The Fluid Project: An Open Community for Inclusive Design

Colin Clark, Fluid Project Technical Lead, Adaptive Technology Resource Centre, University of Toronto
Topics We’ll Cover

- Project vision and goals
- The Fluid community
- Improving the user experience
- Technology framework
- Road map
- How to get involved
Project Vision and Goals
Vision

"Advance the status of UI development and design in academic community source projects"

"...so that they can fulfill their potential as platforms for innovation"

"Create a community of UX expertise"

"Build a presentation layer across applications that can support the diversity of needs within higher education"

"Support the precarious values of usability and accessibility"
The Context for Fluid

"Issues of usability & accessibility are significant for community and open source software

"A great user experience is the next step for wide open source success and adoption

"Our goal is to incrementally improve the overall user experience of uPortal, Sakai, Kuali Student, Moodle, and other projects
Fluid's Approach

"Cross-project collaboration:
- Share scarce UX resources across projects
- Solve common challenges
- Recognize recurring user interface idioms"

"A holistic approach: combine technology & design"

"A two-fold path:
- Social: build a community around UX
- Technical: new UI development tools"
Accessibility Vision

" Disability is an artifact of the environment:
   – A mismatch between user and the system

" Embrace diversity:
   – Users all have different needs
   – Under different circumstances

" One size never fits all

" Build systems that can bend and adapt to meet the users' individual needs
The Fluid Community
Participating Projects

" uPortal
  – Enterprise portal system
  – Aggregates personalized student information

" Sakai
  – Collaboration and learning environment
  – Teaching, research, and group collaboration

" Moodle
  – Learning management system
  – Strong focus on pedagogy

" Kuali Student
  – Upcoming, next generation student system
  – Viable alternative to high-cost commercial products
Fluid is looking at challenges faced by all open source projects:

- How do non-technical people get involved in OSS?
- How can we help designers and developers speak the same language?
- How do you do user testing in a distributed environment?

Our work on DHTML accessibility and personalization will drive open standards

We'd like to collaborate with and learn from you!
Who is Involved?

" Partnership among several universities and corporations
" Toronto, UBC, UC Berkeley, York, Cambridge, Michigan State, and others
" IBM, Sun, and Mozilla Foundation
" Broad range of experience
Interaction Design

" Start with heuristic and usability reviews
" Baseline for usability and accessibility
  – Know where we need to improve
  – Prioritize the pain points
" User research: what are our users' goals?
" UX Toolkit: shared design resources for usability and accessibility
" Design new solutions
What are we going to build?

" Rich, flexible DHTML user interface components
  – Reusable components: work across applications
  – More than just widgets
  – Easy to wire up and customize

" New JavaScript development framework
  – Technical infrastructure for pluggable components
  – JavaScript libraries for DHTML accessibility
  – Builds on existing toolkits

" Open accessibility standards

" Integration with open source projects
User Experience & the UX Toolkit
UX Toolkit

" All the stuff you need to design great interfaces
" User interface components
" UI Design Patterns
  – Good advice when designing UIs
  – Take material from Tidwell, Yahoo!, and other patterns
" UX Walkthroughs & Distributed User Testing
  – Driven by the communities, but collaborative
  – Reusable protocol and checklist
  – VULab
" User Persona Library
  – Leverage the design patterns library
Components

Components are recurring interactions:
- Navigation: wizards, sequences, workflows
- Content: file management, uploading, attachments
- Direct manipulation of objects

Choosing components will be based on:
- Analysis of existing applications across projects
- Recognizing common UI idioms in other applications
- Solving the most frequent and severe problems
UX Walkthroughs: Goals

- Assess what we've got: identify user pain points
- Identify “componentizable” solutions
- Drive our development priorities
- Provide baseline for future evaluation
- Create shared protocol & process for usability and accessibility
  - …that fits smoothly into ongoing development processes
Challenges/Opportunities

"Checking an application for
  – usability
  – access for a screen reader user, screen magnifier user
  – access for someone who can’t use a mouse
could require different evaluators, multiple passes… is this essential?
" Can we bring accessibility walkthroughs to the level of maturity of usability walkthroughs?
" Can we adapt walkthroughs to components rather than applications?
" Can we combine “walkthroughs” and “heuristic evaluations”?
How We Do Walkthroughs

" Working group
  – 3 to 5 evaluators on each project
  – Usability and accessibility focus

" Define process and priorities:
  – Iterative
  – user profiles
  – scenarios of use for cognitive walkthroughs

" Perform individual evaluations

" Synthesize evaluations and prioritize

" Brainstorm design solutions

" Collaborate with participating communities
U-Camps

Our main educational effort:
– Everyone should have a basic UX vocabulary
– Share a repertoire of viable UX techniques
– Opportunity for designers and developers to collaborate
– Loose agenda, open participation

Two successful U-Camps so far, two more planned:
– November 12 at Rutgers University
– December 3 in Newport Beach, California
Technology
Technical Goals

" Make it easier for developers to build better, more accessible user interfaces
" Support collaboration with designers
" Make it easier to share designs within a community
" Enable components to be adapted for a variety of tools and workflows
" Embrace the Web
Unique challenge: how to enable support for very diverse presentation technologies?

Based on JavaScript, DHTML, and AJAX

Thin binding layer between client and RESTful, largely stateless server

Loose coupling, works across applications

In translation:

– Web 2.0 made more usable & accessible
What is a Reusable Component?

"On the client-side, a Fluid component consists of:
- One or more HTML templates
- One or more layers of CSS
- JavaScript for behavioural logic
- Accessibility metadata (control, presentation, etc)

And on the server-side:
- Binding conventions: markup, RESTful server callbacks
- The ability to deliver the appropriate markup, metadata, and user preferences"
Anatomy of a Component

Anatomy of a Fluid Component

**CSS**
- styles
- colours
- sizes
- fonts
- borders
- focus
- etc.

**HTML**
- structure
- ARIA metadata
- labels
- controls

**JavaScript**
- UI logic
- client-side validation

**Metadata**
- detailed roles
- modality
- control types
Fluid Accessibility

"Web 2.0 will be accessible
  – it’s just a matter of time
"ARIA: Accessible Rich Internet Applications (W3C)
"AccessForAll for component metadata
"Ongoing toolkit accessibility support
  – Dojo and others
"Design specific alternatives
"Fluid: Accessibility from the ground up
Fluid will be a highly flexible UI layer
Differing needs of individuals and institutions

At configuration-time:
- Branding, appearance, and styling
- Choose the functionality and experience available

At run-time:
- Swap in accessible controls
- Provide high contrast, large print, etc.
- Components designed for different user needs
Composition = Flexibility

" Fluid components are built out of smaller units
  – Keyboard handlers
  – Layout managers
  – Server callbacks

" Composition enables flexibility
  – At runtime, wire up alternative behaviour
  – Use web standards to change presentation (HTML/CSS)

" Easy to extend or modify component behaviour
Component Composition
The Fluid Framework

" Common accessibility APIs:
  – Focus management
  – Keyboard handlers
  – Getting/setting ARIA properties

" Framework infrastructure:
  – Dependency injection
  – Server-side communication
  – Portal-friendly DOM conventions

" Adaptation:
  – The ability to wire up component behaviour at runtime
The Lightbox & Reorderer

" Component design strategy:
  – Start with solving a real problem
  – Extract reusable code in libraries

" Our first UI component: The Lightbox

" Sakai's Image Gallery:
  – No way to sort images in albums
  – Opportunity to explore direct manipulation on the Web
  – Hard, interesting accessibility challenges

" Built this code up into fully accessible sorting library
Drag & Drop Accessibility

" Start from scratch: focus on the goal, not the task
  – Reordering images
  – Doesn’t necessarily look like drag and drop
  – What alternatives are available on the desktop?
  – Cut and paste-style interactions
  – Shifting images like on a real light table

" What does accessibility mean here?
  – Keyboard access
  – Support for magnification and linearization
Lightbox Demo
Project Road Map
Road map

" Ongoing UX Walkthroughs & user testing
  – Refine protocol and checklist, share with other projects

" Lots of user research and design
  – Navigation schemes in complex multi-tool apps
  – File and content management

" New component development
  – File uploader, browser, picker
  – Tab-based navigation, menus, portlet navigation

" Work on open accessibility standards
  – User preferences, UI metadata
In Summary

"For more information, visit the Fluid Project web site:

www.fluidproject.org

"Goals:
– Better, more inclusive web development tools
– Foster a vibrant open UX community

"Join our community, everyone is welcome!