The Fluid Project
An Open Source Community for Inclusive Design

Colin Clark, Fluid Project Technical Lead, Adaptive Technology Resource Centre, University of Toronto
What is Fluid?

A community source project which creates user experience tools and software capable of addressing the needs of a diverse set of demanding and creative users and projects.

Core Institutions: University of Toronto | UC Berkeley | York University | University of British Columbia | University of Cambridge

Other Participating Institutions: Michigan State | University of Colorado | University of Michigan | Georgia Tech | UK Open University | Your University Here

Participating Projects: uPortal | Sakai | OpenCollection | Kuali Student | Moodle | ATutor

Corporate Partners: Mozilla Foundation | Sun Microsystems | IBM | Unicon
Tools must be usable and undemanding

- Users should focus on teaching, learning, research and administrative tasks… not on operating the tools

- Institutions should invest in furthering research and learning not in…ballooning support needs

- Institutions should not be concerned with cost of tool rejection and difficult implementations

- Tools should be platforms for innovation
Currently...

- Systemic problem of poor and inconsistent user interface
- Often left to programmers
- Tackled at the end
- Redundantly developed
- Inadequately tested and refined
- UX designers not well integrated into development culture
- Poor UX an impediment to adoption
- And....
“You say tomato, I say tomato, lets call the whole thing off”

- Academic communities are very diverse
- We differ greatly in our preferences, needs, habits, concepts, comforts, convictions….
  - Institutional preferences and branding
  - Conventions of academic discipline
  - Cultural differences
  - Linguistic differences
  - Differences related to age
  - Differences related to role and perspective
  - Different teaching approaches
  - Different learning approaches
  - Disability and environmental constraints

*The academic community fosters and thrives on diversity!*
Fluid:
“Flexible User Interface”

• Swappable styles
• Modular, reusable UI components
• Either runtime transformation for unique needs of individual
• Or customization at configuration
The Fluid Approach to UX in Community Source

- UX is a challenge for all open source projects and all institutions
- Cross-project collaboration:
  - Share scarce UX resources across projects
  - Solve common challenges
  - Recognize recurring user interface idioms and needs
- Fluid is looking at common problems:
  - 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?
What are we Building?

- Rich, flexible, reusable user interface components
- Lightweight JavaScript development tools
- User Experience Toolkit
- Great Interaction Designs

Interconnected activities
UX Toolkit
UX Toolkit

• UI Design Patterns
• UX Walkthroughs
• Testing techniques
• User profiles

• All the stuff you need to design great interfaces!
Designing Components

• Components are recurring interactions
• Encompass familiar activities on the Web:
  – Working with files, uploading, finding stuff
  – Navigating through content and tools
  – Rich interactions: drag and drop, etc.
• Activities and contexts, not just controls and widgets…
UX Walkthroughs

- Provide a tool that communities can use to assess their own usability and accessibility
- Identify user pain points and solutions
- Share simple, approachable techniques
- Anyone can do a UX walkthrough:
  - Try out our checklists and heuristics
  - We’re here to help you get started
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

• Join us on Wednesday at 3:30 pm for the uCamp!
Virtual Usability Lab

- Open source distributed usability testing
- Competition to expensive tools like Morae
- Before and after survey questions
- Remote screen recording
- No installation required
- Mouse and keyboard tracking
- Designed within a community that needs it!
UI Design Patterns

• A pattern is a proven solution to a common problem in a specified context
• Practical tool to help designers and developers choose the right interface for the job
• Advice on how to use Fluid components
• Share patterns across communities
  – Tag, customize, adapt for your context
• Open Source Design Patterns Library:
  – The first truly open, collaborative pattern repository

http://www.uidesignpatterns.org/
Component Architecture
Technical Goals

• Make it easier for developers to build better, more accessible user interfaces
• Support collaboration with designers
• Foster sharing of design and code
• Adaptable for a variety of tools and workflows
• Embrace the Web
• Support diverse presentation frameworks
• Don't reinvent the wheel: leverage good existing technologies and fill the gaps
What is a Fluid Component?

- **Client-side:**
  - HTML
  - Style sheets
  - JavaScript
  - Accessibility metadata

- **And on the server-side:**
  - Ability to respond to RESTful requests
  - Ability to deliver the appropriate markup and data
A Flexible Framework

- Solve the need for reuse and accessibility together
- Components need to adapt to different contexts:
  - Available screen real estate
  - Type of content
  - Amount of content
  - Method of control and navigation
- Leverage the web’s strength in separating structure from presentation
- Augment with alternative behaviours
UI Adaptation

• Flexible layouts and linearization:
  – Expandable spacing, sizing, fonts, layouts
  – Flatten multi-column views into a single column

• Enhanced Navigational Aids:
  – Turn on/off sitemaps, summaries, and breadcrumbs
  – "Focus mode:" collapse distracting or extraneous screen real estate

• Keyboard support:
  – Shortcuts: configure or remap them as needed
  – Navigation: comprehensive or quick navigation
Adaptation Illustrated

Heading One

Suspendisse cursus laqueat nisl. Duis orci. Aenean eu arcu. Pellentesque habi-
tant morbi tristique senectus et netus et malesuada fames ac turpis egestas.
Aenean egestas a est. In et turpis a est dictum eleifend. Suspendisse poten-
ti. Elitam ornare, elit consequat volutpat lacinia, purus leo vestibulum nisl, non
posuere justo eros sed felis. Cras id sapien. Praesent interdum tristique tellus.
Elitam placerat. Cras eget purus.

Heading Two
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

CSS

Style Sheets

Fluid Component

HTML Markup

Keyboard Mappings

Layout Handlers

XmlHttpRequest

Server Callbacks

Fluid

Designing software that works - for everyone
The Fluid Framework

• Make DHTML accessibility a lot easier:
  – Focus management
  – Keyboard handlers
  – Getting/setting ARIA properties

• Framework infrastructure:
  – Unobtrusive DOM binding
  – Server-side communication
  – Portal-friendly packaging conventions
  – Client-side template rendering

• Adaptation:
  – The ability to wire up component behaviour at runtime

• As small as possible…
Fluid Framework Illustrated

Components

- UI Adaptation Engine
- Reorderer
- Text Editing Service
- Dependency Management
- Template Renderer
- Accessibility Plugins
- DOM Binding
- Universal View Bus (AJAX)

jQuery
What We’re Not Doing

• Writing yet another JavaScript toolkit
• Writing more of the same widgets
• Expecting everyone to agree
What We Are Doing

• Reusing existing toolkits and technologies
• Addressing the gaps in existing offerings:
  – Accessibility
  – Personalization
  – Client/server cooperation
• Making tools that are aimed at the Web developer skill set, not only the Enterprise Java types
• Working with UI designers to create great components that encompass user activities
Accessibility Plugins

• Part of Fluid 0.3; will be included in jQuery UI base
• Keyboard navigation and assistive technology
• DHTML accessibility often requires a lot of custom code: 50+ lines for handling simple lists
• Fluid makes it easy:

```javascript
// Make the menu focusable with Tab.
var menu = jQuery("#menu").tabbable();
// Make the items selectable with the arrow keys.
var menuItems = jQuery("li", menu);
menuItems.selectable(menu, {
    willSelect: function(item) {
        item.addClass("highlight");
    }
});
```
Portal Friendly

- In the era of Web 2.0 mashups and portals, it’s much harder to write code that plays nice with others
- Fluid components are built to work with portals
- Everything is namespaced
- Support for multiple instances
- DOM searches are constrained to fragments
Unobtrusiveness

- Fluid components expect to find “interesting things,” but they don’t expect a specific DOM structure
- Selectors form a binding between code and template
- Users can change templates and tell the component how to find things
- Easily customized with plain old HTML & CSS
- Next step: template rendering without all the other stuff…
Template Rendering

Server

Server-side Presentation Framework

Client/Server Divide: HTTP

Browser

Fluid Client-side Template Renderer

Fluid JavaScript Components

Components bind with markup via DOM selectors and jQuery

Rendered Markup

Pure HTML Template

Unobtrusive Template Bindings

Fluid

Designing software that works - for everyone
Code Example: Portal Friendlines & Unobtrusiveness

```html
<ul id="fluidTabsComponent">
  ... tabs go here ...
</ul>
<div id="panels">
  ... Panels go here ...
</div>

<script type="text/javascript">
  new fluid.Tabs("fluidTabsComponent", {
    tabContainer: "#tabs",
    tabs: "#tabs > li",
    panels: "#panels > div"
  });
</script>
```
Learning More…

• Fearless JavaScript workshop: podcast coming soon!
  – http://wiki.fluidproject.org/x/7IMk

• Overview of DHTML Accessibility:
  – http://wiki.fluidproject.org/x/U4kk

• Developer’s Checklist:
  – http://wiki.fluidproject.org/x/-QQa

• jQuery a11y plugins:
  – http://wiki.fluidproject.org/x/MhYa

• JavaScript Resources:
  – http://wiki.fluidproject.org/x/6IMk
Project Roadmap
Infusion: A Complete Package

- Components for managing your files and more
- Framework: everyone can build components
- UI design patterns
- User research you can use
- Documentation and lots of sample code
Release Plan

July 2008
Fluid 0.4
- toolkit accessibility
- framework definition

January 2008
Fluid 0.8
- UI accessibility adoption
- file management components
- sustainable, active community

May 2008
Fluid 0.3
- framework development
- open design patterns library

October 2008
Fluid 0.6
- new components, framework refinements
- expand UX toolkit

March 2009
Fluid 1.0
- APIs stabilize

Build and Share Your Own Fluid Components

Early Adoption; Help Shape the Direction of Fluid

USER EXPERIENCE
What’s in Fluid Infusion 0.3?

- Lightbox
- Layout Customizer
- Uploader
- Plugins for keyboard accessibility
- Robust cross-browser support
- New layout and upload design patterns

Fluid Infusion 0.3 beta 1 was released on Friday!
Lightbox

Gallery Collections
arks (5)
wollch Village (6)
gyoccrapers (9)
ork City (all) (27)
Manhattan (11)

Lower Manhattan Collection

Start Slide Show

Sort order  Instructor default  Alphabetical

View an image by pressing Enter. Move an image using CTRL+arrow keys.

wall street.jpg
lower manhattan street.jpg
rebuilding.jpg
approaching wall street.jpg
lower manhattan.jpg
nyc building.jpg
near city hall.jpg

wtr what's left.jpg
battery park.jpg
wtr subway.jpg
Fluid in uPortal

- Next update of uPortal: “Fluid Infused”
- Fluid Layout Customizer for organizing portlets
- A whole new way to support keyboard navigation
- Thanks to Eric, Jen, and Gary for their help
- Upload in portlets
  - AddressBook
  - Briefcase
  - Calendar
  - Messaging
  - Your portlet?
Fluid 0.4: July 2008

- Time and date picker
- Smart Pager
- List Builder
- Several new design patterns

How you can help:
- Get involved in user research
- Code, designs, testing for new components
- Write a design pattern
## Smart Pager

**Fluid**
Designing software that works - for everyone

Start typing a name...  

<table>
<thead>
<tr>
<th>Name</th>
<th>User ID</th>
<th>Email Address</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campeau, Patrick</td>
<td>15234314</td>
<td><a href="mailto:pcampe@berkeley.edu">pcampe@berkeley.edu</a></td>
<td>Student</td>
</tr>
<tr>
<td>Capovilla, Megan</td>
<td>19269508</td>
<td><a href="mailto:megan@berkeley.edu">megan@berkeley.edu</a></td>
<td>Student</td>
</tr>
<tr>
<td>Catania, Natalie</td>
<td>19435570</td>
<td><a href="mailto:ncatania@berkeley.edu">ncatania@berkeley.edu</a></td>
<td>Student</td>
</tr>
<tr>
<td>Chalhoub, George</td>
<td>19206726</td>
<td><a href="mailto:gchalhoub@berkeley.edu">gchalhoub@berkeley.edu</a></td>
<td>Student</td>
</tr>
<tr>
<td>Chan, Julie</td>
<td>19206726</td>
<td><a href="mailto:jchan@berkeley.edu">jchan@berkeley.edu</a></td>
<td>Instructor</td>
</tr>
<tr>
<td>Choi, Daniel</td>
<td>16984247</td>
<td><a href="mailto:dchoi@berkeley.edu">dchoi@berkeley.edu</a></td>
<td>Student</td>
</tr>
<tr>
<td>Clarkson, Ian</td>
<td>19286942</td>
<td><a href="mailto:iclarkson@berkeley.edu">iclarkson@berkeley.edu</a></td>
<td>Student</td>
</tr>
<tr>
<td>Cole, Paul</td>
<td>19446940</td>
<td><a href="mailto:pcole@berkeley.edu">pcole@berkeley.edu</a></td>
<td>Student</td>
</tr>
<tr>
<td>Colgate, Abbey</td>
<td>14644786</td>
<td><a href="mailto:acolgate@berkeley.edu">acolgate@berkeley.edu</a></td>
<td>Instructor</td>
</tr>
<tr>
<td>Colville, Tony</td>
<td>19283731</td>
<td><a href="mailto:tcolville@berkeley.edu">tcolville@berkeley.edu</a></td>
<td>Student</td>
</tr>
</tbody>
</table>
Time/Date Picker

February 21, 2008
Open on: 2/21/2008 11:30 AM

Due on: 2/22/2008 7:30 PM

February 22, 2008

Click

February

2008

21 22 23

20 21

15 16

7 8 9

10 11 12 13 14 15 16

17 18 19 20 21 22 23

24 25 26 27 28 29 30 31

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

User Experience

Fluid
Designing software that works – for everyone
Beyond 0.4

- Accessibility adaptation
- User preferences editor
- Accessibility design patterns
- A suite of new components
- A new wave of UX walkthroughs, targeted at file management

How you can help:
- Accessibility testing
- Help with UX walkthroughs
- Component and framework development help
How Fluid Can Help You

• Use Fluid components in your applications
  – Accessibility & great design for free
• Build new UIs using Fluid’s techniques and plugins
  – Robust strategies for UI development
• UX Walkthroughs
  – Assess and improve your user experience
• Open Source Design Patterns
  – Advice on common UI design considerations
• High Education User Profiles
  – Understand your audience
Getting Involved
How You Can Help

• Join our mailing lists
• Share code
• Help with design effort
  – UX Walkthroughs are fun and easy
  – Contextual inquiry
  – Component design teams
• Use and extend Fluid components in your tools
• Write your own Fluid components
• User testing
• Share design patterns
Join in…

- Fluid Project Web Site:  
  [http://fluidproject.org](http://fluidproject.org)
- Our wiki:  
  [http://wiki.fluidproject.org](http://wiki.fluidproject.org)
- Our source code:  
  [https://source.fluidproject.org/svn](https://source.fluidproject.org/svn)
- Our mailing lists:  
  fluid-work@ for community collaboration
  fluid-talk@ for anything you’re interested in