



UNIUNEA EUROPEANĂ



GVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Interacțiunea om-calculator

30. Proiectare și implementare de interfețe la documente XML (XUL)

XUL –XML USER INTERFACE LANGUAGE

prezentare

- 1. Prezentare generală**
 - 1. Introducere**
 - 2. Contextul apariției**
 - 3. Scopul apariției**
 - 4. Funcționalități generice**
- 2. Sintaxa**
 - 1. Elemente de bază ale sintaxei**
 - 2. Organizarea unei interfețe XUL**
- 3. Cazuri de utilizare**
 - 1. XUL ca o platforma de aplicații**
 - 2. De ce XUL?**
 - 3. Standarde și alte tehnologii suportate**
 - 4. Exemple**
- 4. Instrumente**
- 5. Concluzii**
- 6. Bibliografie**

1. Prezentare generală

1.1. Introducere

XUL (se pronunță "zool") este limbajul bazat pe XML folosit de Mozilla pentru definirea interfeței cu utilizatorul care permite construirea de aplicații independente de platformă care pot rula conectate sau nu la Internet. Aplicațiile pot fi personalizate foarte ușor cu text alternativ, grafică, layout în funcție de scopul lor. XUL este o unealtă ușor de folosit pentru programatorii Web deja familiarizați cu DHTML.

XUL face parte din *Mozilla Application Framework* alături de alte componente precum: *Gecko* (un motor de randare/editare a conținutului web), *WebServices* (Suport pentru XMLHttpRequest, XML-RPC, SOAP, WSDL), *XPIInstall* (un installer independent de platforma pentru instalarea și packagingul aplicațiilor Mozilla), *XPCOM* (pentru adăugarea de noi funcționalități sau conectarea cu alte componente), *Necko* (bibliotecă cu suport pentru majoritatea protocoalelor de rețea).

1.2. Contextul apariției

XUL a fost creat pentru în scopul de a face dezvoltarea browserului Mozilla mai rapidă și mai ușoară. Fiind un limbaj bazat pe XML, toate avantajele XML pot fi folosite și la XUL.

1.3. Scopul apariției

Mozilla, folosind XUL permite dezvoltatorilor de software să dezvolte aplicații independente de platformă a căror interfață cu utilizatorul să atingă nivelul interfețelor de aplicații desktop. Experiența designerilor în dezvoltarea aplicațiilor Web folosind tehnologii standard W3C poate fi folosită în dezvoltarea aplicațiilor desktop.

Înainte ca web-ul să devină un standard pentru transmiterea aplicațiilor între utilizatori aflați la distanță, această problemă era rezolvată prin aplicații client-server independente de platformă. Dar acestea costau mult, se bazau pe licențe și duceau la o dependență de producător. Mozilla realizează același lucru cu o licență open-source.

Deoarece este o tehnologie cross-platform care se dezvoltă rapid, XUL ar fi putea fi folosit cu Java J2EE sau Microsoft .Net care se axează pe dezvoltarea accesului prin rețea a obiectelor distribuite. Logica unei aplicații poate avea mult de câștigat de pe urma unui layer separat de interfața cu utilizatorul care are la bază un standard, este extensibil și distribuibil prin Internet.

XUL *reduce diferența dintre aplicațiile desktop și aplicațiile de browser*. Unele aplicații web pot fi îmbunătățite prin migrarea lor în XUL datorită capabilităților avansate de interfață, implementării consistente a specificațiilor de platformă și a accesului la resursele native cum ar fi sistemul local de fișiere sau librăriile partajate.

1.4. Funcționalități generice

XUL oferă posibilitatea de a crea aproape toate elementele regăsite în interfețele grafice moderne. Este destul de generic ca să poată fi folosit pentru necesitățile specifice ale anumitor echipamente și destul de puternic pentru a permite crearea de interfețe sofisticate.

Pe scurt:

- XUL este un limbaj independent de platformă pentru definirea interfețelor grafice
- A fost creat pentru a ușura dezvoltarea browserului Mozilla
- Interfețele din Mozilla Firefox și Thunderbird folosesc XUL
- Dispune de avantajele limbajelor XML. XHTML, SVG
- Se pot crea extensii la browserele Mozilla și Firefox
- Se pot crea aplicații, precum Mozilla
- Obținerea de elemente din XUL, folosind DOM din javascript;
- Creare de liste tabulare sau ierarhice folosind arbori;
- Definire de stylesheet-uri și de skin-uri, asigurarea localizării prin folosirea entităților și a documentelor DTD.

2. Sintaxa

2.1. Elemente de bază ale sintaxei

XUL este un limbaj bazat pe XML. Ca orice aplicație XML, XUL aderă la specificația W3C XML 1.0 specification și urmează regulile de sintaxă definite acolo.

Widgetii (window + gadget) sunt componentele interfeței grafice. Meniuri, toolbaruri, butoane, scrollbaruri sunt widgeti, precum și elementele box sau spring. Acești widgeti pot fi puși într-un fișier XUL ca simple elemente: <menu>, <toolbar>. Sintaxa pentru widgeti este bazată în cea mai mare parte pe XML. Totalitatea widgets-ilor formează XPToolkit.

Cele patru reguli de sintaxa esențiale în XUL:

1. Toate atributele și evenimentele trebuie să fie scrise cu litere mici
2. Toate șirurile de caractere trebuie puse în ghilimele
3. Fiecare widget XUL trebuie să aibă tagul închis sau să fie bine format, după cum e cazul.(<tag></tag> sau <tag/>)
4. Toate atributele trebuie să aibă o valoare

Ca o observație, XUL este strict cu sintaxa și respectarea regulilor. În majoritatea cazurilor nu va afișa nimic dacă un element sau altul a fost scris greșit.

Elementul rădăcina al unui fișier XUL și nivelul de vârf al interfeței este elementul <window>. Toate fișierele XUL trebuie să conțină neapărat un singur element window. Toată interfața XUL este definită în interiorul acestui element rădăcina. În general, o interfață XUL este alcătuită dintr-o combinație de meniuri, toolbar-uri și "content area".

Un fișier XUL tipic poate defini o simpla fereastră cu structura următoare (chiar dacă nu e validă tehnic):

```
<window>
  <menubar>
    <menu label="..." />
    ...
  </menubar>
  <box>
    <browser />
    ...
  </box>
</window>
```

Fiecare copil al elementului <window> (menu bar sau box) poate să conțină la rândul sau alte elemente. XUL oferă o imbricare absolută a elementelor, astfel că toată interfața este construită pornind de la elementul de baza <window>.

Lipsa de detaliu a exemplului dat mai sus face ca bucata de cod să nu fie folositoare într-o aplicație. De exemplu, widgetul <box> ar trebui identificat printr-un id și poziționat, widgetului <window> ar trebui să i se dea un titlu și diferiți "event listereri". Widgetul box ar putea avea definit și atributul

"flex" care este folosit pentru așezarea în interfață. Alte atribute, cum ar fi valorile pentru item-urile din meniu sau declarațiile de namespace pentru <window> sunt obligatorii și invalidează interfața dacă nu apar definite.

În specificația XUL, sunt descrise toate valorile și atributele pe care widget-urile din XToolkit le pot avea, precum și genealogia și structura documentelor XUL valide. Aceste categorii de informații se regăsesc sub denumirea de Attributes and Common Children pentru fiecare XUL widget.

O lista de grupuri de widgeti XUL este următoarea:

- window
- box
- menus and menubars
- toolboxes and toolbars
- tab widget
- checkbox
- titled buttons
- scrollbar
- splitter
- progressmeter

2.2. Organizarea unei interfețe XUL

Pot fi mai multe fișiere adiționale care sunt folosite de fișierul XUL pentru a crea o singură fereastră. Aceste fișiere sunt referite ca fiind un pachet, iar acest pachet include de obicei style-sheetul pentru XUL, un fișier DTD care conține declarații de entități pentru șirurile localizabile din interfață și fișiere auxiliare XUL care contribuie la interfața XUL din fișierul de bază. Mai pot fi și fișiere pentru imagini sau date specifice platformei. Fiecare director pachet are de obicei trei subdirectoare, content, skin și locale, în care sunt definite XUL, CSS și informația de localizare:

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

O mica colecție de atribute sunt generale la toate widgeturile, dar nu sunt moștenite de la widgeti generali sau instrumente precum <box>. Pot fi imaginate ca fiind moștenite de la o clasă de bază anonimă. Din acestea face parte și atributul "id", cu care fiecare widget XUL poate fi specificat. Aceste atribute sunt specificate într-un fișier common. Din fișierul common mai fac parte și următoarele atribute: context, style, persist, popup, ondragexit, onclick, oncommand, onkeypressed.

Acolo unde atributele comune sunt implementate sau folosite diferit de fiecare widget în parte, acest lucru este specificat în documentație pentru fiecare widget.

Containerele de layout de baza sunt: box, stack, grid, popup iar containerele rădăcină ale documentului sunt: window, wizard, dialog, page, overlay.

3. Cazuri de utilizare

3.1. XUL ca o platforma de aplicații

XUL și Gecko sunt o alegere foarte buna pentru realizarea de aplicații Web sofisticate. Oferă un toolkit de interfață cu utilizatorul bogat, un renderer de HTML și CSS cu suport pentru servicii web ,totul fiind independent de platformă.

Platforma Gecko este în jur de 5 pana la 10 MB, ceea ce este foarte puțin în comparație cu alte platforme de aplicații. Gecko poate fi rulat și de pe un drive de rețea sau de pe CD.

Deoarece XUL poate fi folosit la situri Web, poate fi folosit și în arhitecturile pe partea de server, cum sunt PHP și JSP pentru realizarea conținutului dinamic. În momentul de față sunt în dezvoltare proiecte care tind să integreze Java, Python sau alte limbaje în Gecko direct.

3.2. De ce XUL?

XUL este un limbaj de markup pentru crearea de interfețe dinamice cu utilizatorul. Este parte integrantă a *browserului Mozilla* și este disponibil și ca parte din Gecko. Este construit astfel încât să fie portabil și funcționează pe toate versiunile de Windows, Macintosh, Linux sau Unix. Cu XUL și componentele Gecko se pot crea aplicații sofisticate fără alte unelte speciale.

XUL a fost creat pentru construirea interfețelor aplicațiilor Mozilla, acestea incluzând web browserul, clientul de mail și editorul și extins ulterior pentru descrierea oricărei interfețe.

La fel ca în HTML, și în XUL poți crea interfețe folosind limbaj de markup, să integrezi CSS și să folosești JavaScript. De asemenea există acces la interfețe pentru citirea sau scrierea conținutului la distanță sau pentru apel de servicii web. Spre deosebire de HTML, XUL oferă o gamă mai bogată de componente de interfață (widgeti).

XUL oferă avantajul de a nu scrie în cod blocuri mari de JavaScript, spre exemplu, doar pentru a afișa un meniu popup, deoarece sunt gata integrate. Mai mult, elementele interfeței sunt în așa fel făcute în cat să aibă același 'look and feel' ca platforma folosită, asta dacă nu dorește propriul 'look and feel' al aplicației. Widgetii XUL suportă de asemenea localizare.

3.3. Standarde și alte tehnologii suportate

Mai jos se afla o lista a tehnologiilor suportate de Gecko și care pot fi folosite de aplicațiile XUL:

- HTML 4.0 and XHTML
- CSS2, CSS3
- DOM2, DOM3
- XML
- MathML
- RDF (Resource Description Framework)
- XSLT
- XPath
- Simple XLinks
- XPointer
- XML Base
- XML-RPC
- SOAP (Simple Object Access Protocol)
- WSDL (Web Services Description Language)
- ECMAScript/Javascript
- XBL (eXtensible Bindings Language)
- GIF, JPG, PNG, BMP, ICO images
- HTTP 1.1

3.4. Exemple

Aplicatia HelloWorld.xul:

```
<window xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <box align="center" pack="center" flex="1">
    <description>Hello, world</description>
  </box>
</window>
```

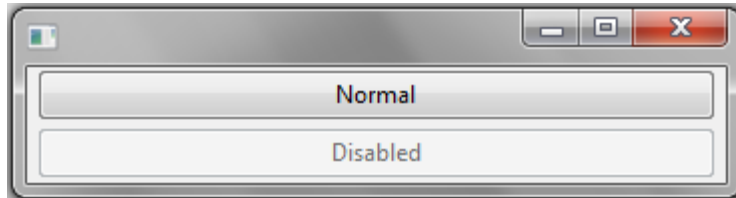
O simpla fereastră Window.xul:

```
<?xml version="1.0"?>
<?xmlstylesheet href="chrome://global/skin/" type="text/css"?>
<window id="findfile-window" title="Find Files" orient="horizontal"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  ...
```

```
</window>
```

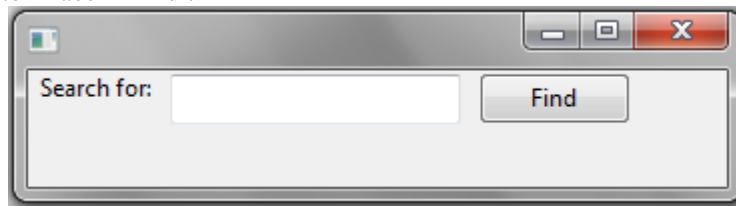
Adăugarea a două butoane la fereastră:

```
<button label="Normal"/>  
<button label="Disabled" disabled="true"/>
```



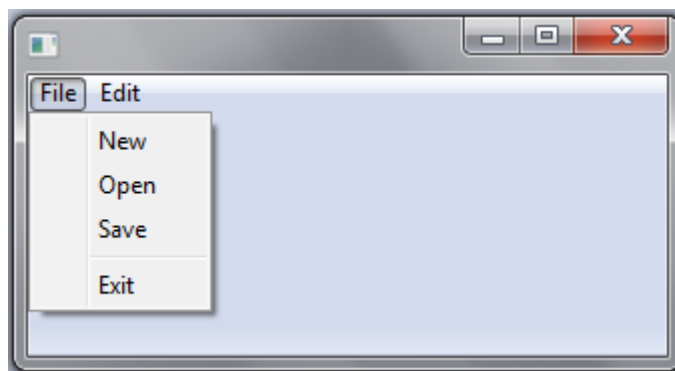
Un câmp de căutare:

```
<label value="Search for:" control="find-text"/>  
<textbox id="find-text"/>  
<button id="find-button" label="Find"/>
```



Un meniu:

```
<toolbox flex="1">  
  <menubar id="sample-menubar">  
    <menu id="file-menu" label="File">  
      <menupopup id="file-popup">  
        <menuitem label="New"/>  
        <menuitem label="Open"/>  
        <menuitem label="Save"/>  
        <menuseparator/>  
        <menuitem label="Exit"/>  
      </menupopup>  
    </menu>  
    <menu id="edit-menu" label="Edit">  
      <menupopup id="edit-popup">  
        <menuitem label="Undo"/>  
        <menuitem label="Redo"/>  
      </menupopup>  
    </menu>  
  </menubar>  
</toolbox>
```

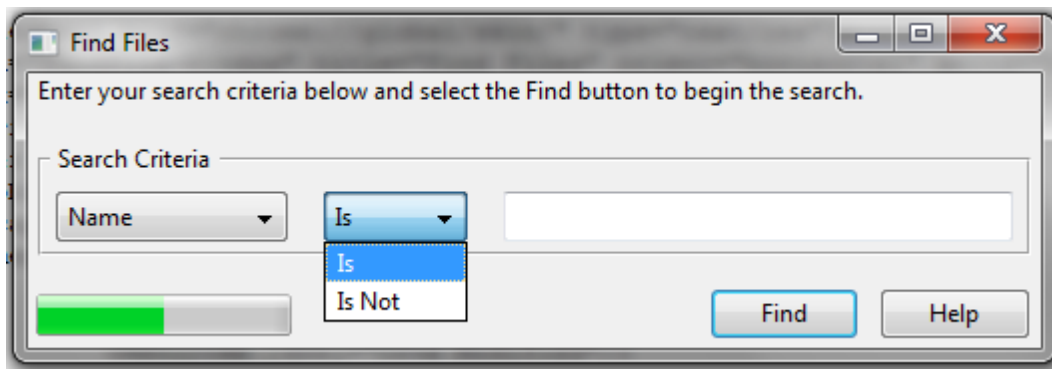



Un exemplu care integrează mai multe funcționalități:

```

<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window id="findfile-window" title="Find Files" orient="horizontal"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
<vbox flex="1">
  <description>Enter your search criteria below and select the Find button to begin the
search.</description>
  <spacer style="height: 10px"/>
  <groupbox orient="horizontal">
    <caption label="Search Criteria"/>
    <menulist id="searchtype">
      <menupopup>
        <menuitem label="Name"/> <menuitem label="Size"/>
        <menuitem label="Date Modified"/>
      </menupopup>
    </menulist>
    <spacer style="width: 10px;"/>
    <menulist id="searchmode">
      <menupopup>
        <menuitem label="Is"/>
        <menuitem label="Is Not"/>
      </menupopup>
    </menulist>
    <spacer style="width: 10px;"/>
    <textbox id="find-text" flex="1" style="min-width: 15em;"/>
  </groupbox>
  <spacer style="height: 10px"/>
  <hbox>
    <progressmeter value="50%" style="margin: 4px;"/>
    <spacer flex="1"/>
    <button id="find-button" label="Find" default="true"/>
    <command id="cmd_openhelp" oncommand="alert('Help');"/>
    <button id="help-button" label="Help" command="cmd_openhelp"/>
  </hbox>
</vbox>
</window>

```



4. Instrumente

XULExplorer este un o interfață de GUI permite drag-and-drop pentru widgeti pentru a construi o interfata XUL, precum și vizualizare, respectiv verificare de erori la nivel XUL sau JS.

5. Concluzii

Aplicațiile XUL pot fi create foarte rapid și ușor, sunt independente de platformă, nu necesită instrumente sofisticate de lucru, pot fi personalizate rapid în funcție de scopul lor și oferă o gamă bogată de componente de interfață la fel ca orice aplicație desktop modernă.

6. Bibliografie

https://developer.mozilla.org/en/XUL_Explorer
https://developer.mozilla.org/En/XUL_Tutorial
https://developer.mozilla.org/en/XUL_Reference