Web Apps — The Next Generation

Access Opportunity Or Challenge?

T. V. Raman

Google Research
Outline

- Web Applications — The access challenge
- What does accessible mean?
- Web applications — the access opportunity
- Mash It Up: Think outside the box
Web Applications
The Access Challenge
Web Apps: Advantages

*Hosted Web applications enable:*

- Easy deployment
- Light-weight user interaction
- Ubiquitous access to data
- Easy upgrades

*Today’s access technologies do not fit this model.*
The Impedance Mismatch

Major shift in application deployment model

- Web Apps — The document is the interface.
- Light-weight UI hosted in Web pages.
- AT assumes desktop application model.

App model shift requires shift in AT.
When Web Apps And Desktop Screen-readers Collide

- AT installed on client workstation
- Depends on native UI widgets
  - All of the disadvantages,
  - And none of the advantages!
Ubiquitous Access

The Access Challenge

- Web promises anytime, anywhere access
- Equal access for users with special needs:
  - Email access at airport?
  - Edit/share information from a kiosk?
What Does Accessible Mean?
Access Goals

- Retain present level of access to functionality
- Increase reach by enabling wider access
- Wider access:
  - Bring within reach of more users
  - Enable access in more user contexts

Important to go beyond the status-quo
Access Building Blocks

\((\text{Content, UA, AT})\)

- Together determine overall user experience.
- Content: Capture adequate semantics
- UA: Degrade gracefully
- AT: Bridge the gap
Building Speech Access

- Identify *what* to speak
- Determine *how* to speak it
- Decide *when* to speak
What To Speak

- Rich markup for Web content
- Separate content from presentation
- Structure content to reflect its intent
- Identify role of content particles
- Expose current state via DOM properties

W3C DHTML Road-map
How To Speak

**Aural CSS — It is Finally Time!**

- Speech solutions need to implement ACSS
- CSS display values no longer sufficient
- Leverage media-specific CSS sections

**Aural CSS: Key styling API for auditory output.**
When To Speak

*Event handlers determine behavior.*

- Event handlers implement web interaction
- Eventing determines *when* things change
- Spoken feedback to reflect visual updates
Intent Based Events

Affecting the interface

- User actions raise events
  - Select, deselect
  - Activate
- Use intent-based events for maximal flexibility
- Enable *late-binding* of UI peripherals

*Final application has wider reach.*
Web Applications
The Access Opportunity
Web Application Model

- Data resides on the network
- Interaction resides on the client
- HTTP operations to synchronize data
- Browser widgets to create UI

*Shift away from monolithic applications*
Web Adaptive Technologies

Adaptive technologies embrace, extend Web model

- AT dynamics no different from mainstream
- Web applications fulfill new needs
- Web AT access enables Web-based tools

Evolve today’s AT to meet tomorrow’s needs

Web Apps — The Next Generation – p. 18
Web Application Container

Web browser functions as universal client

- UI realized through Web pages
- HTML for creating content
- CSS for styling
- DOM eventing for adding behavior

Exposes client-side interaction logic
The Access Opportunity

*Separation of interaction from data:*

- Opens up opportunities for custom clients
- Optimize user interaction to user’s needs
- Multiple UIs can collaborate

*One size no longer need fit everyone*
Adapting To The User

One size need not fit everyone

- CSS for custom styles
- Atom/RSS for content syndication
- XForms for rich interaction
- XBL for custom behaviors
- Atom Publishing Protocol for data APIs

All these technologies are available in Firefox today
New Adaptive Technologies

**New opportunities for AT vendors:**

- A new market for consumer applications
- Custom services tailored to end-user needs
- Task-driven access tools

*This generation of AT will be user-driven.*
Mashing It Up With Web APIs
Web APIs

Separation of content from interaction:

- Leads to light-weight Web APIs
- Atom/RSS based syndication
- AJAX APIs for hosting services
- Examples: Google Maps, Google Calendar
- Web mashups are an automatic follow-on

What is the access equivalent of a mashup?
Essence Of A Mashup

 Syndicate data sources into a custom UI

- Add screen-enlargement (zooming)
- Augment Web UI with spoken output
- Overlay simplified skins
- Create custom aggregations of Web apps
Innovative Web AT

- **Draw inspiration from on-line audio games**
  - Audio Games
  - HTML DOM with Javascript for audio games

- **Games often lead to UI innovation**
Conclusions

- Important to build on what we have.
- But limiting to present AT too limiting.
- Web AT targets consumer products.
- Needs to leverage advantages of Web model.
Watch Web Access Take Off!