

Accessibility with WAI-ARIA

Maria Schiewe, Wikimedia Deutschland,
maria.schiewe@wikimedia.de

Background

- ▶ committee member of Wikimedia Deutschland since May 2009
- ▶ established and head department for usability and IT together with colleague Tim Bartel
- ▶ profession: accessibility consultant for public authority, not a (good) developer
- ▶ focus on blind and visually impaired

Goals

- ▶ accessible Wikimedia projects, not only for consumers but contributors
- ▶ raise awareness for accessibility (core and extension developers, template programmers, authors)
- ▶ cooperate with disability rights organisations to identify existing issues
- ▶ involve disabled users into testing (accessibility **and** usability)
- ▶ fix bugs related to accessibility (currently 34 open bugs: <https://bugzilla.wikimedia.org/buglist.cgi?keywords=accessibility>)
- ▶ today: accessible MediaWiki with WAI-ARIA

Problems

- ▶ fit whole applications into browser, but necessary user interface elements not available in (X)HTML
- ▶ missing elements like tree views or even simple comboboxes must be programmed
- ▶ technology: CSS, Javascript, JSF, Ajax etc.
- ▶ problem: screen reader cannot detect what your constructions **mean**

Solution: Semantics

- ▶ WAI-ARIA: W3C's mark-up for Accessible Rich Internet Applications, <http://www.w3.org/TR/wai-aria/>
1. landmarks like navigation, main content und search
 2. widgets like spinner, progress bar, slider, menu, tooltip etc.
 3. live regions for updating content
 4. useful properties, e. g. for current value, or for required or invalid input

Solution: Semantics

- ▶ there's no magic in WAI-ARIA, only semantics
- ▶ keyboard handling is still programmer's responsibility
- ▶ no official recommendation yet, but doesn't break anything (just gets ignored)
- ▶ requires IE 8 or FF 3 and modern screen reader (e. g. Jaws 11, NVDA 2010.1)
- ▶ many supporters, 3 weeks ago: also Flash will have support
- ▶ ARIA-supporting frameworks: YUI 3 (Yahoo), GWT (Google), Dijit (IBM), jQuery (AOL) ...

Solution: Procedure

1. divide application or document in meaningful regions (landmark roles plus article role)
2. identify user interface elements (widget roles)
3. define states and properties

excellent manual: <http://www.w3.org/WAI/PF/aria-practices>

Landmarks

- ▶ allow for fast navigation inside document or application
- ▶ render skip links unnecessary
- ▶ Jaws: list of landmarks (Ctrl + Ins + ;), jump between landmarks (;)
- ▶ example: `http://news.yahoo.com/`

Landmark Roles

- ▶ `application`
behaves like application (important for screenreaders)
- ▶ `banner`
- ▶ `complementary`
independent, additional content (e. g. weather forecast)
- ▶ `contentinfo`
copyrights, links to privacy statements etc.
- ▶ `form`
- ▶ `main`
main content
- ▶ `navigation`
- ▶ `search`

Possible Landmarks

- ▶ already bug fix for landmarks for Monobook skin¹, but there are some more
- ▶ identified navigation landmarks:
 - ▶ Navigation (p-navigation), relates to site
 - ▶ Interaction (p-interaction), relates to site
 - ▶ Toolbox (p-tb), relates to article and site
 - ▶ Print/Export; Languages etc., relates to article
 - ▶ Article, Discussion (p-namespaces), relates to article
 - ▶ Read, Edit ... (p-views), relates to article
 - ▶ My talk, My preferences ... (p-personal), relates to site
- ▶ should probably be nested to reflect relation

¹https://bugzilla.wikimedia.org/show_bug.cgi?id=18338 by Peter Krantz from Standards schmandards

Possible Landmarks

- ▶ identified other landmarks:
 - ▶ Logo (`p-logo`), role: banner
 - ▶ Search (`p-search`), role: search
 - ▶ Main content (`content`), role: main
 - ▶ Site information (`footer2`), role: contentinfo
- ▶ identified document landmarks:
 - ▶ Article (should start with `firstHeading`), role: article
 - ▶ Table of contents (`toc3`), role: directory
 - ▶ Categories (`catlinks`), likely role: directory
 - ▶ Mathematics (`<math>`), role: math
 - ▶ Info boxes, role: region; such boxes problematic because often put—and therefore—read first in article

²`footer-info-lastmod` relates only to the article. It may be moved into `bodyContent`.

³Element `table` is misused here!

Lost in Links

- ▶ 133 when reading article WAI-ARIA, 112 links when editing

[...] This page link **Bold** This page link *Italic* This page link Link
This page link Embedded file This page link Reference This page
link Advanced This page link Special characters This page link
Help This page link Level two This page link Level three This page
link Level four This page link Level five This page link Heading
Format This page link Bulleted list This page link Numbered list
This page link Indentation This page link No wiki formatting This
page link New line This page link Big This page link Small This
page link Superscript This page link Subscript Insert This page link
Picture gallery This page link Table This page link Redirect This
page link Search and replace Edit alt plus , [...]

Widgets

- ▶ 27 roles for user interface elements and 8 for their containers
- ▶ not everything is a link: finally semantics for user
- ▶ roles appear in screen readers' output
- ▶ Jaws: list of form fields (Ins + F5)

Widget Roles

- ▶ alert
- ▶ alertdialog
- ▶ button
- ▶ checkbox
- ▶ combobox
- ▶ dialog
- ▶ gridcell
- ▶ link
- ▶ log
- ▶ marquee
- ▶ menuitem
- ▶ menuitem-
checkbox
- ▶ menuitemradio
- ▶ option
- ▶ progressbar
- ▶ radio
- ▶ radiogroup
- ▶ scrollbar
- ▶ slider
- ▶ spinbutton
- ▶ status
- ▶ tab
- ▶ tabpanel
- ▶ textbox
- ▶ timer
- ▶ tooltip
- ▶ treeitem

Widget Roles (Containers)

- ▶ grid
- ▶ listbox
- ▶ menu
- ▶ menubar
- ▶ tablist
- ▶ toolbar
- ▶ tree
- ▶ treegrid


Possible Widgets

- ▶ menubars or toolbars
- ▶ menus with expandable sub-menus
- ▶ separators
- ▶ listboxes
- ▶ modal dialogs, important to get focus and block underlying content
- ▶ alerts, important to get focus
- ▶ and probably more I haven't thought of

If time left, demonstrate problems with Heading listbox: new elements added to end of virtual buffer. Also, if tab is used, next tab is on Format, need to shift-tab.

Useful Stuff

- ▶ Fang extension⁴ for Firefox: emulates typical screenreader output, includes headings and links list
- ▶ testing with real screen readers: use VM, read: <http://yuiblog.com/blog/2008/12/30/configuring-screen-readers/>
- ▶ have a look at frameworks that support ARIA (e. g. YUI 3)

⁴<https://addons.mozilla.org/de/firefox/addon/402> 

And Now?

- ▶ direct me to developers responsible and (hopefully) interested
 - ▶ available till Friday evening: discussion and demonstration
 - ▶ prepare concept with and for developers, find and solve problems
1. first stage: basic landmarks
can be delivered on short notice (maybe right here)
 2. second stage: nested landmarks
needs investigation and testing
 3. third stage: widget roles
dependent on Usability Initiative's output