

Peter Armstrong
Foreword by Stuart Eccles

FLEXIBLE RAILS

FLEX 3 ON RAILS 2

 MANNING



Flexible Rails

xml hell

Demo Site

- * <http://pomodo.int.uscm.org/>
- * Username: ludwig
- * Password: fo000

What is it?

- * Cold Fusion?
- * XML!
- * Object Oriented
- * Client-Side

Pros

- * **UI Smarts** (good defaults)
- * **Browser/OS agnostic** (for real this time)
- * **Designer Friendly** (non-programmer types)
- * **Gradients!**

Cons

- * **Expensive IDE** (\$250, thanks Adobe)
- * **My hands hurt** (xml)
- * **Client Needs Flash**

Let's see some code!

What if I actually like HTML, CSS, and JavaScript? David Jun 26



[87 comments](#) Latest by Kris

It seems to be Spring season for rich-interface technologies. Most trying to blossom with a story about how they'll rescue developers from the perils of web programming and its dirty tech of HTML, CSS, and JavaScript. These stories are told with a shadow assumption that the only reason developers put up with this trinity of web specs is because of what they get back in form of ubiquitous distribution.

That assumption then leads to the fallacy that if only someone could come along and give us a competitive distribution story using more "advanced" and "rich" interface technology, they'd surely be golden. That all web developers are waiting on is someone to save them from the browser mess and deliver them the clean desktop-development experience of yester-century.

In the immortal words of Eric Cartman: *Bullcrap.*

As a web developer, I'd like to confess my deep appreciation of the restricted canvas that we get from the basics of the web. We've entered an era where the browsers are good enough, compatible enough, and, most importantly, our understanding of how to use what we got has been raised to a level where things are finally looking pretty good.

I actually find the development experience between a modern web-application framework, Firebug, and current JavaScript libraries more than just bearable, I find it downright pleasant. Even more so because it's born out of the pragmatism of not needing to be perfect. It has evolved over a decade of experimentation

Peter Armstrong 27 Jun 07

Hi David,

While you are a much better coder and a lot richer than I am, you are 100% wrong about rich Internet applications.

This is even more misguided than your "You're not on a fucking plane" post.

This debate is not "Rails vs. rich Internet applications", by the way. I am a proponent of using Rails and Flex together.

See the MIT licensed code samples from my book Flexible Rails for an example of how Flex and Rails can be used together. (Yes, this is blatant and shameless self-promotion, but it's **extremely** relevant to this discussion so I'll mention it anyway: I am trying to show the world how **YOUR** framework can be a key part of the next generation of rich Internet applications—applications which go way beyond the amazing tricks that gurus like yourselves and Thomas Fuchs have been able to do with JavaScript, HTML and CSS .

If someone like me can put together a decent, fairly well-designed application with hardly any code (or time or funding), imagine what someone with your (and your colleagues') talents and resources could do with the technology! I bet it would blow away what you have done so far, which is already very impressive and successful...

Best regards,
Peter Armstrong

Chris Kampmeier 27 Jun 07

@Peter: bringing Rails (and the size of David's wallet) into this discussion is a red herring. This conversation is about the technologies in the title of the post and the new stuff out there from Microsoft and Adobe.

I think the point is that 1) Flash/Flex/Apollo haven't shown themselves to be a credible replacement for our current standards for markup, styling, and interaction; and 2) even if they had, a lot of us find the design of the current standards to be really beautiful in unexpected ways, and we don't need Microsoft and Adobe to reinvent the web in their image for us.

When I think about the environment and time when these standards were conceived, it displays an incredible amount of foresight about how the web would work on behalf of the big thinkers of those days. We owe a lot to the designers at the W3C, browser makers, and elsewhere. There's a lot about the makeup of the DOM, the box model, markup semantics, etc. that has stood the test of time, evolved beautifully, and encouraged the development of the web in unexpected ways.

You'd think it would be the other way around—that these 10+ year-old specs would have stifled us out of business by now. On the contrary, the constraints they introduced were needed. I'm tremendously excited about where we go from here, but it's not because of these productized, closed-development platforms. It's because of the evolutionary, open-source, community-based process that got us here in the first place.

Peter Armstrong 27 Jun 07

@Chris: You're right about the wallet thing being a red herring. My bad. My point about Rails though, is **not** a red herring: the thing that to me is unfortunate about this is that when DHH posts stuff like this it portrays the technology landscape as a "battle" between HTML + CSS + JavaScript on one side vs. Flex (or Silverlight, which I don't care about) on the other.

Now, what do you think David is advocating as the best way to work with HTML + CSS + JavaScript? Rails. So, **implicitly, DHH is framing the debate as Rails vs. Flex.** And since DHH 's opinion obviously carries a lot of weight in the Rails community, the underlying assumption many people in the Rails community will get from reading posts like this is that Flex is somehow the enemy of Rails.

If DHH posted "Hey everybody, if there is something you can't easily do in JavaScript, try using Flex with Rails—they work great together" here, the attitude of many people in the Rails community would be different. The point I am trying to make is that it is **not a battle** or a pissing match. **I really like Rails, and I really like Flex, and I like using them together.**

DHH 27 Jun 07

Peter, with enough extrapolation you can probably also end up with a conclusion that I hate babies and puppies. I'm eagerly awaiting that one ;).

Seriously. Read the entry again without the paranoia hat on and the message might be closer to "me likes html, me needs no rich-interface savior". There's no Rails hates Flex (or babies or puppies) in there.

Peter Armstrong 27 Jun 07

@DHH - I'm glad you're speaking of Flex in the same breath as "babies or puppies", not "yester-century" and "Bullcrap" ;)

I am a huge proponent of Rails, and think that it has a lot to offer the Flex community, and vice-versa. (My first presentation was to Flash and Flex developers advocating Rails, not to Rails developers advocating Flex; that was my second one.) The REST stuff that was added to Rails makes the integration between Flex and Rails even easier—now if only there was an easy way to turn off dasherize globally by default without monkeypatching :) [It doesn't work with E4X . Yes, I should submit a patch or write a plugin, I know...]

Finally, RubyAMF is now MIT licensed, so there is lots of promise in that way of integrating Flex and Rails (or just Ruby) as well...

Cheers,
Peter

Who's Using It?

Notable sites using Flex

Notable sites using Flex include:

- [Pikeo](#) online photo sharing
- [Amsterdam-Schiphol Airport](#) [1] [\[2\]](#)
- [Sherwin-Williams](#) Color Visualizer
- [Sony Ericsson](#) Product Catalogue
- [Yahoo! Messenger](#) for the Web [1] [\[2\]](#)

There is a showcase of Flex applications

* see flex.org

The screenshot displays the 'FLEX Showcase' website interface. At the top, there is a navigation bar with 'FLEX Showcase', 'Featured', 'Browse', 'Tags', a search box, and a 'Login / Signup' button. Below the navigation bar, there are filters for 'Categories', 'Show All', and 'Sort By' (Latest, Popular, A-Z). The main content area shows '174 Results' and a grid of 16 application thumbnails, each with a title and a small preview image. The thumbnails are arranged in a 4x4 grid. At the bottom of the page, there is a footer with 'Built by: Teknision & Adobe', 'Powered by: DRUPAL', and 'Flex is a trademark of Adobe Systems Inc.'

Application Name	Thumbnail Description
Volkswagen UK - used car locator	Screenshot of a car search interface with a grid of car listings.
Scrapblog	Thumbnail showing a blog layout with photos and text.
Buzzword	Thumbnail showing a news or article page with a dark background.
flauntR - photoeditR	Thumbnail showing a photo editing interface with a landscape image.
Activa Live Chat	Thumbnail showing a live chat window overlaid on a website.
TagGraph	Thumbnail showing a network graph of tags and their relationships.
Picnik	Thumbnail showing a scenic landscape with mountains and a blue sky.
Anywhere.FM	Thumbnail showing a music player interface with a dark theme.
INM eLibrary	Thumbnail showing a library or document management interface.
Finetune Desktop	Thumbnail showing a desktop environment with various application windows.
Fauxto.com	Thumbnail showing a desktop environment with a sunset background.
Appointment System	Thumbnail showing a calendar or appointment scheduling interface.
Mindomo.com	Thumbnail showing a mind mapping or diagramming application.
Mini USA	Thumbnail showing a desktop environment with a blue and white theme.
Photo Uploader	Thumbnail showing a simple web interface for uploading photos.
Matchmine	Thumbnail showing a social networking or dating application interface.

When Should I use it?

- * Small component of a larger app
- * Heavy UI app
- * Corporate setting?

Go Deeper

- * Cairngorm
- * Rails on AIR
- * RubyAMF?

What is Cairngorm?

- * The Cairngorm Microarchitecture is a lightweight yet prescriptive framework for rich Internet application (RIA) development.
- * Cairngorm is an implementation of design patterns that the consultants at Adobe Consulting have successfully taken from enterprise software development (with technologies including J2EE and .NET) and applied rich Internet application development using Adobe Flex.

What is Adobe Air?

- * Adobe Integrated Runtime
- * Cross-operating system runtime that allows developers to use their existing web development skills to build and deploy rich Internet applications to the desktop.
- * Adobe AIR is targeted at allowing web developers to build and deploy desktop applications using web technologies.

RubyAMF

- * RubyAMF is a flash remoting gateway that allows a Rails backend to communicate with a Flex frontend. The marriage allows rapid development of rich internet applications.