

Definizione sotto Mozilla

- XUL (XML User Interface Language)
 - https://developer.mozilla.org/En/XUL

0

Ø

- https://developer.mozilla.org/en/XUL_Tutorial
- Utilizzo flessibile di risorse di vario tipo (icone, script, ecc.) I file XUL possono essere referenziati con una URL HTTP regolare (o con altri tipi di URL) come i file HTML...
- ... tuttavia i file che sono installati all'interno del sistema chrome di Mozilla possono essere referenziati attraverso speciali URL di tipo chrome (chrome://...). In pratica esiste in quel caso una directory chrome sotto la quale si trovano le definizioni XUL
 - https://developer.mozilla.org/En/Chrome_Registration
 - http://developer-stage.mozilla.org/en/XUL_Tutorial/XUL_Structure

Laboratorio di Sistemi Interattivi

Chrome

- Chrome è l'insieme di elementi di interfaccia della finestra di applicazione che sono al di fuori dell'area di contenuto della finestra: Toolbar, barre di menu, progress bar e barre dei titoli sono tutti esempi di elementi che sono tipicamente parte di un chrome.
- Un chrome provider fornisce il chrome per un certo tipo di finestra (ad esempio la finestra di un browser). I provider lavorano insieme per fornire un insieme completo di chrome per una particolare finestra, dalle immagini sui bottoni della toolbar ai file che descrivono il testo, il contenuto e la presentazione della finestra.

 - Content: la tonte principale del file per la descritazione di una finestra proviene da un package di tipo content Locale: le applicazioni localizzabili mantengono tutta l'informazione localizzata in provider (package) locale; ciò permette ai traduttori di inserire (plug in) un chrome package diverso per tradurre una applicazione senza alterare il resto del codice sorgente:
 - Skin: uno skin provider deve fornire un set completo di file per descrivere l'apparenza visiva del chrome; tipicamente il package contiene file CSS e immagini

Laboratorio di Sistemi Interattivi



Caratteristiche Linguaggio per la definizione di interfacce utenti portabili

- ٠ XUL stabilisce una chiara separazione tra la definizione della logica di programmazione (implementata in JavaScript) da una parte, e la logica di presentazione dall'altra (XUL, XBL, CSS). .
- presentazione dai altra (XUL, XBL, CSS). XBL: eXtensible Binding Language è un linguaggio di markup che definisce nuovi elementi speciali, o "binding" per i widget XUL. Con XBL, gli sviluppatori possono definire nuovi contenuti, aggiungere event handlers, o aggiungere nuove proprietà e metodi all'interfaccia. Essenzialmente, XBL rafforza la possibilità degli sviluppatori di estendere XUL mediante la personalizzazione dei tag esistenti, creando nuovi tag proprietari. Elementi definibili . Elementi definibili
 - Controlli di ingresso come campi di testo e checkbox Toolbar con pulsanti o altro contenuto

 - Menu su una barra di menu o su menu pop up Dialoghi con tab Alberi per informazione gerarchica o tabellare Scorciatoie di tastiera

Laboratorio di Sistemi Interattivi

Organizzazione I

- · Ogni file XUL organizzato in 4 subfile
 - layout and elements
 - style declarations
 - entity declarations (per localizzazione)
 - scripts

0

0

- · Eventuale file aggiuntivo per specifica di file esterni per immagini o dati specifici alla piattaforma
- **XUL Element Reference** https://developer.mozilla.org/en/XUL Reference
- XUL Tutorial https://developer.mozilla.org/en/XUL_Tutorial









Posizionamento di default

Dimensioni preferite dell'elemento <vbox> <hbox></hbox></vbox>	Login 4, 2 2 2 Login: Anno 2010 Password: Anno 2010
<a>label control="login" value="Login"	:"/> ок
<textbox id="login"></textbox>	Cancel
<hbox></hbox>	
<a>label control="pass" value="Past textbacking"	ssword:"/>
<textbox id="pass"></textbox> < button id="ok" label="OK"/>	hbox = elementi contenuti posizionati in ordine di apparizione in orizzontale
<button id="cancel" label="Cancel"></button>	vbox = elementi contenuti posizionati
	in ordine di apparizione in verticale
	per default non si cura l'allineamento
Discontinuedo di	ooratorio di Sistemi Interattivi

Allineamenti I	
Ottenibili con box aggiuntive vbox> chbox> clabel control="login" va clabel control="pass" va	alue="Login:"/> alue="Password:"/>
 <textbox id="login"></textbox> <textbox id="pass"></textbox> <button id="ok" label="OK"></button> <button id="cancel" label="Cancel"></button> 	Login Login: Password: OK Cancel
Dipartimento di	Laboratorio di Sistemi Interattivi







Altri tipi di gestione

- Stack: elementi visualizzati tutti insieme come pila, con il primo in basso
- Deck: come lo stack, ma visualizzati uno per volta
- Tabbox: contenitore per pagine tabbed, mostra una fila di tab che possono essere selezionati
- · Grid: elementi disposti in righe e colonne

Laboratorio di Sistemi Interattivi







Gestione di eventi • Definita con script

· Possibilità di utilizzare diversi file di script <script src="findfile.js"/> <script src="chrome://findfiles/content/help.js"/> <script src="http://www.example.com/js/items.js"/>

- · Eventi definiti come DOM Events
- E' possibile "attaccare" dinamicamente eventi ai widget tramite dei listener torio di Sistemi Interattivi 0

Fasi di gestione dell'evento • Generato in un elemento

- Fase di cattura
 - Inviato alla finestra
 - Al documento
 - A ogni antenato dell'elemento dall'alto verso il basso - Inviato all'elemento
- · Fase di bubbling A ritroso
- · Gestione dell'evento può avvenire in una delle due fasi
- Terminata la propagazione si effettuano azioni di default previste dal comportamento implicito dell'elemento • Labo io di Si Ø





Esempio		
xml version="1.0"?	<menuitem oncommand="alert('New item</td></tr><tr><td><?xml-stylesheet</td><td><pre><menuitem label=" open"=""/>	
href="chrome://global/skin/"	<menuitem css"?="" oncommand="alert('Close</td></tr><tr><td>type=" text=""/>	handler')" label="Close" />
<window <="" id="events" td=""><td> </td></window>	 	
xmlns:html="http://www.w3.org/1999/xhtml "	<menu class="menu" label="Edit"> <menupopup> <menuitem edit<="" oncommand="alert/" td=""/></menupopup></menu>	
xmlns="http://www.mozilla.org/keymaster/g atekeeper/there.is.only.xul" oncommand="alert('Window handler')">	Source handler')" label="Edit Source" /> <menuitem label="Reload"/> <menuitem label="View Source"/>	
, , , ,		
<vbox></vbox>		
<vdox <="" style="background-color: lightgrey;" td=""><td>-http://</td></vdox>	-http://	
<pre>cmcommand= alen(BOX handler) > <pre><menu <="" class="menu" label="File" pre=""></menu></pre></pre>	<pre> <spring flex="1"></spring></pre>	
oncommand="alert('Menu handler')">		
<menupopup></menupopup>		







Cattura

Qualunque elemento XUL può usare il metodo DOM addEventListener per registrarsi per catturare eventi. La sintassi in XUL è:

element = document.getElementByld("id of the element"); element.addEventListener(event name, event handler function, whether to register a capturing listener);

si possono catturare eventi specificando un valore true per il parametro useCapture parameter del metodo addEventListener

var bbox = document.getElementById("bigbox");

if (bbox)

bbox.addEventListener("click", function() { alert('captured'); }, true);

<box id="bigbox">
<menu value="File">

<menupopup>

<menuitem value="New" onclick="alert('not captured')" />

<menupopup>

</menu> </box>



Esempio oncommand="alert(event.target.tagName);"> <button label="OK"/> <checkbox label="Show images"/>

</vbox>

0

<vbox

In questo esempio, l'evento command "risalirà" dal bottone o dalla checkbox verso la vbox, dove verrà gestito.

Laboratorio di Sistemi Interattivi



Associazione di comandi <command id="cmd_openhelp" oncommand="alert('Help!');"/> <button label="Help" command="cmd_openhelp"/> Dosibilità di riuso