Criteria for Selecting a DHTML Toolkit

Criteria for Selecting a Supported DHTML Toolkit for Fluid

Overview and Background

Fluid is currently evaluating the range of currently available DHTML/JavaScript toolkits on which to build our framework and components. There are a variety of different libraries at varying levels of maturity and with very different feature sets, including:

- Dojo
- jQuery
- YUI
- MochiKit
- Prototype and Script.aculo.us
- Ext JS

While we had originally been hoping to select a toolkit much earlier in the project, it became clear that the level of expertise and community awareness in these projects is significant. Given that this is a fundamental architectural decision, our goal now is to come to a community-wide consensus on a recommended toolkit at the Fall 2007 Fluid Summit in September.

What about supporting multiple toolkits? Or building Fluid to work with any toolkit?

Here is my (Unknown User (colin.clark@utoronto.ca)) thinking on the issue. We need to be very careful about mixing and matching different DHTML toolkits, particularly when it comes to user-facing widgets and interactions. The very real risk is that there are subtle differences in the behaviour of the toolkits that will totally destroy the effect of interaction consistency that Fluid is striving to offer. This is a bad thing.

So while I can imagine using aspects of several toolkits for low-level utilities and libraries, the reality in my mind is that we're going to have to be very careful about choosing a single toolkit for any user-facing widgets that we use. At the moment, Dojo is way ahead of the rest in terms of accessibility. Implementing the Reorderer in several toolkits, which we will take a stab at during the Fluid summit, will provide us with some common ground on which to compare them.

Criteria

Please feel free to add your own suggestions for toolkit selection criteria.

- Accessibility support, existing and/or planned, including at a minimum:
  - ARIA
  - Keyboard control
  - High contrast support
- Ease of debugging
- Cross-browser support (ideally IE 6 & 7, Firefox 1.5 and 2, Safari on 10.4, Opera)
- Solid library for DOM manipulation
- Plays nice in a portal and with other toolkits
  - No global pollution of the global namespace
  - Does not override language-provided objects in unpredictable ways
  - Clear initialization lifecycle such that it can be initialized in mid-page render
  - Works within mutiple placements of the same portlet)
  - Gentle use of coarse-grained event handlers e.g. doesn't use onload()
- Sufficient event abstraction to avoid having to deal with browser event handling inconsistencies
- Basic affordances for security
- Clear extension points
- Skinability: the ability to easily change the default appearance
- Strong community support and a clear roadmap for improvements
- License compatibility with ECL and BSD
- Plays nice with RSF and Spring MVC at a *bare minimum*. Client-side toolkits should play nice with all server-side frameworks, really.