Leveraging Web 2.0

Design Patterns For Enhanced Accessibility

T. V. Raman

Charles L. Chen

Google

Outline

- Introduction to accessibility.
- Accessible Web applications.
- Usable user interfaces.
- Injecting accessibility via AxsJAX.

Introduction To Accessibility

Everyone Has Special Needs

- Mainstream UI caters to the majority.
- Usability for the long tail?
- Mainstream UI augmented by:
  - Screenreaders, magnifiers, . . .

How Does It Work?

Adaptive technologies augment UI to:

- Produce alternative feedback,
- Enable keyboard access,
- Generate appropriate notifications.

Leverage platform-level APIs.

Web Access Via Screenreaders

Web pages treated as documents

- Users explore via the tab key.
- Speak content under focus.
- Enable structured content navigation.

Impedance mismatch with Web Apps.

Bridging The Gap

- The *document* is the *interface*!
- But an *interface* is *not* a document!
- This talk will cover user interfaces.

Goal: Overcome impedance mismatch.

Web 2.0
From Content To Applications

WebFormation

The document *is the interface!*

- Traditional documents —News
- Interactions —Shopping sites
- APIs —Programmatic lookup

Distinction blurs over time.

Aggregations And Mash-Ups

Web applications built of Web blocks

- Original Web brought content on-line
- URI Addressable content
- REST APIs $\leftrightarrow$ addressable apps

$Web\ 2.0 = Web(Web)$

Augmenting Web Interaction

- Web architecture enables flexible UI.
- Agile Web development:
  - Discover patterns that work,
  - Experiment, iterate, codify,
  - Deliver solutions that work.

Usable UI Patterns

Eliminating Feature Gap

W3C ARIA: enable AT regain lost ground.

DOM Live properties expose metadata.

Role Identifies widget type.

State Reflects current interaction state.

Live Regions Observer-observable.

Web UI gains parity with desktop GUI.

Usable UI

Accessible widgets to usable applications!

- ARIA makes UI controls visible to AT.
- Web apps are more than UI controls.
- Task completion is a true measure.

ARIA is necessary *but not* sufficient*!*

End-To-End Usability

Steps in UI augmentation

- Augment UI elements with metadata.
- Speak relevant updates.
- Add nav keys for *random* access.
- Produce feedback for user actions.
- Allow user to query for information.

AxsJAX —Injecting Usability

AxsJAX Framework

Discovering patterns for eyes-free access.

- Abstracts details of ARIA *op-codes*.
- Authors focus on UI experimentation.
- Solutions distributed via the Web.

http://google-axsjax.googlecode.com

Augmenting Web Search

Augmenting Web Search

Injecting Behavior Via AxsJAX

- Add keyboard navigation.
- Enable random access.
- Magnify *current* result.
- Produce spoken feedback.

Site-Specific Augmentation

- Site-specific augmentation declared in CNR file.
- CNR files consumed by AxsJAX framework.
- Enable rapid prototyping of site enhancements.

Sample Code

Content Navigation

Essential for efficient eyes-free access!

- Identify *useful* node-sets.
- Node-sets define *cursors*.
- Associate hot-keys to cursors.
- Declare default traversal action.

Augmenting Behavior

Adding a magic lens.

- Lens follows page traversal.
- Magnifies content under the *cursor*.
- Implemented via CSS.
- User can control magnification.

Auditory User Interface

Speak relevant information.

- Spoken feedback from:
  - ARIA-enabled Screenreaders.
  - Self-voicing browser plugins.
- Abstracts away ARIA details.

Google Reader

Support now live!

- Feed reader on steroids!
- Provides fluent spoken access.

Discover, experiment, codify, ship!

Mash It Up!

AxsJAX can make XKCD talk!

- Comic transcripts available on a Web.
- Separate from actual comic.
- AxsJAX mashes them up!

Watch The Web Take Off!

code.google.com