Canon

New Directions for Web Applications

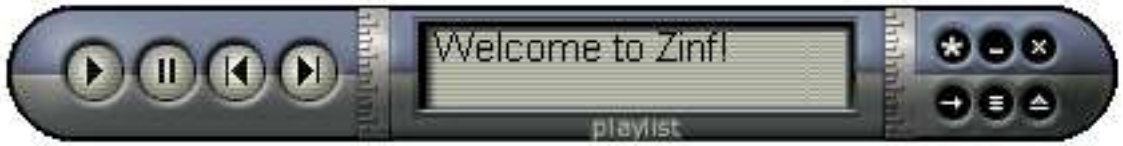
Dave Raggett, Canon, TV Raman, IBM
Web Applications Workshop, San Jose, June 2004

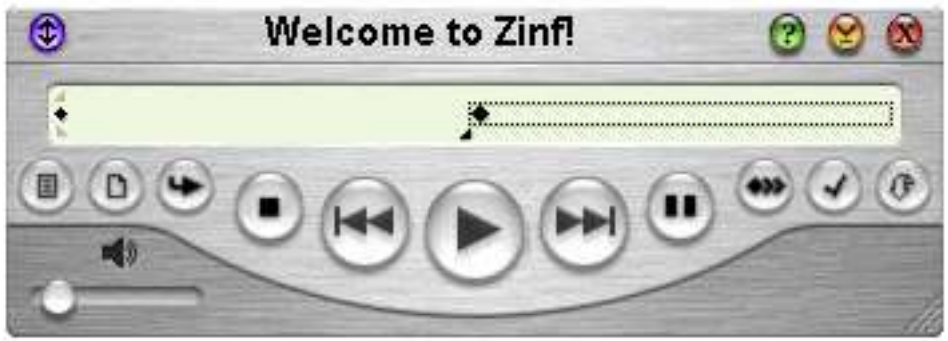
Goals

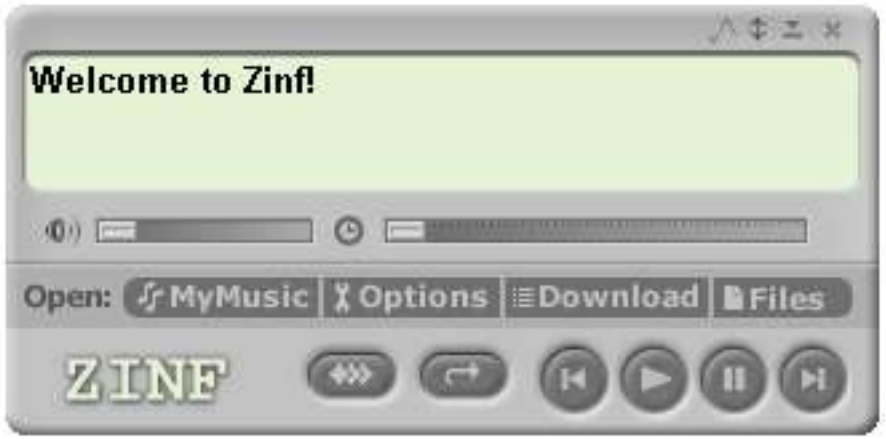
- Break Web applications out of the browser!
- Freedom to build wider variety of applications
- Reduced costs and increased flexibility
- Easy adaptation to wide range of devices
- Multimodal User Interface
- Author defined controls
- Ability to mix novel and standard markup

Theming

- Zinf – one application, three themes

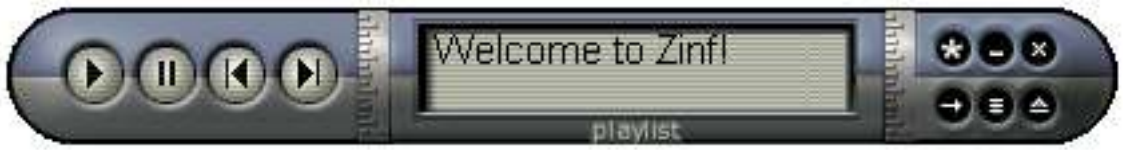






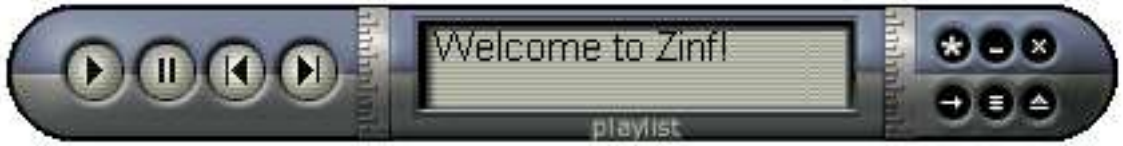
Theses can also be applied across applications, vis Gnome desktop themes

Break free out of the Browser



- Non-rectangular windows
- No window chrome
- Treat like regular applications
  - Launched from application menus or desktop icons
  - Can be bound to media types, e.g. audio/mpeg

Application Model



- Describe object model in XML
  - Properties
    - Preferences
    - Play-list
    - Current track
      - Duration, current position, name, artist, ...
  - Methods
    - volume, pause, rewind, fast forward, previous/next track, quit
  - Events
    - Playback error, end of track, end of play-list, ...

Themes and Intentions

- Split user interface into
  - Abstract UI controls and layout intentions
  - Theme = presentation and behavior for controls
- XForms range control as scalar value
  - Themed as rotary dial or thermometer
  - Role of SVG and XBL for defining such controls
- Layout intentions
  - Vertical, horizontal, grid, ...
  - Delegate size/position decisions to layout manager
  - Detailed appearance determined by current theme

Canon

Declarative Treatment of Behavior

- Simple event binding
  - dom:activate on button invokes application *pause* method
  - xf:xforms-value-changed on dial sets app *volume*
- State transition models
  - Event driven transitions between named states
  - Represented in XML and XPath
  - Transitions invoke methods, update data, raise events
  - Nested states and concurrent execution
  - Run on device or server, or on both
    - Distributed execution model

Multimodal Interaction

- Treat all input uniformly as events
  - Audio, speech, keystrokes and stylus
- Use of grammars for recognition and extracting application semantics
- EMMA – extensible multi-modal annotations
  - XML language for interpreted input
    - Application specific + standard annotations
    - Role of XPath for dealing with EMMA
- SMIL timing model

Mixing Novel and Standard Markup

- Treat XML as instructions for creating a composition of objects
- Bind elements to objects
  - Implicit binding for standards-based markup
- Interoperability depends on
  - Binding mechanism, and object language
    - e.g. XBL and ECMAScript
  - Libraries and interfaces these objects depend on

Canon

Who's doing what?

- VB and MMI WGs
  - EMMA and grammars for interpreted input
  - System & Environment for dynamic adaptation
  - State transition models for modeling behaviors
  - Binding novel markup to code
- DI WG
  - CC/PP, core presentation attributes, document profiles
- SVG WG
  - SVG and XBL for themes and novel UI controls
- XForms WG
  - Application data models and abstract UI controls
- XHTML WG
  - XHTML2 and XFrames

What's missing?

- Using XML for application object models
  - Relation to XForms and XBL?
- Layout intentions
  - Representation
    - CSS: http://www.w3.org/TR/NOTE-layout from 1996
    - XML: XSL-FO, Glade, XUL and XAML
  - Interoperable means for dealing with UI resources
    - e.g. creating/removing new windows, audio mixers, ...
- What else?

XAML example

```
<Canvas xmlns="http://schemas.microsoft.com/2003/xaml">
   <Button Canvas.Left="10"
           Canvas.Top="10"
           Width="90px"
           Height="32px">Click Me</Button>
</Canvas>
```

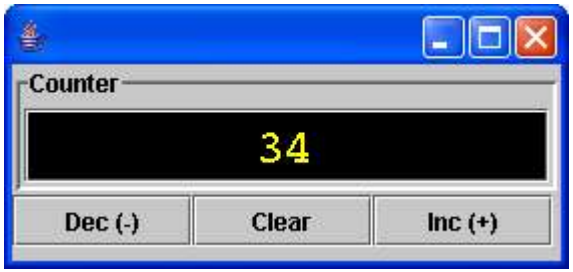
In procedural C# code this can be done with:

```
Button btn = new Button();
btn.Width = new Length(90);
btn.Height = new Length(32);
Canvas.SetTop(btn, new Length(10));
Canvas.SetLeft(btn, new Length(10));
btn.Content = "Click Me";
```

XAML

- Layout controls
  - Canvas, DockPanel, FlowPanel, GridPanel, Table, TextPanel, Text
- UI controls
  - Button, CheckBox, RadioButton, TextBox, HyperLink, HorizontalScrollBar, ListBox, ComboBox, ...
  - Embedded presentation
- Declarative animation support
- Events bound to named handlers
- Behavior defined in code e.g. C# or script

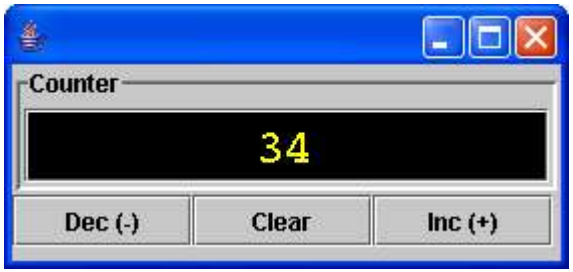
XUL – Mozilla UI language




```
<xul>
  <vbox id="COUNTER">
   <groupbox>
     <caption label="Counter"/>
     <textbox id="DISPLAY"
       style="align: center; color: yellow; background: black;
              font: 24 bold monospace;" />
   </groupbox>
   <hbox>
     <button label="Dec (-)" command="dec" style="width: 90px" />
     <button label="Clear" command="clear" style="width: 90px" />
     <button label="Inc (+)" command="inc" style="width: 90px" />
   </hbox>
  </vbox>
</xul>
```

See XUL coding challenge: http://xul.sourceforge.net/counter.html

Hypothetical Web Apps language




```
<webapp>
  <data>
    <counter>34</counter>
  </data>
  <vbox>
   <groupbox caption="Counter">
     <textbox ref="/counter"/>
   </groupbox>
   <hbox>
     <button command="dec(/counter)">Dec (-)</button>
     <button command="clear(/counter)">Clear</button>
     <button command="inc(/counter)">Inc (+)</button>
   </hbox>
  </vbox>
</webapp>
```

With separation of data and use of XPath and theme engine for styling

CSS for Policy based layout

- Proposal in June 1996
  - Bert Bos, Dave Raggett and Håkon Lie
    - http://www.w3.org/pub/WWW/TR/NOTE-layout

```
<style type="text/css">
@page {layout: fixed; width: 500px; height: 500px}
@frame header {left: 0px; top: 0px; width: 500px; height: 100px}
@frame toc {left: 0px; top: 100px; width: 500px; height: 400px}
h1 {flow: header}
ul {flow: toc}
</style>

<h1>Welcome to my Home Page</h1>

<ul>
<li>Favorite Places
<li>Picture of my family
<li>My plans for this week
</ul>
```

XBL

- Binding XML elements to properties, methods and event handlers
  - Written in XML with embedded JavaScript
  - Developed by Mozilla team
  - Now being applied to SVG
    - SVG-XBL task force
- XBL – behavior sheets
  - Adding behaviors to novel elements
  - Use for defining novel controls (SVG components)

Canon

XBL example

```
<?xml version="1.0"?>
<?xml-stylesheet href="notes.css"?>
<bindings xmlns="http://www.mozilla.org/xbl"
  xmlns:html="http://www.w3.org/1999/xhtml">
  <binding id="board" styleexplicitcontent="true">
    <implementation>
      <property name="dragging"> null </property>
      <property name="currX"> 0 </property>
      <property name="currY"> 0 </property>
    </implementation>
    <handlers>
      <handler event="mousedown">
        if (event.originalTarget.parentNode.className == 'caption') {
          this.dragging = event.originalTarget.parentNode.parentNode;
          this.currX = event.clientX;
          this.currY = event.clientY;
        }
    </handler>
    <handler event="mouseup">
      this.dragging = null;
    </handler>
    <handler event="mousemove">
        this.currY = event.clientY;
        ...
    </handler>
  </handlers>
</binding>
```