HTML5 and the Future of Adobe Flash

HTML5 poses a threat to Adobe Flash (and to other rich Internet application [RIA] plug-ins). However, the dynamics are complex, and the time frames longer than one might expect.

Key Findings

- Statements about HTML5’s impact on Flash are, in most cases, applicable to other plug-in RIA technologies, such as Silverlight and client-side Java.

- Concerns about Adobe Flash, with regard to performance, robustness and security, should take into account the context of competitive platforms, which encounter many of the same issues.

- Millions of websites use Flash, and some high-growth, high-traffic sites depend on Flash in a business-critical context.

- HTML5 will become the mainstream of the Web during the next decade.

- HTML5 is a potential threat to the continued adoption of plug-in-based RIA approaches, including Flash.

- Adobe is responding to this threat, and its continued success depends on how well it executes.

Recommendations

- Enterprises should try to avoid becoming dependent on any one browser or client-side technology, especially for Internet-facing applications.

- Enterprise developers should “design for standards” and not browsers or runtimes.

- Developers should favor the lightest-weight technology that will meet their requirements.

- Enterprise designers should specify user interaction patterns and workflows in a technology-independent manner, and only in later phases of the project should they map these interaction sequences to specific technologies.
• Architects should consider hybrid approaches (HTML plus HTML5 plus Flash or Silverlight) as alternatives to the monolithic use of client-side technology; in any event, architects must ensure “graceful degradation” that preserves the usefulness of the application in scenarios when higher-order RIA technology is unavailable.

• Before purchasing or committing to a new user interface (UI) technology or platform, enterprises should first invest in a user-centered design process based on objective data about user behavior.

ANALYSIS

Recently, Gartner has received many inquiries from end-user organizations seeking to make strategic platform decisions regarding HTML5 and its impact on the future of Flash. The trigger for this interest was the announcement by Apple in January 2010 of its iPad tablet computer. This announcement elicited many reactions from various industry quarters. Conspicuously absent from the iPad platform was Adobe Flash. This absence dashed the hopes of Flash advocates and continues the precedent established by the iPhone, a platform from which Flash was also excluded. The fierce debate that emerged in the wake of the announcement was in regard to the future of Adobe Flash, and whether the future Web and mobile RIAs will be built with HTML5, rather than Flash or other “heavy RIA” alternatives.

Opposing perspectives were voiced in blogs and online forums. On the one hand are Flash opponents and skeptics, who view the technology as being past its prime. These opponents hold a negative view, not just of the technology, but also of some Flash-based content (i.e., interactive banner ads and splash pages). On the other side are pro-Flash advocates who see genuine value in a technology platform that supports rich interactions, has ubiquitous penetration, maintains a solid track record, and goes beyond the current and near-term limitations in HTML5.

Regardless of one’s advocacy for Flash or HTML5, here are key aspects of the situation.

The future of plug-in-based RIA is not just about Flash. Any discussion of Flash should also (depending on the level of detail) bring into the discussion other plug-in-based RIAs – for example, Microsoft Silverlight and client-side Java. Although these approaches differ with regard to vendor portfolio, market penetration, installed base, etc., many of the issues that impact Flash also impact these other approaches. For example, none runs on the iPhone or iPad today. Although Flash has broader penetration and a more solid track record than alternatives, there are no screeches published in blogs about the imminent “death of Silverlight” due to its absence on iPhone and iPad devices.

There is a message in the media that Flash is a subpar technology. The concerns that have appeared in the media for not allowing Flash on the iPhone and iPad is that Flash is unstable, has low performance, is unsecure and drains battery life. However, there is little evidence that Flash differs from the other rich, complex technology available from other vendors. Tests published in February 2010 by the Streaming Learning Center show that Flash is roughly equivalent to HTML5 in terms of CPU utilization and performance (i.e., better in some configurations and worse in others).

Despite the message, the reality is that Flash is ubiquitous and has a solid track record. Flash is found in 97% of Internet-connected devices, and has been in use in one form or another since 1996. Gartner estimates that Flash is used in 70% of the top 100 websites (although usually in a superficial form). There are millions of monthly active users of Flash-based video (e.g., YouTube and Yahoo), Flash-based gaming sites and game portals (such as Zynga, Pogo, Kongregate and Omegpop), social-networking sites (Club Penguin and IMVU) and video chat sites (Stickam, WoWMe and Chatroulette). Zynga alone claims 230 million unique monthly users, and enjoys annual revenue estimated at $400-plus million and a private-company valuation that could exceed $2 billion. If Flash has unacceptable levels of performance or efficiency, these agile, fast-moving companies (or their competitors) would quickly shift to alternative platforms, and, to date, they have not embarked on such a process. Some small players, such as NPR and Brightcove, have announced a shift away from Flash. However, the driver behind the decision was not any of the usual rationales (performance, stability, etc). Instead, the stated driver was Flash’s absence from the iPhone. Also, with regard to security, the track record of Flash has been better than average, when compared with other widely used Internet technologies, such as Internet Explorer. In terms of adoption, Flash has succeeded in the market, while contemporaneous RIA alternatives faded (client-side Java, ActiveX, the initial release of Windows Presentation Foundation [WPF], etc.)

Any large powerful application will consume CPU and battery power, whether written in Flash, Silverlight or HTML5. Simple applications consume minimal resources, and most HTML5 and Flash applications are simple. Complex applications with high interactivity and large amounts of computation will consume CPU and battery no matter what technology they are implemented in. Some applications may be better than others in this regard – perhaps even 20% or 30% better – but such differences are incremental, not game-changers.

The “quality of experience” theme regarding Flash’s absence from the iPhone is inconsistent. The iPhone system has its own security issues, and many third-party applications suffer from performance problems. Regarding the topic of poor quality and performance, many thousands of applications in the App
Apple’s divergence from Adobe is not a recent phenomenon. The Preview application used to view PDF files is Apple’s long-standing replacement for Adobe Reader on OS X, with code that dates back to the days of Display PostScript and the NeXT Computer. Based on this 15-year history, it appears that the decision not to allow Flash on the iPhone is unlikely to change anytime soon. Regardless of history, Apple’s resistance to Flash may have more to do with RIA technology’s potential to disrupt Apple’s monetization mechanism – the App Store – than any real or perceived shortcomings in Adobe’s technology. This supposition is supported by Apple’s reported decision on 8 April 2010 to modify the developer agreement for iPhone SDK 4.0, to mandate that applications for iPhone can only be written in a small number of languages (Objective-C, C, C++ and JavaScript), and that the use of any intermediary translation layer is prohibited.

Flash has the potential to threaten the iPhone platform, given that Flash is a cross-platform presentation layer on mobile and desktop machines. This is true also for Silverlight and cross-platform mobile development tools such as Appcelerator Titanium and Unity. Barriers to Flash on the iPhone/iPad will linger as long as the presence of a “closed” App Store provides a competitive advantage for Apple. The question is what impact this resistance will have on Adobe, and to what extent Adobe can work around these limitations (as it has tried to do with its Flash-to-iPhone cross-compiler).

HTML5 is the future of Web, for simple interactivity, including charting, some limited 3D vector graphics, image transforms, video and audio. It is possible that 90% to 95% of an average enterprise’s needs could be met by HTML5. Today, there are only a few classes of corporate applications that would gain significant benefit from Flash, Silverlight or Java over what is available in HTML5 or even in Ajax. Both HTML5 and “heavy RIA” alternatives expand the palette of available design alternatives beyond the ability of most enterprise Web developers.

HTML5 is the future of the Web, but the complete rollout of that future could take a very long time. The total set of HTML5 standards is a large and complex collection, and current projections by the people working on the specifications are for all parts to be finished in 2022, some 18 years after the process began (in 2004).

Nevertheless, the use of HTML5 among top websites is already under way. Web designers are not waiting until 2022 to make use of a working subset of HTML5. For example, YouTube and Vimeo have already rolled out use of the video element in HTML5. Other websites and applications are using Canvas and offline storage. A de facto working subset of HTML5 is already starting to appear, both on the “desktop Web,” as well as on the mobile Web. Although there are tens of thousands of Flash-based games, there are millions of websites that use Flash in a simple manner (for basic interactive content such as banner ads or splash pages). One could argue that much of this content is of low value (users get “banner blindness,” and are habituated to skip useless introduction or splash pages). Regardless of its value, much of this simple interactive content could be replaced by today’s HTML5 working subset.

The working subset of HTML5 has nowhere near the power of Flash. Many advanced effects are only available in Flash, Silverlight or Java. For example, Google, which is driving HTML5, relies on Flash in Google Maps (for the Street View) and in Gmail (for the multiple-file upload capability). Tens of thousands of Flash games on the Web would be difficult to do (in a performant way) with HTML5. Microsoft Silverlight’s Deep Zoom feature would be difficult, if not impossible, to replicate using only HTML5.

It’s not just about features, but also about deployed infrastructure. This benefits Flash. A pragmatic perspective should look at the numerous tools, ad engines, business processes, infrastructures and platforms that support and/or enable Flash-based advertising. This aggregate mass will take a long time to shift to an alternative, no matter how good that alternative may be, due to the sheer inertia of large-scale systems that are operationally functional.

A portion of the Web requires richer interaction. Some applications might require extensive offline processing, direct manipulation of graphics, real-time notifications and alerts, high-speed binary communication protocols, tight integration with local devices, etc. In these scenarios, you might need to use Flash, Silverlight or Java (the exact choice would depend on your context, such as your development team, IT landscape, vendor relationships, etc.).

The choice among these technologies is not “all or none.” One pragmatic approach that can deliver the best of both worlds is a hybrid strategy – one that incorporates “islands of RIA” or within the sea of HTML. In the near term, this requires a plug-in-based approach. Over the long term (five or 10 years), HTML5 may fit the bill and provide a unified standards-based approach.

The informal anti-Microsoft grouping of Google, Adobe and Apple of times past has splintered. This was only a loose federation to begin with (“the enemy of my enemy is my friend”). Now there are multiway tensions and collision courses (Apple versus Google in mobile, browser and media space; Apple versus Adobe in browser/plug-in; and of course Microsoft versus each one of these). One emerging dynamic is a closer alliance between Google and Adobe, with a recent announcement that Adobe would bundle the Chrome browser in the Flash distribution.

HTML5 poses a strategic threat to Adobe, and also poses a threat to Microsoft. Adobe is the more impacted of the two in the short term, because Microsoft has solid territory in enterprise IT environment (for example, Silverlight is leveraging the success of the SharePoint portal and the soon-to-be-launched Windows Phone platform). By contrast, the consumer Web (especially the smaller, more-agile Web properties) can change direction and platform more quickly. Flash is used in 70% of high-traffic websites, but some of these uses are surface-level and easily removed. (However, as mentioned earlier, it will be harder to turn large-scale ad engine operations around them).
Adobe sees the writing on the wall and is responding. Adobe has undertaken various initiatives, from the Flash library for iPhone that allows compilation and embedding into native iPhone applications to Flash 10.1, which is a more efficient implementation for mobile CPUs that needs to conserve battery life, to improved security procedures and a better development process. Adobe’s future depends on how well and how fast it executes. Adobe can improve its opportunities if it can successfully educate a larger market about the importance of user-centric design processes and clearly articulate a value proposition that makes Flash and Omniture the centerpiece of a “foolproof” set of enterprise Web design, development and continuous-improvement tools.

The average enterprise will continue to make ineffective use of any and all available UI technologies. The root problem is not lack of powerful UI technology. Instead, the root causes for a suboptimal user experience consist of lack of appropriate process and governance, and lack of a genuine commitment to a quality user experience. Such a commitment would lead organizations to adopt a user-centered, usability-oriented development process. Rather than taking these steps, we see a lot of projects that are “stakeholder-driven” (i.e., driven by internal politics). Very few organizations center development around user needs by relying on objectively measured data about user behavior. Most enterprises don’t seem to care enough about the user experience to change their habits (in terms of processes that are developer-driven, vendor-driven and stakeholder-driven, rather than user-driven). The principles of creating effective user experiences are well-known among successful external-facing e-commerce or consumer sites, such as Amazon, eBay, Expedia or Facebook. Unfortunately, it will likely be a long time before these principles become part of the average enterprise skill set.