independente de limbaj și platforma spre alte obiecte care pot să acceseze

diferite servicii. Astfel, proiectarea și compilarea aplicației se pot realiza în diferite moduri, implementarea obiectelor, programatorii trebuind să cunoască numai interfețele orientate de obiecte. Aceste interfețe vor fi descrise într-un limbaj special denumit XPIDL, extensie a limbajului de definire a interfețelor IDL (Interface Definition Language) folosit de aplicațiile CORBA și COM. XPConnect este tehnologia prin intermediul căreia interfețele descrise de XPIDL sunt conectate la părțile de cod care implementează obiectele în JavaScript, limbajul utilizat de XUL.

Nucleul navigatorului, purtând numele de cod *Gecko*, reprezintă o componentă modulară, bazată pe clienții Netscape/Mozilla, care poate fi inclusă atât în aplicații Web de mari dimensiuni cu propriile lor interfețe-utilizator cât și în dispozitive miniaturizate. Filosofia dezvoltării actuale și viitoare a nucleului ține cont de următoarele aspecte:

- *Gecko* nu reprezintă un browser Web, neavând propria sa interfață, și nu are integrate servicii auxiliare (de exemplu, clientul de poșta electronică sau editorul HTML) și nu oferă facilități suplimentare (e.g. *bookmarks*);
- se bazează numai pe standarde deschise redactate de Consorțiul Web: HTML 4.0, CSS1, CSS2, DOM1, DOM2, XML, RDF etc., iar HTML dinamic este simulat prin DOM;
- este independent de platforma (prin tehnologia XPCOM), foarte compact (prima versiune putea fi stocată pe o singură dischetă) și rapid;
- partea de comunicare în rețea este ea însăși modulară, purtând numele de *Necko*;
- este extensibil, sursele fiind publice;
- oferă suport pentru navigatoarele mai vechi.

În acest mod, nucleul poate fi folosit de mai multe navigatoare sau aplicații, în diferite circumstanțe și rulând pe diverse platforme, interfața cu acesta putând fi aleasă de utilizator, beneficiind de flexibilitatea oferită de XUL.

În loc de final

Dezvoltatorii de situri Web complexe, bazate pe XML, pot beneficia de implementarea precisă, bazată pe standardele Consorțiului WWW, a limbajului. Mai mult, Netscape 6 este singurul browser suportând RDF și oferind facilități avansate pentru căutarea de informații pe Web, configurarea interfeței în XUL sau folosirea componentelor sale în cadrul unor aplicații proprii utilizatorului.

Sabin-Corneliu Buraga este doctorand în *Computer Science* la Universitatea "A.I.Cuza" din Iași, putând fi contactat la adresa busaco@infoiasi.ro.