



Emerging Internet Technologies

Real World Accessibility with ARIA

Becky Gibson
Web Accessibility Architect

Agenda

- ARIA Overview
- Dojo Overview
- Implementing ARIA in Dojo
 - Keyboard Support
 - ARIA Roles & States
- Demo
- Summary

ARIA - What is it?

- Accessible Rich Internet Applications
- W3C Specification, like HTML, CSS, XML etc.
- Within Protocols & Formats Working Group which is part of WAI - Web Accessibility Initiative
- It is very close to becoming a W3C Recommendation
- Implemented in Firefox, IE8 with Opera and Safari under development
- Gaining increasing support by browsers, Web toolkits and assistive technologies

ARIA Overview

- Add role semantics to scripted user interface (UI) elements
- Update state information dynamically
- Make items focusable via tabindex attribute
- Add keyboard event handling
 - Mimic the keyboard behavior of the rich client UI
 - Minimize tab key navigation
- Add live region information and notification to support Ajax

ARIA Example - Tree



Role = tree
(on outer container)

Role = treeitem
expanded=true
(on open Africa node)

Role = treeitem
selected=true
(on highlighted Egypt child node with no children)

Role = treeitem
expanded=false
(on closed Australia node)

Dojo - What is it?

- Open Source JavaScript Toolkit
- “Easy” Ajax
- Data Binding
- Full event system
- Browser abstraction layer
- User Interface Widgets
- Dual Licensed
 - Academic Free License v2.1
 - BSD License
- Dojo 1.0 available in November, 2007
 - 1.1 in March, 2008
 - 1.1.1 in May, 2008
- Dojo 1.2 released October, 2008



Core Widget Set - dijit

- Accessible
- Internationalized
- Customizable Look and Feel
- Developer Documentation
 - API
 - User Manual
- Supported Browsers
 - Firefox 2, Firefox 3
 - IE 6, IE 7, IE 8 support in progress
 - Safari 3
- Data Binding
 - Tree, Grid*, Select

*Grid is in dojox (dojo extensions)

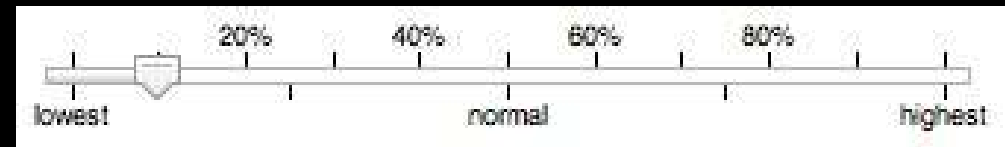
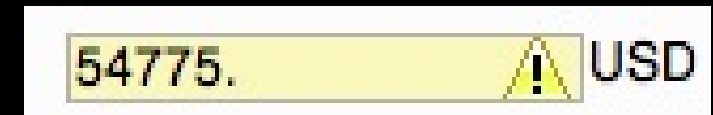
Dojo Core Widgets

- Form Widgets
- Layout Widgets
- Advanced Widgets

Form Widgets



- Button, Dropdown Button, Combo Button
- Checkbox, Radio
- ComboBox, Filtering Select, Multi Select
- Textbox
- Currency & Integer Validation Textboxes
- Resizable Textarea
- Slider
- Integer Spinner
- Inline Editbox
- Dropdown Calendar



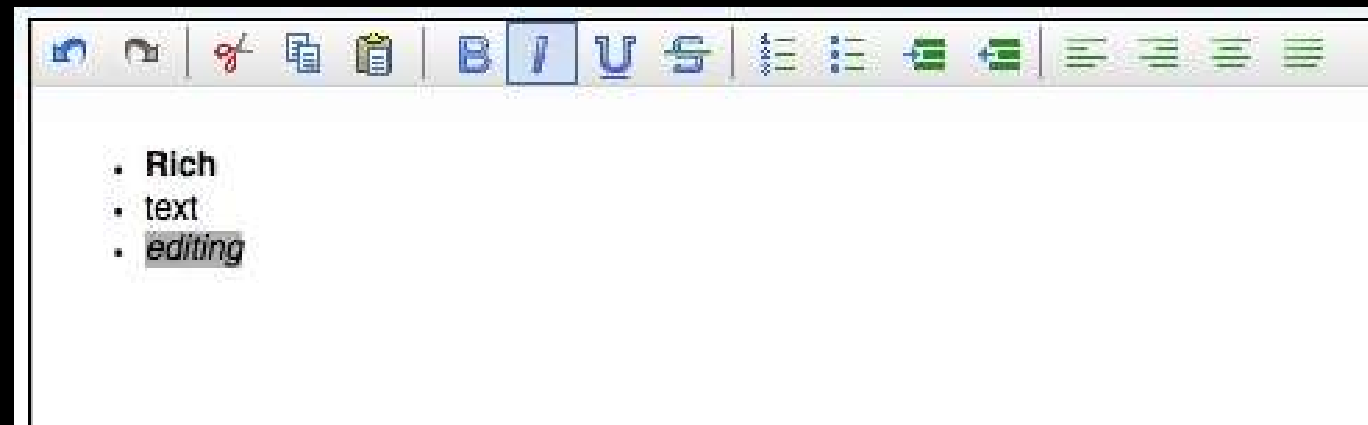
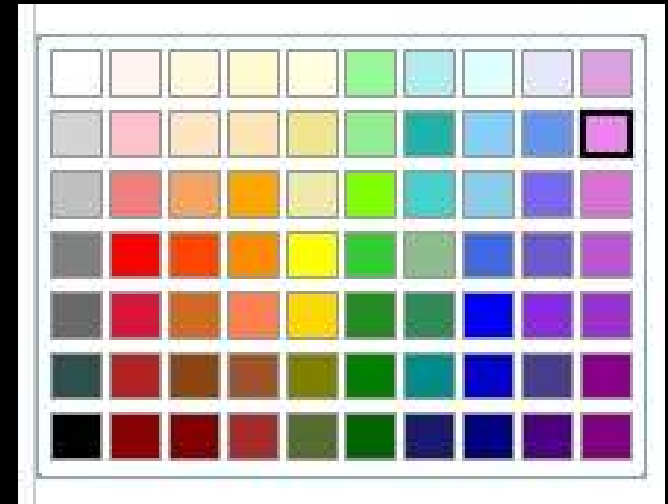
Layout Widgets

- Accordion Container
- Content Pane
- Dialog
- Border Container
- Tab Container
- Title Pane



Advanced Widgets

- Color Palette
- Context Menu
- Rich Text Editor
- Progress Bar
- Toolbar
- Tooltip, Tooltip Dialog
- Tree



Dojo Core Widget Accessibility

- ARIA Specification Implemented
 - Fully keyboard accessible in Firefox and Internet Explorer
 - Screen reader accessible in Firefox
- Support Low Vision
 - Work in Operating System High Contrast mode
 - No fixed font sizes
 - Work with images off

Implementing Accessibility in Dojo

- Educating the Dojo Community on Accessibility
 - Why accessibility is important
 - Low vision issues
 - Need for keyboard support
 - Basics of screen reader and assistive technology use
- Making it easy to incorporate ARIA
- Maintaining Performance
- Providing a11y documentation
- Testing, flagging and fixing issues

ARIA Keyboard Support in Dojo

- Extended tabindex (IE, FF, Opera 9.5, HTML5)
- Keyboard event normalization
 - onkeypress onkeydown as appropriate
 - ondiijitclick event handles onclick, enter or space key press
- Functions for finding elements in the tab order
 - `dijit._getTabNavigable(node)`
 - `dijit.getFirstInTabbingOrder(node)`
 - `dijit.getLastInTabbingOrder(node)`
- Setting Focus
 - Focus manager
 - `focusNode` attach point

Setting ARIA Roles and States

- Setting roles and states in dijit templates
 - waiRole="treeitem"
 - waiState="expanded-true"
- Dynamically set and update roles and states via dijit apis
 - dijit.setWaiRole/State()
 - dijit.getWaiRole/State()
 - dijit.hasWaiRole/State()
 - dijit.removeWaiRole/State()

Example - Creating a Dijit Dialog

■ From Markup

```
<div dojoType="dijit.Dialog" id="ex1" title="Confirm Purchase">  
  <div>dialog contents</div>  
</div>
```

■ Via Scripting

```
<script type="text/javascript">  
  var pane = dojo.byId('myDialogContents');  
  pane.style.width = "300px";  
  newDlg = new dijit.Dialog({  
    id: "dialog",  
    title: "Programmatic Dialog"  
  }, pane);  
  newDlg.show();  
</script>  
  
<div id="myDialogContents" style="display:none">dialog  
  content</div>
```

Setting Roles & States in Templates

```
<div class="dijitDialog" tabindex="-1" waiRole="dialog"
  waiState="labelledby-#{id}_title">
  <div dojoAttachPoint="titleBar">
    <span id="#{id}_title">#{title}</span>
    <span class="dijitDialogCloseIcon"
      dojoAttachEvent="onclick: onCancel">
      <span dojoAttachPoint="closeText"
        class="closeText">x</span>
      </span>
    </div>
    <div dojoAttachPoint="containerNode"
      class="dijitDialogPaneContent"></div>
  </div>
```

Setting Role and States via APIs

From `dijit.ProgressBar.update()`:

```
dijit.setWaiState(this.internalProgress,  
    "valuenow", this.progress);  
dijit.setWaiState(this.internalProgress,  
    "valuemin", 0);  
dijit.setWaiState(this.internalProgress,  
    "valuemax", this.maximum);
```

High Contrast Detection & Support

- Detect high contrast mode
- Detect images off mode
- Add dijit_a11y class to body to trigger a11y styles
- provides text alternatives for CSS Background images
- No hard coded font sizes

High Contrast with CSS

```
<div class="dijitDialog" tabindex="-1"
  waiRole="dialog" waiState="labelledby-#{id}_title">
  <div dojoAttachPoint="titleBar">
    <span id="#{id}_title">#{title}</span>
    <span class="dijitDialogCloseIcon"
      dojoAttachEvent="onclick: onCancel">
      <span dojoAttachPoint="closeText"
        class="closeText">x</span>
      </span>
    </div>
    <div dojoAttachPoint="containerNode"
      class="dijitDialogPaneContent"></div>
  </div>
```

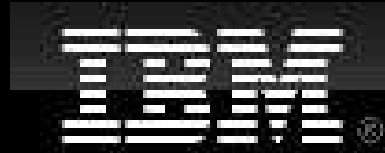
High Contrast with CSS

```
.tundra .dijitDialogCloseIcon {  
  background : url("images/tabClose.png");  
}  
  
.dijitDialog .closeText {  
  display:none;  
}  
  
.dijit_a11y .dijitDialog .closeText {  
  display:inline;  
}
```

Dojo Demo

(pre-recorded demos at <http://www.weba11y.com/demos>)

ARIA Support (current or in-progress)



Summary

- ARIA can make Web 2.0 applications accessible
- Dojo makes developing Web 2.0 apps easier
- Dojo + ARIA enables **easy** & **accessible** Web 2.0
- Wide support for ARIA
- Other toolkits are adopting ARIA
- ARIA - Use it!

Resources

- **Becky's Accessibility Presentations and Papers**
 - <http://www.weba11y.com/Presentations/presentations.html>
- **ARIA Roadmap, Best Practices, Primer, Specifications**
 - <http://www.w3.org/wai/pf>
- **Mozilla Developer Center - Firefox ARIA Information**
 - http://developer.mozilla.org/en/docs/Accessible_DHTML
- **Mailing list and communities for ARIA issues**
 - <http://lists.w3.org/Archives/Public/wai-xtech/>
 - <http://groups.google.com/group/free-aria>
- **Dojo**
 - Home - <http://www.dojotoolkit.org/>
 - Examples & Tutorials - <http://dojocampus.org/>
 - Book - <http://docs.dojocampus.org/>
- **DHTML Style Guide Draft**
 - http://dev.aol.com/dhtml_style_guide