Screen Readers, ARIA & HTML5

Jason Kiss
accessibleculture.org
@jkiss

Accessibility Summit 2011
Sept. 27, 2011
Today's Agenda

• Basic HTML5 elements
• ARIA landmark roles
• Browsers, accessibility APIs, and screen readers
• Expected screen reader behaviour
• Actual screen reader behaviour (with videos)
  – JAWS, NVDA, Window-Eyes
  – VoiceOver
• HTML5/ARIA accessibility: a shared responsibility
Basic HTML5 section elements

- article
- aside
- footer
- header
- nav
- section
WAI-ARIA

- a bunch of attributes: roles, states, properties
- dynamic, interactive content
- document structure
- part of the HTML5 specification
Landmark Roles

“regions of the page intended as navigational landmarks”

- application
- banner
- complementary
- contentinfo
- form
- main
- navigation
- search

Example

<div id="main" role="main">
Document Structure Roles

“structures that organize content in a page”

- article
- columnheader
- definition
- directory
- document
- group
- heading
- img
- list

- listitem
- math
- note
- presentation
- region
- row
- rowheader
- separator
- toolbar
<table>
<thead>
<tr>
<th>HTML5 Element</th>
<th>Implied ARIA Role</th>
<th>Permitted ARIA Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>article</td>
<td>article*</td>
<td>article, document, application, or main</td>
</tr>
<tr>
<td>aside</td>
<td>complementary</td>
<td>note, complementary, or search</td>
</tr>
<tr>
<td>footer</td>
<td>none</td>
<td>contentinfo (one per document)</td>
</tr>
<tr>
<td>header</td>
<td>none</td>
<td>banner (one per document)</td>
</tr>
<tr>
<td>nav</td>
<td>navigation</td>
<td>navigation</td>
</tr>
<tr>
<td>section</td>
<td>region*</td>
<td>alert, alertdialog, application, contentinfo, dialog, document, log, main, marquee, region, search, or status</td>
</tr>
</tbody>
</table>

* Not a landmark role
<header>

<nav>
  • Link #1
  • Link #2
</nav>

<div id="main">
  <section>
    <article>
      <header>
      </header>
      <footer>
      </footer>
      <div id="application">
        button
      </div>
      <div id="form">
        button
      </div>
    </article>
  </section>
</div>

<aside>
  <form>
    Search
  </form>
</aside>

<footer>
Screen Readers, HTML5 & ARIA

<header role="banner">

<nav role="navigation">
  • Link #1
  • Link #2
</nav>

<div id="main" role="main">
 <section role="region">
  <article role="article">
    <header>
    </header>
    <footer>
    </footer>
  </article>
 <div id="application" role="application">
  button
 </div>
 <div id="form" role="form">
  button
 </div>
</section>

<aside role="complementary">
 <form role="search">
  Search
 </form>
</aside>

<footer role="contentinfo">
</footer>
### HTML5 Only Test Case

#### Header

#### Nav
- HTML5 Only
- [HTML5 plus ARIA](#)

#### Content

#### Section
- Article Header
- article content
- Article Footer

#### Application
- Application Button

#### Form
- Form Button

#### Aside
- Search

#### Footer
HTML5 plus ARIA Roles Test Case

Header / Banner

Nav / Navigation
- HTML5 Only
- HTML5 plus ARIA

Content / Main

Section / Region
- Article Header
- article content
- Article Footer

Application

Form

Aside / Complementary
Search

Footer / ContentInfo
How Screen Readers Get Information

Accessibility APIs

• Interface between the browser and screen reader

• Browsers relay roles, states, properties, and events to the API

• Screen readers access the API and are notified of changes in elements, states, properties
Accessibility APIs

Internet Explorer 6 and 7
• Microsoft Active Accessibility (MSAA)

Internet Explorer 8 and 9
• Microsoft Active Accessibility (MSAA)
• User Interface Automation (UIA) [Windows 7]
Accessibility APIs

Firefox in Windows
• Microsoft Active Accessibility (MSAA)
• IAccessible2 (IA2)

Safari and the Mac Accessibility API
• Mac Accessibility API
Accessibility APIs

Extreme Variability in:

• What type of information each API supports
• The information the browser sends to the API
• The information the screen reader takes from the API
Document Object Model (DOM)

• A tree of nodes: elements and their attributes
• Screen readers can access the DOM directly
• Is useful when:
  – the API does not support certain information, or
  – the information sent by the browser to the API is incomplete
http://www.paciellogroup.com/blog/

- Great source of info on:
  - ARIA support in browsers & screen readers
  - Accessibility API support
Screen Reader & Landmark Interaction

Navigation by landmark
• Special keyboard commands

Reading through the page
• Announce landmarks as they come
• Note the entering and leaving of landmarks

ARIA Landmarks List Dialog
• Special dialog window listing all landmarks
Expected Screen Reader Behaviour

HTML5 Only

• `<aside>` → “complementary landmark”
• `<footer>` → “contentinfo landmark” once per document
• `<header>` → “banner landmark” once per document
• `<nav>` → “navigation landmark”
Expected Screen Reader Behaviour

HTML5 plus ARIA Roles

- All explicitly identified landmark roles are announced
- Document structure roles, e.g., article and region, are not announced
### JAWS 12/13

**HTML5 Only**

<table>
<thead>
<tr>
<th>HTML5 Element</th>
<th>JAWS 12/13</th>
</tr>
</thead>
<tbody>
<tr>
<td>article*</td>
<td>article (IE6+ and FF)</td>
</tr>
<tr>
<td>aside</td>
<td>No</td>
</tr>
<tr>
<td>footer</td>
<td>contentinfo (FF Only)</td>
</tr>
<tr>
<td>header</td>
<td>banner (FF Only)</td>
</tr>
<tr>
<td>nav</td>
<td>navigation (IE6+ &amp; FF)</td>
</tr>
<tr>
<td>section*</td>
<td>No</td>
</tr>
</tbody>
</table>

### Issues

- **article** is not a landmark
- In FF, every **header** is a banner landmark
- In FF, every **footer** is a contentinfo landmark
- **Heading structure is incorrect**
JAWS 12/13

HTML5 Only

• JAWS reads from the DOM to identify `article` and `nav` elements as landmarks

• This is why `article` and `nav` elements are considered landmarks even in IE6

• In FF, JAWS also uses the API: all `header` and `footer` elements are called `banner` and `contentinfo` landmarks
Screen Readers, HTML5 & ARIA

Header

Nav <h2> JAWS: <h3>
Link #1
Link #2

Content <h1>

Section <h2> JAWS: <h3>
<article>
  Article Header <h3> JAWS: <h5>
  Article Footer <h3> JAWS: <h5>
</article>

Application <h2> button
Form <h2> button

Aside <h2>
<form>
  Search
</form>

Footer <h2>
## JAWS 12/13

### HTML5 plus ARIA Roles

<table>
<thead>
<tr>
<th>ARIA Role</th>
<th>JAWS 12/13</th>
</tr>
</thead>
<tbody>
<tr>
<td>application</td>
<td>Yes</td>
</tr>
<tr>
<td>article*</td>
<td>Yes</td>
</tr>
<tr>
<td>banner</td>
<td>Yes</td>
</tr>
<tr>
<td>complementary</td>
<td>Yes</td>
</tr>
<tr>
<td>contentinfo</td>
<td>Yes</td>
</tr>
<tr>
<td>form</td>
<td>Yes</td>
</tr>
<tr>
<td>main</td>
<td>Yes</td>
</tr>
<tr>
<td>navigation</td>
<td>Yes</td>
</tr>
<tr>
<td>region*</td>
<td>Yes</td>
</tr>
<tr>
<td>search</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Issues
- **article and region** are not landmarks
- Heading structure is incorrect
VIDEO

JAWS 12.0.1170

Firefox 6

HTML5 and ARIA roles

Navigation by landmark

1 min 4 seconds
New Behaviour in JAWS 13

• Each landmark is called a “landmark region”, e.g., “navigation landmark region”

• JAWS announces the start and end of landmarks, e.g.,
  – “banner landmark region start”
  – “banner landmark region end”
VIDEO

JAWS 13.0.97
Internet Explorer 9
HTML5 and ARIA roles
Reading through the page
46 seconds
JAWS 10/11

HTML5 `<header>` bug in Firefox

• Content inside the `header` element is not accessible!

• Fixed in JAWS 12
### Issues

- In FF, every `header` is a banner landmark
- In FF, every `footer` is a `contentinfo` landmark

### Table: NVDA 2011.2 HTML5 Only

<table>
<thead>
<tr>
<th>HTML5 Element</th>
<th>NVDA 2011.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>article*</td>
<td>No</td>
</tr>
<tr>
<td>aside</td>
<td>No</td>
</tr>
<tr>
<td>footer</td>
<td><code>contentinfo</code> (FF Only)</td>
</tr>
<tr>
<td>header</td>
<td><code>banner</code> (FF Only)</td>
</tr>
<tr>
<td>nav</td>
<td><code>navigation</code> (FF Only)</td>
</tr>
<tr>
<td>section*</td>
<td>No</td>
</tr>
</tbody>
</table>
NVDA 2011.2

HTML5 Only

• NVDA is not using the DOM, but relying on the APIs

• This is why in IE, no HTML5 elements on their own are considered landmarks

• This is why in FF, nav, header, and footer are considered landmarks
VIDEO
NVDA 2011.2
Firefox 6
HTML5 Only
No ARIA roles
Navigation by landmark
33 seconds
Screen Readers, HTML5 & ARIA

NVDA 2011.2

HTML5 plus ARIA Roles

<table>
<thead>
<tr>
<th>ARIA Role</th>
<th>NVDA 2011.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>application</td>
<td>No</td>
</tr>
<tr>
<td>article*</td>
<td>No</td>
</tr>
<tr>
<td>banner</td>
<td>Yes</td>
</tr>
<tr>
<td>complementary</td>
<td>Yes</td>
</tr>
<tr>
<td>contentinfo</td>
<td>Yes</td>
</tr>
<tr>
<td>form</td>
<td>No</td>
</tr>
<tr>
<td>main</td>
<td>Yes</td>
</tr>
<tr>
<td>navigation</td>
<td>Yes</td>
</tr>
<tr>
<td>region*</td>
<td>No</td>
</tr>
<tr>
<td>search</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Issues

- Doesn't identify application or form landmarks
- In FF, header and footer elements without landmarks are still called banner and contentinfo landmarks
NVDA 2011.2

HTML5 plus ARIA Roles

• In FF, NVDA is reading landmark role information from the IA2 API
• In IE, NVDA gets it straight from the DOM, since IE passes no landmark information to the API
• This is why in IE6 and above, NVDA announces the same landmark roles as it does in FF
Screen Readers, HTML5 & ARIA

VIDEO
NVDA 2011.2
Firefox 6
HTML5 and ARIA roles
Navigation by landmark
51 seconds
VoiceOver / Mac OS X Lion

HTML5 Only

<table>
<thead>
<tr>
<th>HTML5 Element</th>
<th>VoiceOver</th>
</tr>
</thead>
<tbody>
<tr>
<td>article*</td>
<td>No</td>
</tr>
<tr>
<td>aside</td>
<td>No</td>
</tr>
<tr>
<td>footer</td>
<td>No</td>
</tr>
<tr>
<td>header</td>
<td>No</td>
</tr>
<tr>
<td>nav</td>
<td>No</td>
</tr>
<tr>
<td>section*</td>
<td>No</td>
</tr>
</tbody>
</table>

Issues

- Doesn't announce any implied landmarks
VoiceOver / Mac OS X Lion

HTML5 plus ARIA Roles

<table>
<thead>
<tr>
<th>ARIA Role</th>
<th>VoiceOver</th>
</tr>
</thead>
<tbody>
<tr>
<td>application</td>
<td>Yes</td>
</tr>
<tr>
<td>article*</td>
<td>No</td>
</tr>
<tr>
<td>banner</td>
<td>Yes</td>
</tr>
<tr>
<td>complementary</td>
<td>Yes</td>
</tr>
<tr>
<td>contentinfo</td>
<td>Yes</td>
</tr>
<tr>
<td>form</td>
<td>No</td>
</tr>
<tr>
<td>main</td>
<td>Yes</td>
</tr>
<tr>
<td>navigation</td>
<td>Yes</td>
</tr>
<tr>
<td>region*</td>
<td>No</td>
</tr>
<tr>
<td>search</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Issues**
- Doesn't identify form landmarks
- Doesn't announce the landmark type when navigating by landmark

**Features**
- Announces entering and leaving landmark
VIDEO
VoiceOver / Mac OS X Lion
Safari 5
HTML5 and ARIA roles
1 min 53 seconds
Window-Eyes 7.5.0 and Earlier

- Just don't “do” HTML5 or ARIA landmarks

HTML5 and Internet Explorer

- Lists and headings get corrupted when used with basic HTML5 elements
- Window-Eyes finds frames where none exist
- Page structure is misrepresented
Window-Eyes

Fortunately, these bugs...

• are not a problem in Firefox
• are fixed in Window-Eyes 7.5.1
VIDEO
Window-Eyes 7.5.0
Internet Explorer 9
HTML5 and ARIA roles
1 min 43 seconds
Accessibility: A Shared Responsibility

- Browser vendors
- Screen reader vendors
- Developers
- Users
Accessibility: A Shared Responsibility

Browser vendors

• Implement HTML5 and ARIA according to their specifications
• Fully utilise the accessibility APIs to pass information to assistive technology
Accessibility: A Shared Responsibility

Screen Reader vendors

• Implement HTML5 and ARIA according to their specifications
• Fully utilise the accessibility APIs to present page structure, content, and events to the user
Accessibility: A Shared Responsibility

Developers

• Implement HTML5 and ARIA according to their specifications
• Offer reasonable workarounds as necessary to improve the experience for screen readers
• Know the audience and the technology users have at their disposal
• Identify and report bugs
Accessibility: A Shared Responsibility

Users

• Use screen readers that support HTML5 and ARIA
• Keep the screen reader up-to-date
• Not settle for less: Demand improved support from vendors
• Identify and report bugs
A More Accessible Web

Use HTML5 and ARIA, but...

• Understand how it can impact users

• Make an informed decision that balances your development needs with the access needs of your potential audience
Thanks!

Jason Kiss
accessibleculture.org
@jkiss

Accessibility Summit 2011
Sept. 27, 2011