



Rich Web Interfaces

Philip Fearheller



Agenda

- Intro: Rich ain't pretty
- What's missing?
- The Contenders
- DTML and Javascript, simple to complex
- XUL... there is only XUL
- SVG: Images in Action
- Flash: OpenLaszlo
- Swing and other Java UI Toolkits

DHTML and Javascript

[Simple hide and show](#)

[Remote Data Access](#)

[Putting it together...](#)

XUL, XML UI Language

[Find Files...](#)

[Editable Table](#)

[A complete Timesheet UI](#)

SVG: Images in Action

[Using images](#)

[Selling out my coworkers!](#)

Flash: OpenLaszlo

[More Dynamic](#)

[Bugs...](#)

Active Form Example

Projects:

Mattinlv (Order Entrv) ▼

Developer: Al Iacovella

Bugs: Add On Reason Column Sorting ▲

Dupe Cust Acnt No

Remove Frn Grp From Corp

Build Hier Error When No Acnt#

Hierarchy Setup No Contact Sav ▼

Date: 2003-04-14 00:00:00.0

Description:

Column sorting via clicking the header column throws an ArrayIndexOutOfBoundsException exception.

DHTML and Javascript

[Simple hide and show](#)

[Remote Data Access](#)

[Putting it together...](#)

XUL, XML UI Language

[Find Files...](#)

[Editable Table](#)

[A complete Timesheet UI](#)

SVG: Images in Action

[Using images](#)

[Selling out my coworkers!](#)

Flash: OpenLaszlo

[More Dynamic](#)

[Bugs...](#)

WEATHER DHTML EXAMPLE!

Current Conditions Forecast

Current Conditions

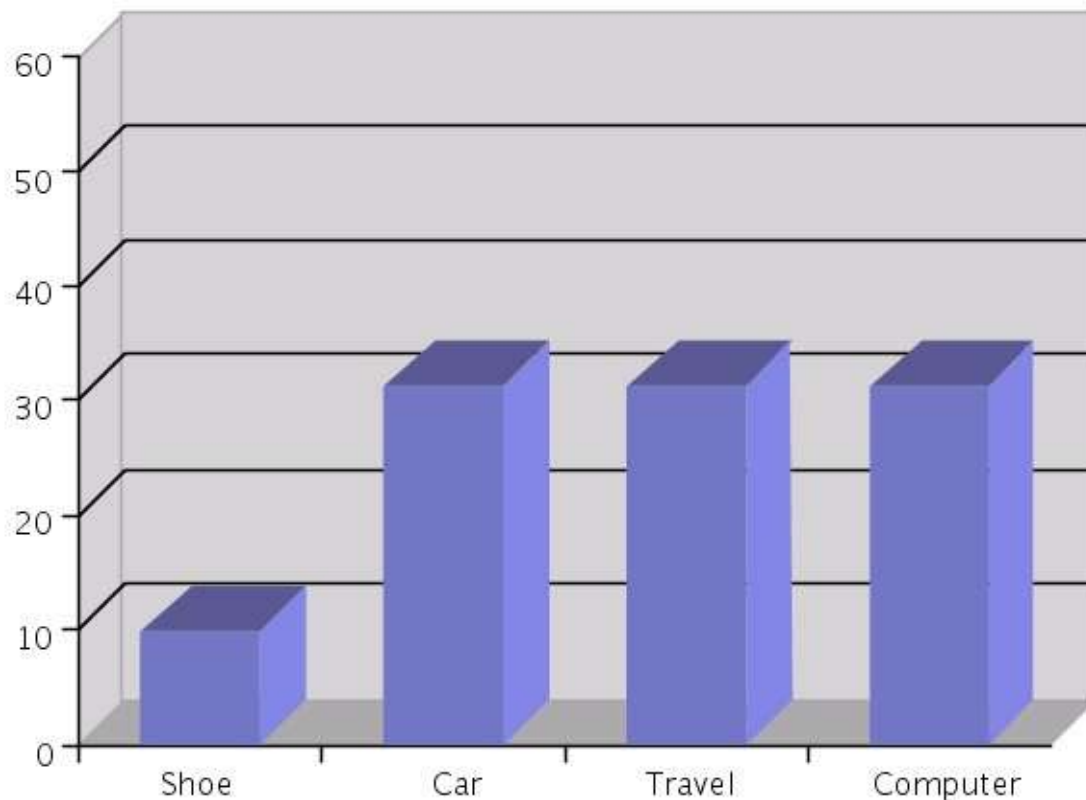
WEST CHESTER, PA



46°F

FOG/MIST

Humidity	89 %
Barometer	29.96" (1014.5 mb)
Windspeed	Calm
Dewpoint	43°F (6°C)
Heatindex	None
Last Updated	Mar 20, 2:54 pm EST

DHTML and Javascript[Simple hide and show](#)[Remote Data Access](#)[Putting it together...](#)**XUL, XML UI Language**[Find Files...](#)[Editable Table](#)[A complete Timesheet UI](#)**SVG: Images in Action**[Using images](#)[Selling out my coworkers!](#)**Flash: OpenLaszlo**[More Dynamic Bugs...](#)**Interactive SVG Bar Char**

DHTML and Javascript

- [Simple hide and show](#)
- [Remote Data Access](#)
- [Putting it together...](#)

XUL, XML UI Language

- [Find Files...](#)
- [Editable Table](#)

[A complete Timesheet UI](#)

SVG: Images in Action

- [Interacting with images](#)
- [Selling out my coworkers!](#)

Flash: OpenLaszlo

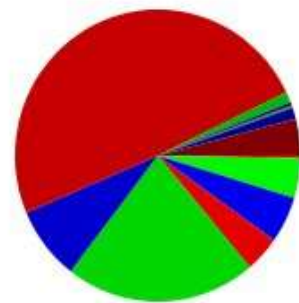
- [More Dynamic Bugs...](#)

Dynamic SVG Pie Char

Developers:

- Keavney, Pete
- Confino, Joel
- Dummy, Test
- Mulder, Erin
- Iacovella, Al
- Swartley, Matt
- Wright, Andrea O. K.
- Welson-Rossman, Tracey
- Canfield, Dave
- Mulder, Aaron

Chart Bugs



- Total Bugs: 1360
- Don Coleman
 - Tom Purcell
 - Andrea O. K. Wright
 - Al Iacovella
 - Erin Mulder
 - John Shepard
 - Aaron Mulder
 - Ricky Williams
 - Matt Swartley
 - Gina Rappaport
 - Elisia Silverstone
 - Melissa Rivers
 - Dave Canfield

DHTML and Javascript

- [Simple hide and show](#)
- [Remote Data Access](#)
- [Putting it together...](#)

XUL, XML UI Language

- [Find Files...](#)
- [Editable Table](#)
- [A complete Timesheet UI](#)

SVG: Images in Action

- [Using images](#)
- [Selling out my coworkers!](#)

Flash: OpenLaszlo

- [More Dynamic Bugs...](#)

File Edit

Open Save

Search Options

Enter your search criteria below and select the Find button to begin the search.

Search Criteria

Filename	Location	Size
mozilla	/usr/local	2520 bytes

DHTML and Javascript

- [Simple hide and show](#)
- [Remote Data Access](#)
- [Putting it together...](#)

XUL, XML UI Language

- [Find Files...](#)
- [Editable Table](#)
- [A complete Timesheet UI](#)

SVG: Images in Action

- [Using images](#)
- [Selling out my coworkers!](#)

Flash: OpenLaszlo

- [More Dynamic Bugs...](#)



Week Start:

Employee:

Billing type

- Regular Hours
- Holiday Hours
- Sickday Hours
- Overtime Hours

Client	Project	Task	Mo	Tu	We	Th	Fr	Sa	Su	Comments
Mattingly	Order Entry		6	5	3	7	8			writing specs
Mattingly	Order Entry		6	5	3	7	8			writing specs
Mattingly	Order Entry		6	5	3	7	8			writing specs
Mattingly	Order Entry		6	5	3	7	8			writing specs
Mattingly	Order Entry		6	5	3	7	8			writing specs

Applications

Jellico

Bug Tracking Projects Timesheet

Bug #: Search

Project: Deliciousio (Order Entry)

	Bug #	Developer	Description
	802	Al Iacovella	Add On Reason Column Sorting
	790	Andrea O. K. Wright	Dupe Cust Acnt No
	760	Andrea O. K. Wright	Remove Frn Grp From Corp
	754	Andrea O. K. Wright	Build Hier Error When No Acnt#
	748	Andrea O. K. Wright	Hierarchy Setup No Contact Sav
	752	Andrea O. K. Wright	Build Hier No Show Cust
	755	Andrea O. K. Wright	Build Hier Sys Error No Acnt#
	753	Andrea O. K. Wright	Build Hier Shrink Acnt Level

Chariot People

contacts

Rappaport, Mike

Richardson, Mike

Coleman, Don



dcoleman@chariotsolutions.com
165 Indiana Ave
<http://www.chariotsolutions.com>

details

Duck, Donald

Rivers, Joanne

Manning, Peyton

Silverstone, Elisia

Rivers, Melissa



Rich ain't pretty...

- ... and pretty ain't rich!
- Talking about web application functionality, not glitz
- Using Flash but not for the 'Flash'
- Adding features found in traditional fat clients
 - ✓ Advanced UI Controls (trees, tabs, tables, sliders)
 - ✓ Improve usability
 - ✓ Context sensitivity
- Marrying “web experience” with application usability
 - ✓ Support the Back button!



What's Missing

- From the user's perspective
 - ✓ Drag and Drop
 - ✓ Dynamic form and data
 - ✓ Interactive tables
 - ✓ Live data
 - ✓ Tool bars and menu bars
 - ✓ Interactive graphics
 - ✓ Context sensitive menus



What's Missing (con't)

- From the developer's perspective
 - ✓ Event driven development
 - ✓ Full featured common controls
 - ◆ Trees
 - ◆ Tables
 - ◆ Sliders
 - ◆ Tabs
 - ◆ Progress Meters
 - ✓ Data binding
 - ✓ State management



Why is it Missing?

- Inherent limitations of using a content browser as a User Interface Device
- HTTP is a stateless protocol
- Lowest common denominator syndrome
- Browsers wars and control
- To plug-in or not to plug-in



The Contenders

- DHTML and Javascript
 - ✓ Cross browser support getting better
 - ✓ Well known by developers
 - ✓ Bolstered by XMLHttpRequest and Google
 - ✓ Inherent Browser Limitations
 - ✓ Google Maps, Suggestions, GMail
- XUL, XML UI Language
 - ✓ Full feature UI Controls
 - ✓ Many growing implementations: Mozilla, Thinlets, Luxor, SwixML
 - ✓ Drag and Drop, with targets
 - ✓ Firefox, Thunderbird



The Contenders (con't)

- SVG: Images in Action
 - ✓ Open Standard from W3C
 - ✓ Adobe and Batik
 - ✓ Interactive Shipping Maps
- Flash: OpenLaszlo
 - ✓ XUL like schema
 - ✓ Full featured UI using standard plugin
 - ✓ Data driven application
 - ✓ Closed Flash bytecode



The Contenders (con't)

- Swing and other Java UI Toolkits
 - ✓ Full event driven User Interface
 - ✓ Java plugin and Web Start help
 - ✓ Same language as back end
 - ✓ Not consistent with Web experience
 - ✓ Swing, SWT



DHTML and JavaScript

- Simple screen manipulation
- Hiding and showing controls
- Moving data between controls
- Changing graphics based on events



Simple hide and show

```
<select name="sel"
  onchange="hideShow(this.options[this.selectedIndex].value);">
...
<div id="button">button Control:
  <input type="radio" value="button"/></div>
...
function hideShow(id) {
  var button = document.getElementById("button");
  var text-box = document.getElementById("text-box");
  var multi-select = document.getElementById("multi-select");

  button.style.visibility = "hidden";
  text-box.style.visibility = "hidden";
  multi-select.style.visibility = "hidden";

  button.style.display = "none";
  text-box.style.display = "none";
  multi-select.style.display = "none";

  var target = document.getElementById(id);
  target.style.visibility = "visible";
  target.style.display = "inline";
}
```



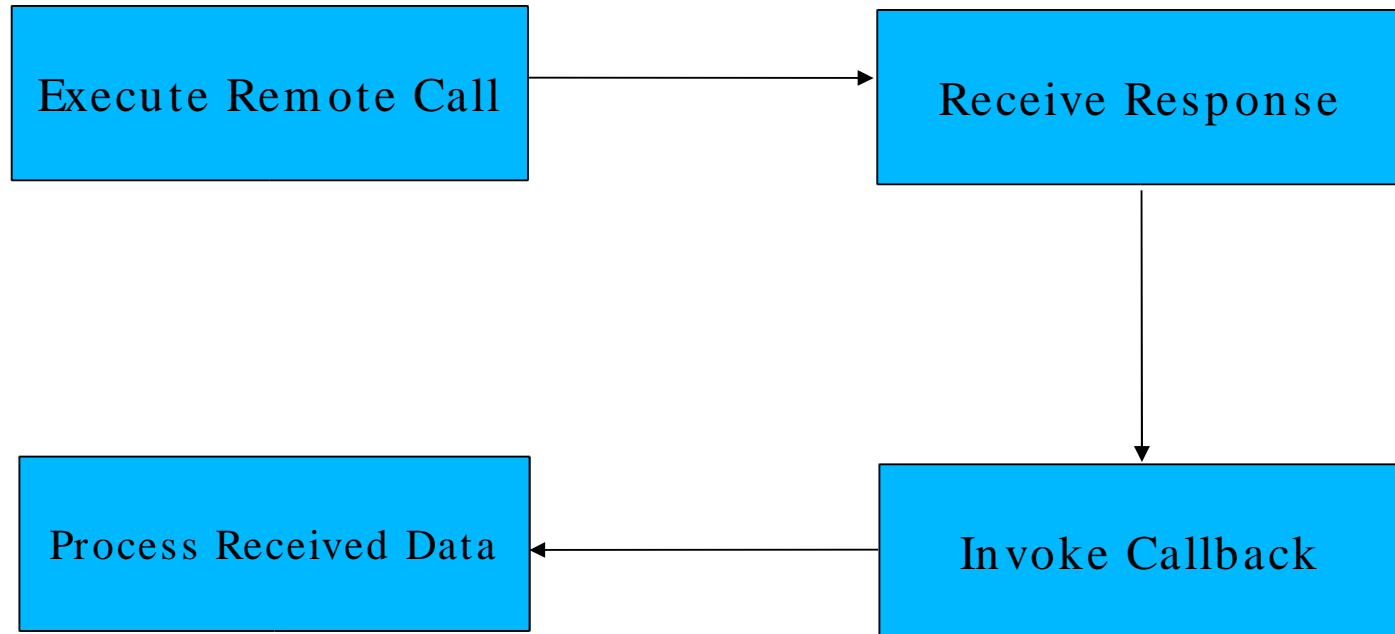
Remote data access

- Using XMLHttpRequest
- Transferring XML only... reduces bandwidth
- No screen refresh... improved usability
- From Simple XML generation to SOAP



Remote Data Access

- Remote Data Access Flow





Remote Data Access

```
Projects: <select name="projects"
  onchange="loadBugs(
    this.options[this.selectedIndex].value);">
</select>

...

<select size="5" name="bugs"
  onchange="loadBugInfo(
    this.options[this.selectedIndex].value);">
</select>

...

Developer: <input type="text"
  name="developer" value=""/>
Date: <input type="text" name="cdate" value=""/>
Description:

<textarea id="desc" rows="10" cols="80"></textarea>
```



Remote Data Access

```
var xmlhttp;

function initRemote() {
    try {
        xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {

        try {
            xmlhttp = new ActiveXObject
                ("Microsoft.XMLHTTP");
        } catch (E) {
            xmlhttp = false;
        }
    }

    if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
        xmlhttp = new XMLHttpRequest();
    }
}
```



Remote Data Access

```
function send(str, func) {
    xmlhttp.open("GET", str ,true);
    xmlhttp.onreadystatechange= func;
    xmlhttp.send(null);
}

function loadProjects() {
    var func = function() {
        if (xmlhttp.readyState==4) {
            var select = document.activeform.projects;
            var div = document.createElement("div");
            Gparse(document, div, xmlhttp.responseText);
            //var div = xmlhttp.responseXML.documentElement;
            var list = div.getElementsByTagName("project");
            for(var i =0; i < list.length; i++) {
                select.options.add(new Option(
                    list[i].getAttribute("name"),
                    list[i].getAttribute("id")));
            }
        }
    };

    send("/lps-2.2.1/my-apps/programs/jellico-projects.jspx", func);
}
```



Remote Data Access

```
function loadBugInfo(bugId) {
    var func = function() {
        if (xmlhttp.readyState==4) {
            clearForm();
            var div = document.createElement("div");
            Gparse(document, div, xmlhttp.responseText);
            var bug = div.getElementsByTagName("bug").item(0);

            document.activeform.desc.value =
                bug.getAttribute("desc");

            document.activeform.developer.value =
                bug.getAttribute("developer");
            document.activeform.cdate.value =
                bug.getAttribute("date");
        }
    };
    send("/lps-2.2.1/my-apps/programs/jellico-bug-
        info.jspx?bug_id=" + bugId, func);
}
```




Remote Data Access

```
<jsp:root
...
  version="2.0">

<sql:setDataSource
  var="example"
  driver="org.postgresql.Driver"
/>

<sql:query var="people" dataSource="${example}">
  SELECT bug.id, bug.name, person.first_name, person.last_name
  FROM bug, person
  WHERE bug.developer_id = person.id and project_id =
  <jsp:expression>request.getParameter("project_id")
  </jsp:expression>
</sql:query>
<bugs>
  <c:forEach var="row" items="${people.rows}">
    <bug id="${row.ID}" name="${row.NAME}"
      date="${row.CREATION_DATE}"
      developer="${row.FIRST_NAME} ${row.LAST_NAME}" />
  </c:forEach>
</bugs>
</jsp:root>
```



Putting it all together

- Screen manipulation with data access
- Accessing a remote site
- Could have used an applet
- Useful for utilities on larger pages
- The way portlet should be...



Putting it together...

```
<table>
  <tr>
    <td><a href="javascript:showCurrent();">
      Current Conditions</a></td>
    <td><a href="javascript:showForecast();">Forecast</a></td>
  </tr>
</table>

<div id="weather_div" style="position: absolute; x: 0; ">
  <div class="title"><u>Forecast</u></div>
  <table id="target">
  </table>
</div>

<div id="current_div" style=
  "position: absolute; x: 0;
  visibility: visible; display: inline">

  <div class="title"><u>Current Conditions</u></div>
  <table id="current_target">
  </table>
</div>
```



Get the data

```
function loadWeather() {
  var func = function() {
    if (xmlhttp.readyState==4) {
      var select = document.activeform.projects;
      var div = document.createElement("div");
      Gparse(document, div, xmlhttp.responseText);
      buildForecast(div);
      buildCurrent(div);      }
    };

    send("/richweb/weatherproxy.jspx?zip=" +
        document.forms[0].zip.value,true);, func);
  }
}
```



Clearing the way

```
function buildForecast(div) {
    var list = div.getElementsByTagName("day");
    var count = list.length;
    var weather = document.getElementById("weather_div")
    var target = document.getElementById("target")

    if(target) {
        var n = document.createElement("table");
        n.setAttribute("id", "target");
        weather.replaceChild(n, target);
        target = n;
    }
    ...
}
```



What's today going to be like?

```
function buildCurrent(div) {
    var currentElement =
        div.getElementsByTagName("current").item(0);
    ...
    row = target.appendChild(document.createElement("tr"));
    col = row.appendChild(document.createElement("td"));
    var img = col.appendChild(document.createElement("img"));
    img.setAttribute("border", "0");
    img.setAttribute("src", div.getElementsByTagName("day")[0]
        .getAttribute("imageurl"));

    col = row.appendChild(document.createElement("td"));
    col.id = "current_temp";
    col.innerHTML = getText(currentElement, "temp");

    ...

    col = row.appendChild(document.createElement("td"));
    col.className = "desc";
    col.appendChild(document.createTextNode(
        getText(currentElement, "when")));
}
```



Forecasting the weather

```
function buildForecast(div) {  
    ...  
  
    for(var i = 0; i < count; i++) {  
        var day = list.item(i);  
        var row = target.appendChild(document.createElement("tr"));  
        var col3 = row.appendChild(document.createElement("td"));  
        var img = col3.appendChild(document.createElement("img"));  
        img.setAttribute("border", "0");  
        img.setAttribute("src", day.getAttribute("imageurl"));  
  
        var col1 = row.appendChild(document.createElement("td"));  
        col1.width = "100%";  
        col1.align = "left";  
        var innerTable = col1.appendChild(  
            document.createElement("table"));  
        ...  
        var col2 = row.appendChild(document.createElement("td"));  
        col2.align = "right";  
        col2.innerHTML = day.getAttribute("temp");  
    }  
}
```



XUL, XML UI Language

- There is only XUL





XUL Explained

- Full featured UI
- Context menus and accelerator keys
- Tabs and Progress Meter, Split pane
- Menus and toolbars
- RDF, XBL, CSS, XPCOM... oh my!
- <http://www.xulplanet.com/>
- Good news / Bad news, no IE



Find Files

- Layout sample only
- Demonstrates :
 - Context menus and accelerator keys
 - Tool bar and Menu bar
 - Tabs
 - Progress Meter
 - Split pane



Windows, CSS and Javascript

```
<?xml version="1.0"?>
<?xml-stylesheet href="findfile.css" type="text/css"?>

<window
  id="findfile-window"
  title="Find Files"
  persist="screenX screenY width height"
  orient="horizontal"
  onload="initSearchList()"
  xmlns="http://www.mozilla.org/keymaster/
    gatekeeper/there.is.only.xul">

<script src="findfile.js"/>
```



Context menu and accelerators

```
<popupset>
  <popup id="editpopup">
    <menuitem label="Cut" accesskey="t"/>
    <menuitem label="Copy" accesskey="c"/>
    <menuitem label="Paste" accesskey="p" disabled="true"/>
  </popup>
</popupset>

<keyset>
  <key id="cut_cmd" modifiers="accel" key="X"/>
  <key id="copy_cmd" modifiers="accel" key="C"/>
  <key id="paste_cmd" modifiers="accel" key="V"/>
  <key id="close_cmd" keycode="VK_ESCAPE"
    oncommand="window.close();" />
</keyset>
```



Menu and toolbar

```
<toolbox>
<menubar id="findfiles-menubar">
  <menu id="file-menu" label="File" accesskey="f">
    <menupopup id="file-popup">
      <menuitem label="Open Search..." accesskey="o"/>
      <menuitem label="Save Search..." accesskey="s"/>
      <menuseparator/>
      <menuitem label="Close" accesskey="c" key="close_cmd"
        oncommand="window.close();" />
    </menupopup>
  </menu>
  ...
</menubar>

  <toolbar id="findfiles-toolbar">
    <toolbarbutton id="opensearch" label="Open" />
    <toolbarbutton id="savesearch" label="Save" />
  </toolbar>
</toolbox>
```



The tabs, for the love of God, the tabs

```
<tabbox>
  <tabs>
    <tab label="Search" selected="true"/>
    <tab label="Options"/>
  </tabs>

  <tabpanels>
    <tabpanel id="searchpanel" orient="vertical"
      context="editpopup">
...

```



Its a tree, its a table, its a tree?!?!

```
<tree id="results" style="display: none;" flex="1">
  <treecols>
    <treecol id="name" label="Filename" flex="1"/>
    <treecol id="location" label="Location" flex="2"/>
    <treecol id="size" label="Size" flex="1"/>
  </treecols>

  <treechildren>
    <treeitem>
      <treerow>
        <treecell label="mozilla"/>
        <treecell label="/usr/local"/>
        <treecell label="2520 bytes"/>
      </treerow>
    </treeitem>
  </treechildren>
</tree>
```



Advance controls and layout

```
<vbox>
...
<splitter id="splitbar" resizeafter="grow"
          style="display:inline;"/>

<spacer style="height: 10px"/>

<hbox>
  <progressmeter id="progmeter" value="50%"/>
  <spacer flex="1"/>
  <button id="find-button" label="Find" default="true"
          oncommand="doFind();" />
  <button id="cancel-button" label="Cancel"
          oncommand="window.close();" />
</hbox>
</vbox>
```




Editable Tree

- XBL: eXtensible Binding Language
 - XUL is the language for screen layout, CSS is used for styling the features of the screen.
 - XBL is XUL's sister language for changing the behavior of the controls.
 - An XML file that is married to an XUL file with embedded JavaScript for performing control behavior.



The Table

```
<tree id="thetree" flex="1" width="350" height="200"
      enableColumnDrag="true">

  <treecols>
    <treecol id="name" label="Name" flex="1" fixed="false"/>
    <treecol id="occupation" label="Occupation" flex="1"
            fixed="false"/>
    <treecol id="emotion" label="Emotion" flex="1" fixed="false"/>
  </treecols>

  <treechildren>
    <treeitem>
      <treerow>
        <treecell label="Jim Richards"/>
        <treecell label="Mechanic"/>
        <treecell label="Happy"/>
      </treerow>
    </treeitem>
    ...
  </treechildren>
</tree>
```



The CSS ties it together

```
tree
{
  -moz-binding: url("edittree.xml#edittree");
  -moz-user-focus: normal !important;
  -moz-user-select: text;
}

textbox {
  min-height: 1.8em;
  -moz-user-focus: normal !important;
  -moz-user-select: text !important;
}

tree[editing] > treechildren:-moz-tree-row(selected)
{
  background-color: transparent;
  border: none;
}
```



Bindings

```
<bindings id="treeEditBindings"
  xmlns="http://www.mozilla.org/xbl"
  xmlns:xul="http://www.mozilla.org/
    keymaster/gatekeeper/there.is.only.xul">

  <binding id="edittree" extends=
    "chrome://global/content/bindings/tree.xml#tree">
    <content>
      <children includes="treecols"/>
      <xul:stack flex="1">
        <xul:textbox ileattr="text" left="0" top="0" hidden="true"/>
      </xul:stack>
    </content>
    ...
    <handlers>
      <handler event="dblclick">
        var treeBox = this.treeBoxObject;
        var row = {}, col = {}, obj = {};
        treeBox.getCellAt(event.clientX,event.clientY,row,col,obj);
        this.setEditMode(row.value,col.value,true);
      </handler>
    </handlers>
  </binding>
```



Method definition

```
<method name="setEditMode">
  <parameter name="x"/>
  <parameter name="y"/>
  <parameter name="val"/>
  <body>
    var txt = document.
      getAnonymousElementByAttribute(this, "ileattr", "text");
    if (val){
      if (x < 0) return;

      if (this._editRow >= 0) this.
        _assignValueToCell(txt.value,true);
    ...
      txt.addEventListener("keydown", this.fieldKeyDown, false);
      txt.addEventListener("blur", this.fieldChange, true);
    }
    else {
      this.removeAttribute("editing");
    ...
      txt.blur();
    }
  </body>
</method>
```



Tracking Time

- Tying together the pieces
- Cleaning up the joint
 - ✓ Images for toolbars
 - ✓ Getting rid of the menus
- Digging under the tree



Cleaning up the joint

```
<toolbar id="findfiles-toolbar">
  <toolbarbutton id="open"
    image="images/stock_open.png"/>
  <toolbarbutton id="save"
    image="images/stock_save.png"/>
  <toolbarbutton id="print"
    image="images/stock_print.png"/>

  <toolbarseparator flex="5"/>

  <toolbarbutton id="help"
    image="images/stock_help.png"/>
</toolbar>
```



Digging under the tree

- **tree:** This is the outer element of a tree. The content of the tree goes in here.
- **treecols:** This element is a placeholder for treecol elements.
- **treecol:** This is used to declare a column of the tree. By putting these in you can specify additional information about how the data in the columns are sorted and if the user can resize the columns. You should always place an id attribute on a column.
- **treechildren:** This contains the main part of the tree where the individual rows of data go.



Digging under the tree (con't)

- **treeitem:** This contains a single top level row and all its descendants. This element also serves as the item which can be selected by the user. Everything inside here can be selected as a unit. This would go around the entire row so that the entire row is selectable as a whole.t
- **treerow:** A single row in the tree, which should be placed inside a treeitem tag.
- **treecell:** A single cell in a tree. This element would go inside a treerow element.



XUL Revisited

- **XUL**: Laying out the controls
- **RDF**: Providing data for controls
- **XPCOM**: Provides access to external services
- **CSS**: Styling the user interface
- **XBL**: Modifying the behavior for all controls



SVG

Images in Action

“XML Graphics for the Web”



Interacting with images

- Scalable Vector Graphics
 - SVG is a platform for two-dimensional graphics. It has two parts: an XML-based file format and a programming API for graphical applications. Key features include shapes, text and embedded raster graphics, with many different painting styles. It supports scripting through languages such as ECMAScript and has comprehensive support for animation.
 - SVG is a royalty-free vendor-neutral open standard developed under the W3C Process.
- Unique opportunity not available in traditional fat clients
- sXBL (draft)
 - SVG's XML Binding Language (sXBL). sXBL is a mechanism for defining the presentation and interactive behavior of elements described in a namespace other than SVG's. sXBL is intended to be used to enable XML vocabularies (tag sets) to be implemented in terms of SVG elements. For instance, a tag set describing a flowchart could be mapped to low-level SVG path and text elements, possibly including interactivity and animation.



Basic SVG Structure

- **svg:** An SVG document fragment consists of any number of SVG elements contained within an 'svg' element.
- **g:** The 'g' element is a container element for grouping together related graphics elements.
- **path:** Paths represent the outline of a shape which can be filled, stroked, used as a clipping path, or any combination of the three
- **symbol and use:** The 'symbol' element is used to define graphical template objects which can be instantiated by a 'use' element.
- **Basic shape:** Rect, circle, ellipse, line, polyline, polygon
- **Text:** Complete control of font style, size color and effects



The Bar Chart

```
<g stroke="black">
  <!-- "floor" and "wall" -->
  <path fill="lightgray" stroke="darkgray"
        d="M 27,240 l 15,-15 v -224 l -15,15" />
  <path fill="lightgray" stroke="darkgray"
        d="M 41,225 v -224 h 316 v 224" />
  <path fill="darkgray" stroke="none"
        d="M 27,240 l 15,-15 h 316 l -15,15" />

  <!-- axis lines -->
  <path d="M 27,240 h 316"/>
  <path d="M 27,240 v -224"/>

  <!-- value axis major gridlines -->
  <g style="fill:none;">
    <path d="M 27,202 l 15,-15 h 316" />
    <path d="M 27,165 l 15,-15 h 316" />
    <path d="M 27,127 l 15,-15 h 316" />
    <path d="M 27, 90 l 15,-15 h 316" />
    <path d="M 27, 53 l 15,-15 h 316" />
  </g>
</g>
...

```



The Bar Chart (con't)

...

```
<!-- category axis major ticks -->  
<path d="M 27,245 v -5"/>  
<path d="M 106,245 v -5"/>  
<path d="M 185,245 v -5"/>  
<path d="M 264,245 v -5"/>  
  
<!-- value axis minor ticks -->  
<path d="M 22,240 h 5"/>  
<path d="M 22,202 h 5"/>  
<path d="M 22,165 h 5"/>  
<path d="M 22,127 h 5"/>  
<path d="M 22, 90 h 5"/>  
<path d="M 22, 53 h 5"/>  
<path d="M 22, 15 h 5"/>  
</g>
```



Move bar, move!

```
<g id="ShoeBar">
  <polygon style="fill:#5B5B97;" points="86,203 47,203 64,188 101,188"
    id="shoe_bar"
    onmousedown="dragOn(evt)" onmouseup="dragScaleOff()"/>

  <polygon id="shoe_bar_side".../>
  <polygon style="fill:#7575C3;" id="shoe_bar_front" .../>
</g>

function dragOn(evt) {
  actionNode = evt.target;
  actionNode.setAttribute('style', 'fill: red');
  action = DRAG;

  var ar = getPoints(actionNode.getAttribute("points"));
  var currY = parseFloat(ar[0][1]);
  var nowToY = translateY(evt.clientY);

  // handle the current zoom and pan
  ...
  dy = nowToY;
  dragged=false;
}
```




Move bar, move!

```
<g id="barChart" transform="translate(40, 100)" fill-rule="evenodd"
  clip-rule="evenodd" stroke="none" class="legend"
  stroke-width="1" stroke-linecap="round"
  stroke-linejoin="round"
  stroke-miterlimit="10" style="text-anchor:start"
  onmousemove="dragScaleImg(evt)">
```

```
function dragScaleImg(evt) {
  if (actionNode == null) return;

  dragged = true;

  var side = document.getElementById(
    actionNode.getAttribute("id") + "_side");
  var front = document.getElementById(
    actionNode.getAttribute("id") + "_front");
  var sideAr = getPoints(side.getAttribute("points"));
  var frontAr = getPoints(front.getAttribute("points"));

  var nowToY = translateY(evt.clientY);

  // handle the current zoom and pan
  ...
}
```



Move bar, move!

...

```
if (action == DRAG) {
    ar = getPoints(actionNode.getAttribute("points"));
    ar[0][1] = parseFloat(ar[0][1]) + (nowToY - dy);
    ar[1][1] = parseFloat(ar[1][1]) + (nowToY - dy);
    ar[2][1] = parseFloat(ar[2][1]) + ((nowToY - dy));
    ar[3][1] = parseFloat(ar[3][1]) + ((nowToY - dy));

    sideAr[2][1] = parseFloat(sideAr[2][1]) + ((nowToY - dy));
    sideAr[3][1] = parseFloat(sideAr[3][1]) + ((nowToY - dy));

    frontAr[2][1] = parseFloat(frontAr[2][1]) + ((nowToY - dy));
    frontAr[3][1] = parseFloat(frontAr[3][1]) + ((nowToY - dy));

    dy = nowToY;
    actionNode.setAttribute("points", setPoints(ar));
    side.setAttribute("points", setPoints(sideAr));
    front.setAttribute("points", setPoints(frontAr));
}
}
```



Charting Incompetence

- Ties together XMLHttpRequest and SVG
- Using XML and XSL for SVG generation
- Generating interactive SVG
- Alternate implementation:
 - An alternate approach, and one used by Google Maps would be to perform the XSL transformation on the client side using the XSLTProcessor object. This could be used to off-load processing from the server to the client.



Get the SVG

```
<applet width="500" height="300"
  code="com.chariotsolutions.svg.SVGApplet"
  archive="applet/applet.jar,..." id="svg" mayscript="true">

function loadSVG(options) {
  var param = "";
  var svg = document.getElementById("svg");

  for(var i = 0; i < options.length; i++) {
    if(options[i].selected) param += options[i].value + ",";
  }
  param = param.substring(0, param.length - 1);

  svg.setSVGUrl(http://localhost:8000/lps-2.2.1/
    my-apps/programs/jellico-developer-bug-count.jspx
    ?developer_ids=" + param);

  var target = document.getElementById("holder");
  target.style.visibility = "visible";
  target.style.display = "inline";
}
```



The JSPX transformed

```
<c:import url="http://localhost:8000/lps-2.2.1/my-apps/programs/developer-pie.xsl" var="xslt" />

<x:transform xslt="{xslt}">
  <developers>
    <meta>
      <pie>
        <height>200</height>
        <width>200</width>
      </pie>
    </meta>
    <c:forEach var="row" varStatus="i" items="{people.rows}">
      <developer
        red="{(i.index mod 3 == 0) ? 255 - (i.index * 10): 0}"
        green="{(i.index mod 3 == 1) ? 255 - (i.index * 10): 0}"
        blue="{(i.index mod 3 == 2) ? 255 - (i.index * 10): 0}"
        name="{row.FIRST_NAME} {row.LAST_NAME}">{row.C}
      </developer>
    </c:forEach>
  </developers>
</x:transform>
```



XSLT

```
<xsl:template match="developers">
  <svg
    width="{ $width }"
    height="{ $height }">

    <script type="text/ecmascript"><![CDATA[
      function showTotal(evt, count) {
    ...
      }
    </script>

    <g transform="matrix(1 0 0 1 { $width div 2 } { $height div 2 })">
      <text style="font-family:Verdana;font-size:12"
        x="100" y="-85">
        Total Bugs: <xsl:value-of select="sum(//developer)"/>
      </text>
      <xsl:apply-templates select="developer"/>
    </g>
    ...
  </svg>
```



XSLT (con't)

```
<xsl:template match="developer">
  <xsl:variable name="curpos" select="position()"/>
  <xsl:variable name="angleStart" .../>
  <xsl:variable name="angle" select="sum(.) * $x1degree"/>
  <path
    d="M0, 0 L {$xystart} A {$radius}, {$radius} 0 0 1 {$xyend} z"
    style="fill:rgb(@red,@green,@blue); stroke-width:1"
    onmousemove="showTotal(evt, {text()});"
    onmouseout="onOut(evt);"/>

  <rect style="fill:rgb(@red,@green,@blue);
    stroke-width:1" x="110" y="{(position() *14) - 90}"
    width="10"
    height="10"/>

  <text style="font-family:Verdana;font-size:12" x="125"
    y="{(position() *14) - 81}">
    <xsl:value-of select="@name"/>
  </text>
</xsl:template>
```



Open Laszlo

- A Complete UI in a Flash... plug-in
- XML format similar to XUL
- Event driven UI development
- Data binding features
- Object oriented XML



Basic LZX Structure

- **canvas:** The canvas is the container for all views within an application.
- **class:** The `<class>` tag defines a new tag, which be used anywhere in the source file where a built-in tag such as `<view>` or `<layout>` can be used. Defines a new XML tag name, that can be used in the remainder of the application source. An element whose name is this tag name will inherit the attributes and content of this definition.
- **view:** The `<view>` is the most basic viewable element in a Laszlo application. Anything that is displayed on the canvas is a view or extends the view class.
- **Basic controls:** Inputs, buttons, menus, grids, trees, scrollbars, etc.
- **datapaths and datapointers:** Datapaths are used to merge data hierarchies with the hierarchy of a Laszlo application. This is done using the the XPath syntax



The Jellico LZX Application

```
<canvas height="600" width="800" oninit="">
  <include href="jellico-people.lzx" />
  <include href="jellico-main.lzx" />
  <window id="people" x="530" y="34" width="260" height="550">
  ...
    <jellico-people id="powerToThePeople">
      <attribute name="current"/>
      <method name="addBug" args="dPath">
        var currentPerson = people.getAttribute("first_name") + " "
          + people.getAttribute("last_name");
        mdText.setText("Bug # " + dPath.getNodeAttribute("id") +
          " has been reassigned to " + currentPerson);
        dPath.setNodeAttribute("developer", currentPerson);
        md.open();
      </method>
    </jellico-people>
  </window>
  <window id="jellico" x="7" y="34" width="525" height="550"
    title="Jellico" resizable="true" closeable="true"
    visible="false">
    <jellico-main/>
  </window>
</canvas>
```



The Bug Window

```
<library>
  <include href="lz/tree.lzx"/>
  <dataset src="jellico-projects.jspx" request="true"
    type="http" name="projectData" autorequest="true"/>
  <dataset src="jellico-bugs.jspx" type="http" name="bugData"/>

  <class name="jellico-main" bgcolor="#eaeaea" height="100%">
    <tabs x="5" width="100%" height="100%">
      <tabpane text="Bug Tracking" inset_left="10" inset_top="10"
        inset_bottom="10" inset_right="10">
        <form layout="axis: y; spacing: 15" id="bugForm">
          <method name="queryBugs">
            var item = project_id.getSelection();
            var proj_id = item.getValue();
            //Debug.write("project id=" + proj_id);
            bugData.setQueryString({project_id: proj_id});
            bugData.doRequest();
          </method>
          <datapointer xpath="bugData:/bugs">
          </datapointer>
        </form>
      </tabpane>
    </tabs>
  </class>
  ...
```



The Contact Window

```
<library>
  <font src="bitstream-vera-1.10/verabd.ttf" name="boldfont"/>

  <dataset src="jellico-people.jspx" request="true"
    type="http" name="mydata" autorequest="true"/>

  <class name="row" height="30" bgcolor="0xE7E7D6" width="100%">
  ...
  </class>

  <class name="jellico-people" bgcolor="#eaeaea" height="100%">

    <view name="contactsborder" bgcolor="0x676767"
      width="100%" height="100%">
      <text name="test" x="5" y="1" fgcolor="white">contacts</text>
      <view datapath="mydata:/resultset"
        x="2" y="20" width="100%" height="100%" clip="true">
        <view width="{parent.width-16}">
          <row datapath="result"/>
          <simplelayout axis="y"/>
        </view>
      ...
    </library>
```



Sample JSPX

```
<jsp:root
  xmlns:jsp="http://java.sun.com/JSP/Page"
  xmlns:c="http://java.sun.com/jsp/jstl/core"
  xmlns:sql="http://java.sun.com/jsp/jstl/sql"
  version="2.0">
  <sql:setDataSource
    ...
  />

  <sql:query var="people" dataSource="{example}">
    SELECT * FROM person
  </sql:query>

  <resultset>
    <c:forEach var="row" items="{people.rows}">
      <result id="{row.ID}" img="art/{row.ID}.gif"
        firstname="{row.FIRST_NAME}" lastname="{row.LAST_NAME}"
        displayname="{row.LAST_NAME}, {row.FIRST_NAME}"
        nickname="{row.CVS_USERNAME}" email="{row.EMAIL}"
        work_webpage="..." />
    </c:forEach>
  </resultset>
</jsp:root>
```



Questions?

Thank you!