Accessibility in Rich Internet Applications

Bob Regan

Adobe Systems

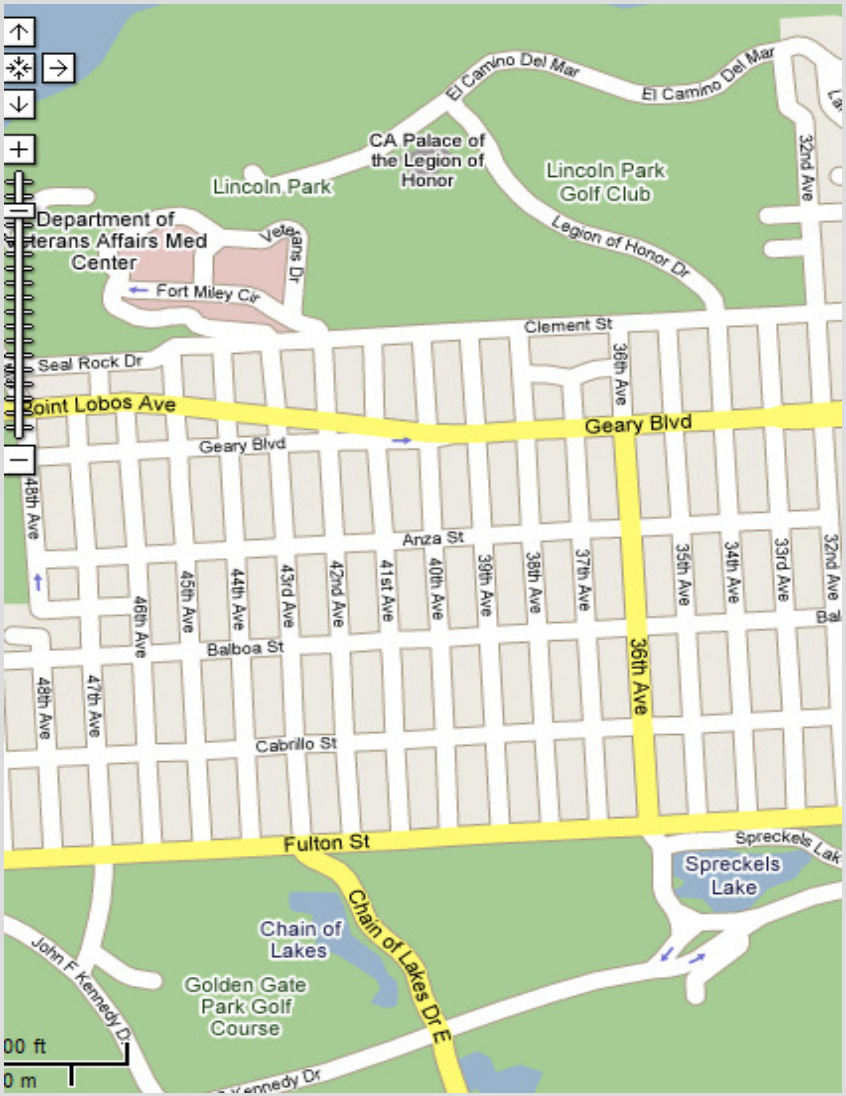
October 3, 2006

Overview

- What is a Rich Internet Application?
- Anatomy of an Accessible RIA
- Disability Use Cases
- Testing an RIA for Accessibility
- RIA Development Strategies
- Key Concepts
- JK Rowling.com Case Study
- Resources

Adobe

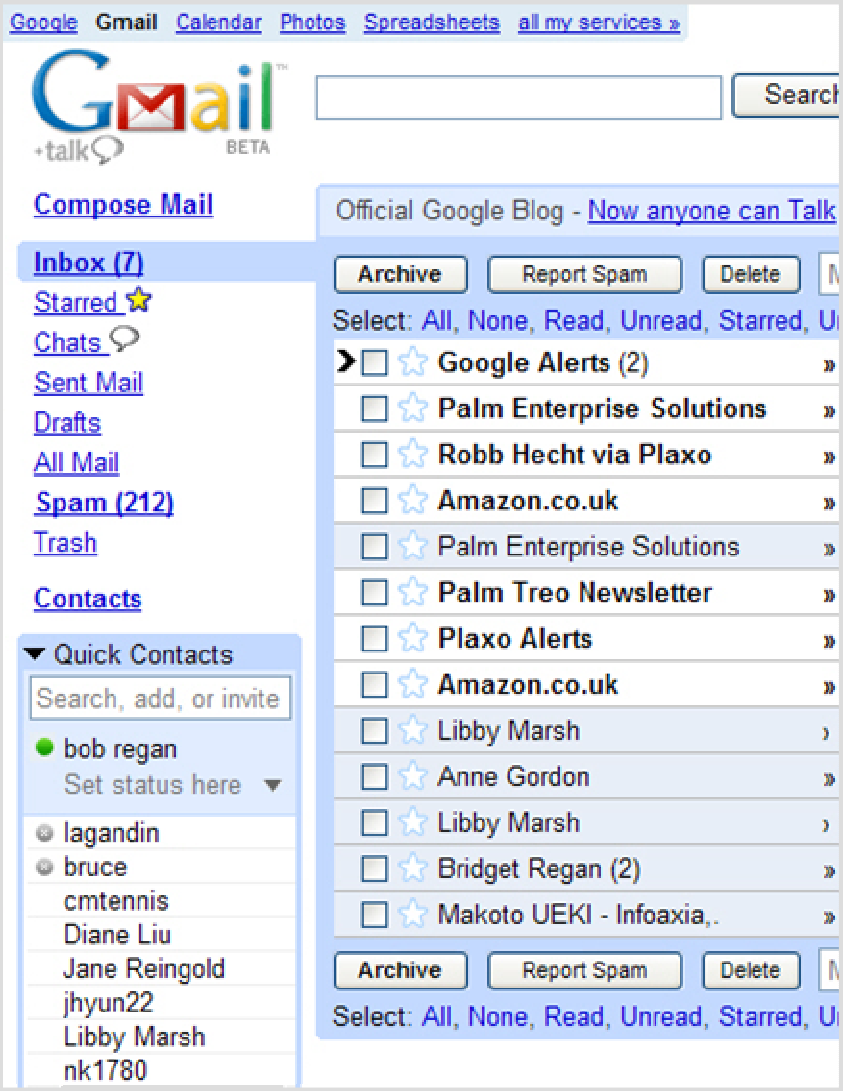
What is a Rich Internet Application?



- A web application with desktop software features and functionality
  - Single screen updates
  - Diverse controls
  - Live data updates
- Technologies:
  - AJaX
  - Flash (Flex)

Adobe

Anatomy of an Accessible RIA



An accessible RIA reflects:

- Standards
  - W3C – Web Content Accessibility Guidelines
  - Interoperability Standards
    - Microsoft Active Accessibility
    - DOM based mapping to MSAA
    - ATK – Linux
    - Mac Accessibility API

Adobe

Anatomy of an Accessible RIA

An accessible RIA reflects:

- Standards
- Interoperability with Assistive Technology
  - Screen Readers
  - Screen Magnifiers
  - Keyboard

Adobe

Anatomy of an Accessible RIA

An accessible RIA reflects:

- Standards
- Interoperability with Assistive Technology
- Usability

Adobe

Disability Use Cases

- A person who is blind
- A person with a mobility impairment
  - Someone who relies entirely on the keyboard
  - Someone who uses the mouse but has trouble with small targets
- A person with low vision
  - Someone who requires a screen magnifier
  - Someone who requires larger text
- A person who is color blind
- A person who is deaf
- A person with a cognitive disability

Adobe

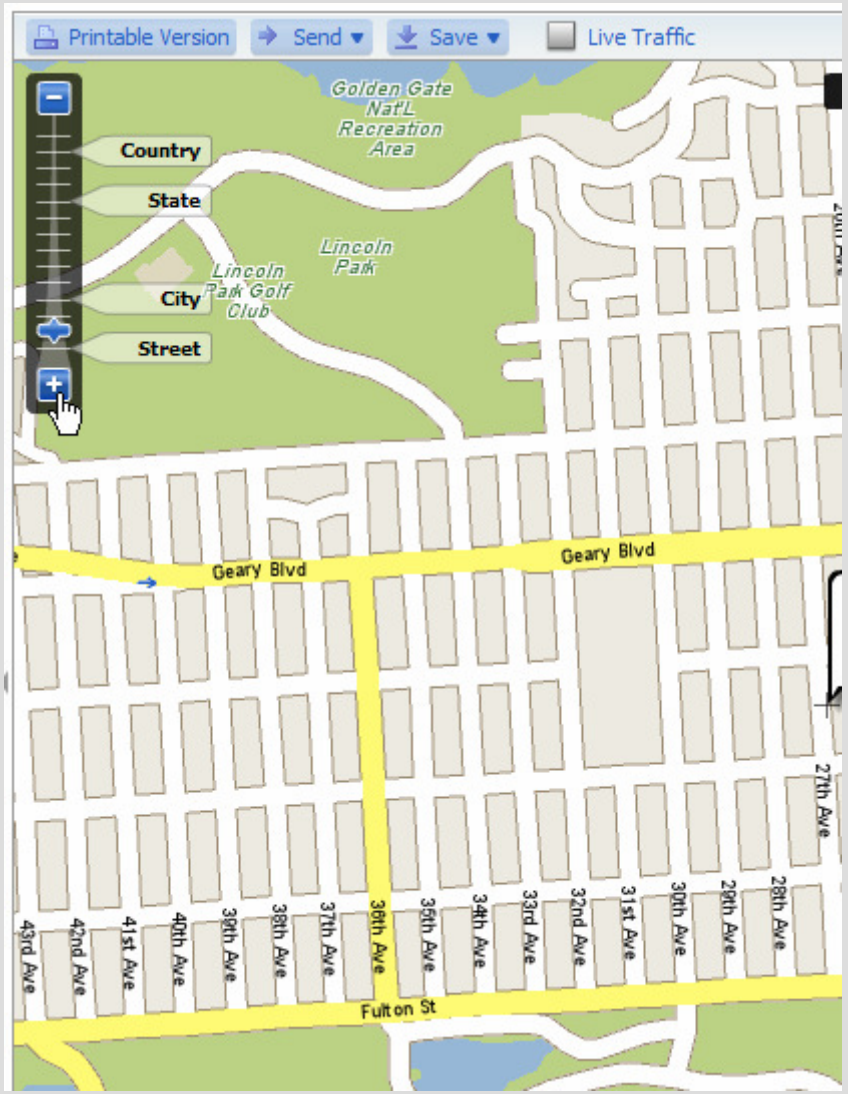
Testing an RIA for Accessibility

- Designers are visual in their way of looking at the world
- Devote time before project to learning the screen reader, then use it
- Test for accessibility are regular intervals
  - Test for screen reader access at least twice times a day
  - Test other use cases at least once a week (less on compressed schedules)
- Involve people with disabilities in the process
  - User testing for large scale projects
  - Email based feedback for smaller projects

RIA Development Strategies

- Unstructured RIA development
  - Using Flash Authoring Tool
  - Using hand coded AJaX
- Structured RIA development
  - Using Flex
  - Using an AJaX component framework

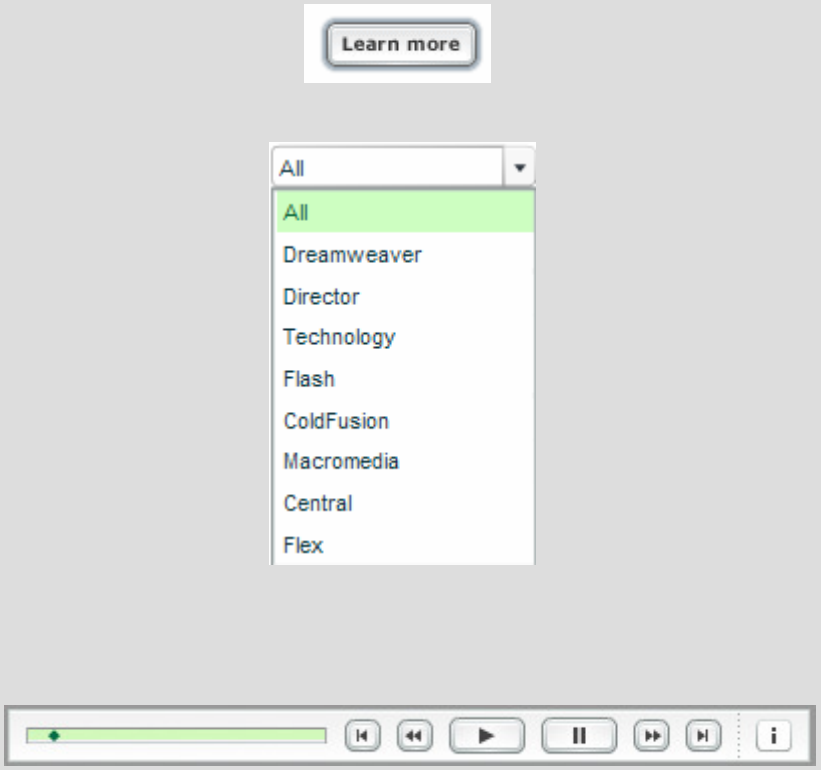
Key Concepts



- **Label**
  What is this thing?
- **Role**
  What does this thing do?
- **State**
  Is this thing on or off?
- **Structure**
  How does this thing relate to the rest of the things on the screen?

Adobe

Key Concepts: Label



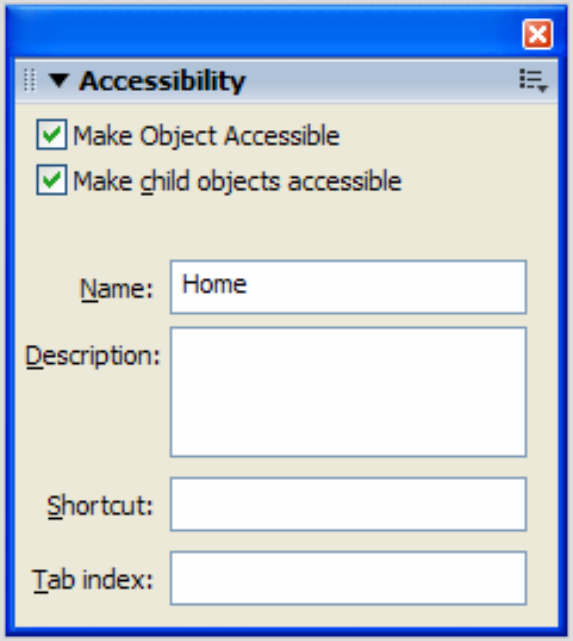
- Label differentiates repeated instances of the same control
- Every control should have an associated label
- Label should describe function
- If function changes, so should label

Adobe

Key Concepts: Label

- Negative Examples
  - HP Experiences
    welcome.hp.com/country/us/en/msg/corp/flashgateway.html
    - Navigation Bar not labeled, read as ‘button, button, button’
- Positive Examples
  - Sample XML Nav Bar
    weblogs.macromedia.com/accessibility/files/pagemodel/pagemodel.html
    - Similar to nav bar used in previous example with equivalents defined.
  - Source file

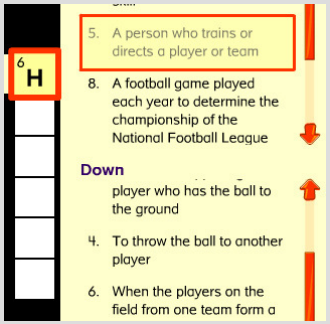
Key Concepts: Label



- Assigning labels using text equivalent
  - In Flash, labels can be assigned using the `.name` property.
  - In HTML, labels can be assigned using the `alt` attribute
  - When building a control that has an analog in HTML, use the same descriptions found there
  - When building unique controls, provide more detailed descriptions with functional details or instructions

Adobe

Key Concepts: Role



- Screen reader user should know what every control does
  - Buttons correctly identified
  - Controls emulating standard windows controls should be identified
  - Unusual controls should provide cues to users as to their identification, operation and state information

Adobe

Key Concepts: Role

- Negative Example - AJaX
  - Google Maps
    maps.google.com/maps
  - Controls not exposed at all.
- Positive Example – Flex
  - Flex Blog Reader
    flexapps.macromedia.com/flex15/blogreader/blogreader.mxml?accessible=true

Key Concepts: State

- Every control should indicate:
  - Current selection.
  - Number of possible selections.
  - Update when selection changes.

Adobe

Key Concepts: State

- Positive Example - Flash
  - GeoNet
    www.eduplace.com/geonet/geonet.html
    - Notice that the score board on the first screen includes text hints.
    - The level selector inside updates as the users changes levels.

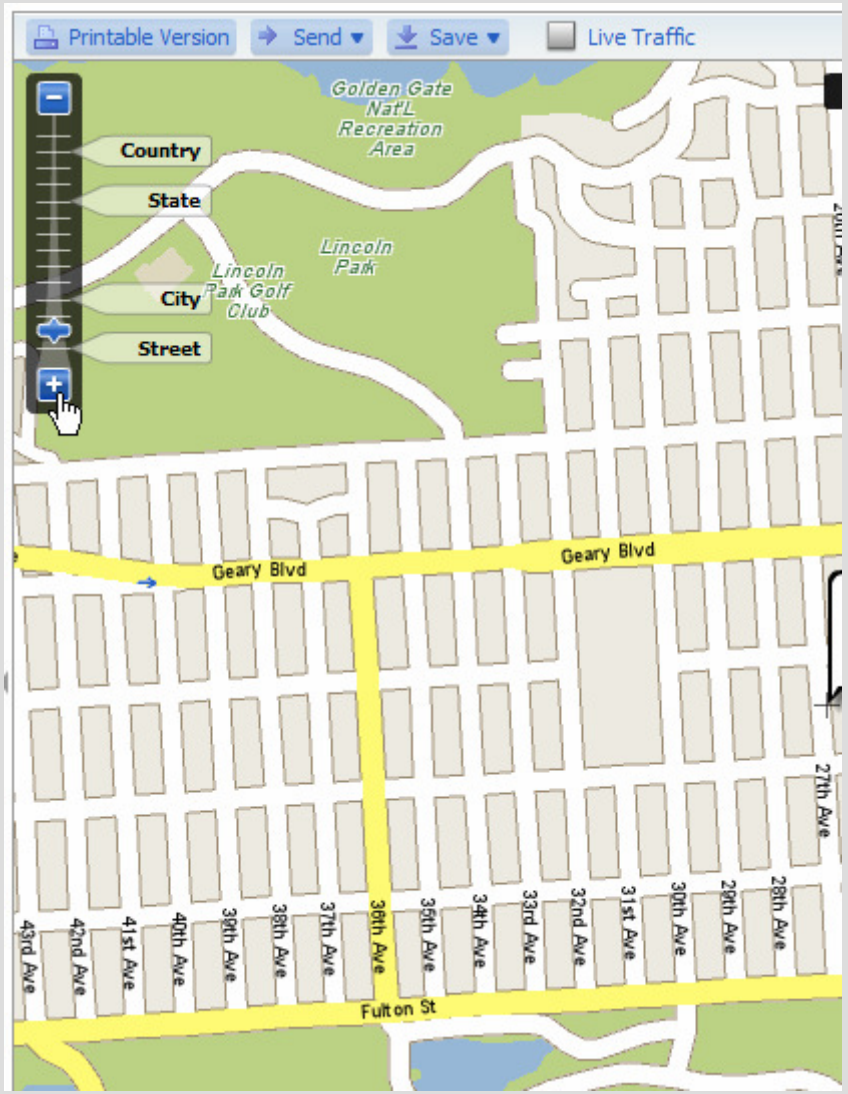
Key Concepts: State

- Provide a text field for visual indicators.
- Update info as state changes
- Use accessible components for complex controls.
  - Accessible components include MSAA support to dynamically deliver this content.

Key Concepts: Structure

- Related controls should be read as a group
  - Make sure buttons from navigation are not mixed in with text
  - Ensure buttons in different columns or rows are read together
  - Ensure that elements that are obscured or off-stage are not read.
  - Make sure portlets or ‘pods’ and their contents are read as a group
- Hierarchical relationships between elements should be exposed
  - As in levels of navigation within a tree control

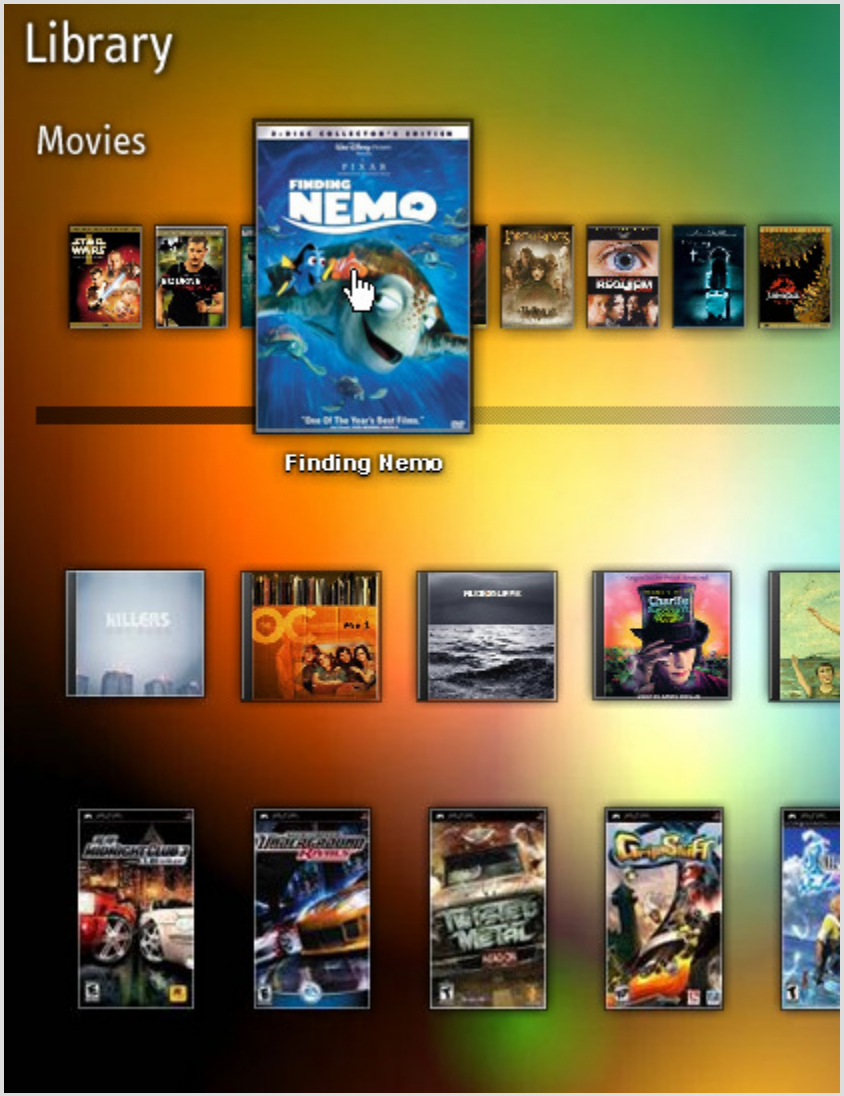
Recap: Key Concepts



- **Label**
  What is this thing?
- **Role**
  What does this thing do?
- **State**
  Is this thing on or off?
- **Structure**
  How does this thing relate to the rest of the things on the screen?

Adobe

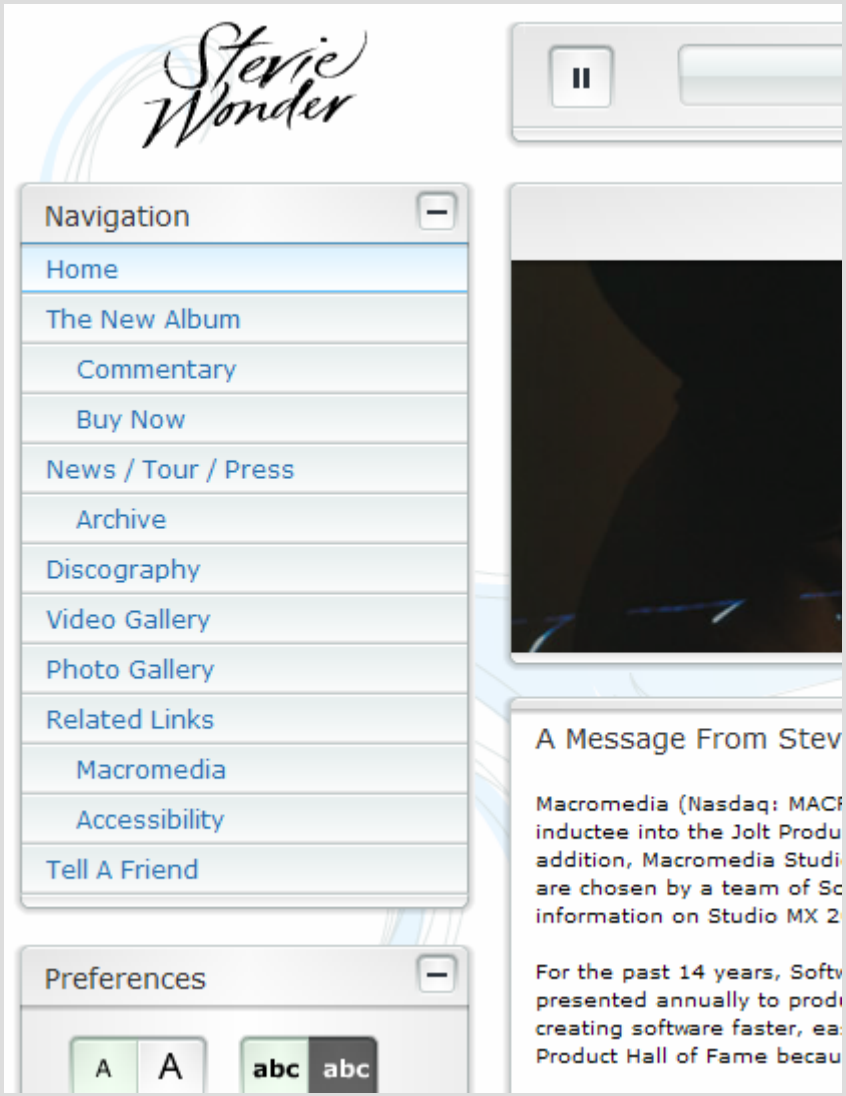
Key Questions: Choreography



- Choreography describes how objects on a screen interact with one another
- In Rich Internet Applications, animation is used to put the things that are important to a given task in front of the user

Adobe

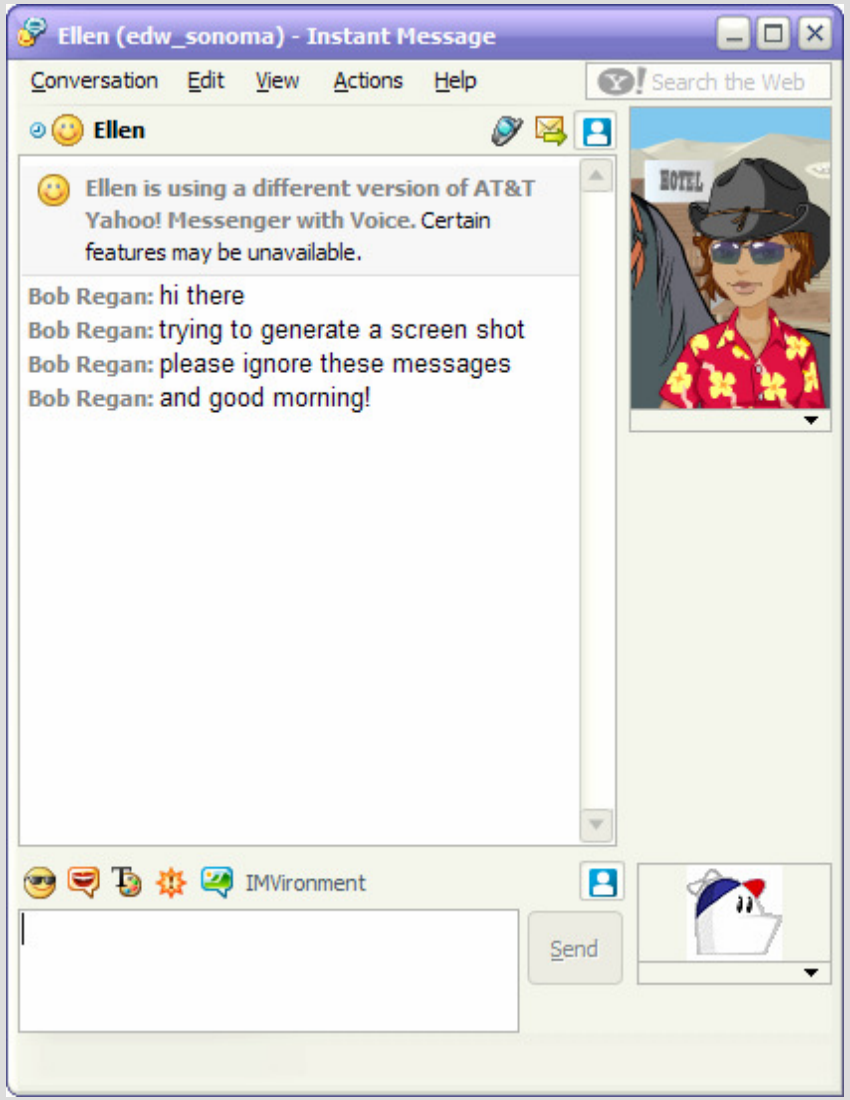
Key Questions: Choreography



- Choreography describes the way multiple objects interact with one another
- In Rich Internet Applications, animation is used to put the things that are important to a given task in front of the user

Adobe

Key Questions: Screen Updates



- A common feature of an RIA is that the screen might change layout, or change content visually without a specific user initiated event
- Critical updates must be communicated to the user
- IM Background sounds provide an exemplar
  - *Changes in presence of users I subscribe to*
  - *Notification of new messages*
  - *Notification of a successfully message that has been sent successfully*

Adobe

Design Patterns: A Review of JK Rowling.com

- Labels
  For context and character
- Progressive Disclosure
  Presenting info only as needed
- Keyboard Shortcuts
  Creating equivalent controls
- Handling Audio
  Delivering equivalents and providing cues

Adobe

Design Patterns: Labels

Design Patterns: Labels

- Top level introduction sets the stage:
  - *Welcome to my desk, which was specially tidied for your visit. Please wander around and explore all of the objects you find here.*
- Each of the other areas lets the user know where they are:
  - *You have entered the extra stuff area, the news board is full of interesting bits and pieces for you to browse around*
- Label for the phone provides information on the control, within the context of the experience
  - *Mobile phone, click here to pick up the phone button*

Design Patterns: Progressive Disclosure

- Original phone was tedious
  - Contains 13 buttons
  - Positioned near the top of the reading order
- Grouped buttons into a single control
  - Label explains grouping
- Once activated by user
  - Individual buttons exposed
  - Other items on stage hidden
  - Close button added

Adobe

Design Patterns: Keyboard Shortcuts

- Site includes several objects that can be manipulated by the mouse
- Must provide keyboard equivalent action that:
  - Allows user to accomplish same task
  - Removes the obstacle represented by the task to continue

Adobe

Design Patterns: Handling Audio

- Captions for audio built into the user interface
- Audio used to reinforce events on the site
  - Not required for success on first visit
  - Nature of site makes repeat visits likely among users
  - Glossary provided to make use of sounds easier

Adobe

Resources

- Accessibility in Rich Internet Applications
  my.adobe.acrobat.com/riaaccess/
- Adobe Accessibility Resource Center
  www.adobe.com/accessibility
- AJAX and Screenreaders: When Can it Work?
  www.sitepoint.com/article/ajax-screenreaders-work
- Flash Accessibility
  www.adobe.com/resources/accessibility/flash8/
- Flex Accessibility
  www.adobe.com/macromedia/accessibility/features/flex/
- W3C - Dynamic Accessible Web Content Roadmap
  www.w3.org/WAI/PF/roadmap/DHTMLRoadmap040506.html