



Kevin Lynch on Adobe's Plans for a New Generation of Software

Published : July 20, 2006 in [Knowledge@Wharton](#)

Having completed the acquisition of Macromedia in December 2005, Adobe Systems now controls two of the de facto standards for electronic content -- the Portable Document Format (PDF) for electronic documents and the Flash SWF format used for web-based animations and interactive content. Adobe believes that combining these technologies with a unified set of software development tools and programming interfaces ("APIs") -- creating what Adobe terms its "Engagement Platform" -- will uniquely position the company to deliver a new generation of software development tools.



This is a single/personal use copy of Knowledge@Wharton. For multiple copies, custom reprints, e-prints, posters or plaques, please contact PARS International: reprints@parsintl.com
P. (212) 221-9595 x407.

As he explained when [Knowledge@Wharton spoke with Adobe CEO Bruce Chizen](#) following the completion of the acquisition, Macromedia's Flash technology was the key strategic asset that motivated Adobe's acquisition decision. In addition to integrating Flash into Adobe's other technologies, Adobe also has to convince developers that Flash is more than a tool for creating simple web animations and can serve as the foundation for developing full-fledged web applications.

But Adobe's ambitions are greater than merely integrating its developer tools and expanding the use of Flash. With its forthcoming technology, code-named "Apollo," Adobe hopes to lay the foundation for an entirely new category of software applications.

Currently, programmers can create traditional "native" software applications that take advantage of the full capabilities of the computer on which they run but are locked to a particular operating system. Or they can build light-weight web applications that run in the web browser -- and hence work on any operating system -- but have limited capabilities.

With Apollo, Adobe hopes to provide a third alternative that provides the same cross-platform capabilities of a web browser, but with a richer set of features -- such as the ability to work when the computer is not connected to the Internet.

These moves put Adobe on a collision course with companies like Microsoft which have a significant investment in operating-system native applications. And Microsoft has announced products that compete with Adobe on almost every front -- from its Windows Presentation Foundation development environment that stands in contrast to Apollo, to new file formats that compete with Adobe's Flash and PDF.

This battle for the next-generation development environment may be the biggest gamble in Adobe's history. Before his [panel discussion on "The Future of the Desktop" at Supernova 2006](#), Knowledge@Wharton met with Kevin Lynch, Adobe's chief software architect and senior vice president of Adobe's Platform Business Unit, to discuss Adobe's vision for the future of software applications on the web, the desktop, and mobile handheld devices. An edited version of the conversation follows.

Knowledge@Wharton: Adobe has talked a great deal lately about what it terms the "Adobe Engagement Platform." That's a very clever marketing term, but what does it really mean?

Kevin Lynch: The way I like to talk about it is [in terms of] how we brought Adobe and Macromedia together. I was on the integration team bringing our companies together, and we were looking at the missions of the two companies, how we were going to unite those, and what value we were bringing to end users.

If you look at the mission of Macromedia at the time, it was about the experience. Our slogan was "experience matters." And at Adobe the focus was on having the best communication, whether via print, or video, or whatever means you're communicating.

And in these brainstorming meetings we got to the point [where we asked], "Why are people doing [these activities]?" instead of "What are people doing?" And why people are doing them, we believe, is that they want to have a better engagement with someone, they want to create something that is engaging. And if you're actually engaged with something, that creates a closer relationship and a better understanding.

And that is the core of what we're trying to enable people to do -- engage effectively. That's what bringing our software together is about. All of our tools, our servers, our client [software], and [our developer] frameworks -- all that stuff is to help people engage better.

And so we call that the "Engagement Platform." It's basically the collection of software you can use to create these experiences and engage people.

Knowledge@Wharton: So rather than a single product, it's the set of products across the combined company?

Lynch: Right. You can't go to the store and buy "the Engagement Platform."

Now, for developers, the word "platform" implies a stack of technologies that you can program to, a set of APIs [application programming interfaces], and the Engagement Platform does contain those things. The client [software] run-time [environments], for example, have a consistent programming language, which is the ECMAScript standard. We call it ActionScript in Flash, it's called JavaScript in HTML -- but it's the same language. And the set of APIs in [software such as] Flex, for example, provides that technology stack which is very much the traditional notion of a platform.

We're bringing the client programming models together and we're bringing the servers together. [For example,] we're rationalizing the LiveCycle [suite of server products] with Flex data services and Flash Media Server. You'll see those things being more aligned over time.

And the same on the tool side -- there is more and more integration between the interfaces of Photoshop and DreamWeaver.

Knowledge@Wharton: The other thing you've been talking about is something code-named "Apollo." What is Apollo?

Lynch: That's a new client [software application]. Right now we have the Flash Player and [Adobe] Reader widely distributed to run applications in [web] browsers and to view electronic documents. The [web] browser is really designed for documents, and -- with a lot of the innovation that's happening about Web 2.0 right now -- people are pushing the boundaries of what you do with that, which is exciting.

We thought it is a great opportunity to provide a runtime [environment] for applications outside the browser. So you can take your web app that you use in the browser and put it on your desktop. That means you can use that application when you're not even connected to the Internet. If you're using an

application like [Google's] Gmail or your financial [software], you could still run that application and look at your information [offline]. Those applications could be first-class citizens on your computer -- so they can have an icon you double-click to launch the application, they appear in your Start Menu like other desktop applications, and you can remove them with Add/Remove programs in Windows. So it's just like a desktop app, but is built with web technologies.

Knowledge@Wharton: And because it's built with web technologies, is it operating system agnostic? Would this application run under Windows, Macintosh, Linux?

Lynch: Yes, that's right. Consistently.

Knowledge@Wharton: Is Adobe thinking about this exclusively as a developer platform or will we see products from Adobe that are Apollo applications?

Lynch: I think most of the applications that will be run on Apollo will be developed by other folks. Though we may make some at Adobe.

[Adobe's] Breeze [web conferencing software] is an example of an application that Adobe makes that runs on the Engagement Platform; it runs on the Flash Player. So, we'll make some of those ourselves. But I think -- just like the web today -- there will be thousands and thousands of different applications built by designers and developers all over the world, [both] inside companies and individually.

Knowledge@Wharton: As you position this as an operating-system independent platform, doesn't this put you in collision with companies that have a strong investment in having people code to the operating system?

Lynch: This is about web applications. There are also, of course, native code applications like [Microsoft] Office. And [Apple Macintosh's] OS X has a lot of great native applications. I think that those would continue to have great value, and there's a lot of relevance to those. People use Office all the time.

But there's a new class of applications that are emerging that are web apps. I think about this as a new medium. It's not necessarily displacing an existing medium. When television came out there was concern it would kill radio. Of course, radio is healthy and continuing to go. I think this is just another technique people can use to deploy their work. It doesn't eliminate the other options that people have for building apps.

Knowledge@Wharton: Does that mean that there's a "topping off" point [for Apollo applications], that there are certain apps that would be appropriate for the Apollo environment and there are other applications that would only be appropriate as native operating system applications?

Lynch: Yes, I think you'll see that spectrum, absolutely. You'll see some experiences that will stay just within the browser; some which will get more ambitious and want to run as a desktop app but built with web technologies; and then you will continue to have the full spectrum of [applications that] need to take advantage of specific hardware on my machine, or high acceleration 3D graphics, or other more extreme technologies. Multiplayer games like World of Warcraft will continue to be written in native code, because you need [an application like] that super-optimized for the particular machine you're running on.

But for a lot of applications going to those lengths isn't as necessary. You could build a web app that would show you your financial numbers just fine. So, I think it will be a spectrum.

Knowledge@Wharton: What kind of applications do you think would be at the high end of what an

Apollo application could be? Are there examples you can give us?

Lynch: Sure. Apollo is new and, obviously, is in development right now. We're planning to have it be available for developers to start playing with later this year.

But the kinds of applications that I expect we'll see are ones that you tend to use more than others. There are some applications you use on the web that are of a transient nature -- you use them infrequently. But there are others that people use daily or several times every day -- like your email application, or checking your stock portfolio, or communicating with people, or analyzing the current sales numbers for a project you're working on.

And those applications which you have a closer relationship with and you have a higher frequency of interaction with, those are the ones that, I think, have an icon on your desktop and you carry around with you and you want to look at if you're not connected [to the Internet] -- they're that important to you.

Like being able to write an email message when you're offline. That's important. But you might [wait] to read a news story until you get into a browser again.

So it varies by the information.

Knowledge@Wharton: You don't envision a day when the operating system is -- to use the term Bruce Chizen was quoted as recently saying [in the *San Jose Mercury News*] -- "irrelevant"?

Lynch: [Laughs.] I think a little bit of liberty was taken with that. Certainly the operating system is not irrelevant. It's key. It's what makes all of our machines operate. And I think it's got a healthy future going forward.

But, at the same, there are new opportunities that will emerge. And new innovative techniques are arising. The web is where a lot of the innovation is happening. So that's where we're focused right now.

But operating systems are a healthy space. There's a lot of money to be made in operating systems!

Knowledge@Wharton: While we're on this topic -- Microsoft, in specific, is competing with Adobe on several fronts.

Lynch: Yes.

Knowledge@Wharton: On the tools side you have some competitive products. But at a more strategic level, Microsoft's Windows Presentation Foundation could be a competitive platform. How would you compare Apollo and the Engagement Platform with Windows Presentation Foundation?

Kevin Lynch: In general, Adobe and Microsoft are partners as well as, in some areas, competing. So you can think of it as "coopetition." We're one of the largest Windows software developers in the world. We build on Windows. Vista is coming out, and we'll be taking advantage of Vista functionality.

In terms of WPF and what we're doing with Apollo, WPF is a framework for building Windows applications. So, if you're running on Windows, it could be very appealing. Rather than, for example, using [Microsoft's] Visual Basic, you would use WPF. I think they've done a very nice job on that.

Now, there's an aspect of WPF called WPF/E [Windows Presentation Foundation/Everywhere], which is

[Microsoft's] intention to bring WPF to other platforms. It is not available yet. And, historically, Microsoft has not done a good job at crossing platforms with technologies. Or, when they have, they have not sustained that effort.

For example, [Microsoft's] Internet Explorer used to run on Macintosh. They don't maintain it anymore, and it's been canceled. Windows Media used to run on [the Apple Macintosh's] OS X. It's also been canceled.

So, occasionally [Microsoft will have] footholds into other platforms, but I think it's hard for people to really count on that.

And, in a lot of ways, it's not really in Microsoft's interest. They have a Windows franchise and they sure would like to sell a lot of operating systems. So, I think that's a hard balance for them.

Our dynamic is very different. Our core effort is to make sure things *do* run consistently across operating systems and devices. And our runtimes are distributed for free. So, our business motivation is to get those as widely deployed as possible -- and not make you buy a new computer; you can use your current computer. Our goal is to be even more cross-Windows than Microsoft is. The Flash player will be more consistent across Windows than, say, WPF. We'll run on Windows 98. WPF will not do that.

We're actually motivated to have that level of compatibility, and that is fairly unique for us versus the approach that Microsoft is taking.

Of course, we have other products that we sell given that ecosystem -- tools and servers and frameworks. We work to make the best ones in the world, so we hope people will [buy them]. And they have been.

Knowledge@Wharton: We've seen a big explosion of Flash video lately. You've jumped to the number two spot in terms of videos online -- behind Windows Media format and ahead of QuickTime and Real [Media]. How did that happen so fast?

Lynch: The Flash Player is actually the number one player for video today -- it has more reach than any of the other video players. It's more widely distributed than Windows Media or QuickTime or Real. That is part of how it's gotten so popular in terms of content usage now.

Before, you had to quiz people about which player they wanted to use to view the video or what bandwidth they had. And after going through that quiz you could see the video. Now that the Flash Player is so reliably distributed, you can just assume it's on your machine and it has video capability in it. You can present a better video experience to the end user.

That's why you're seeing the radical shift for people who deploy video like ABC, which is playing "Desperate Housewives" and "Lost" [online in Flash video]. And it's all ABC branding, they don't have to quiz you before you see [the video]. It just works.

In addition you can have interactivity -- things like user interface elements or menu choices -- overlaid on the video. It allows more flexibility of what you can create.

These are the factors -- the wide distribution of the player, the strong branding, and the flexible interface -- that make it the best solution for video. And people are starting to realize that.

Knowledge@Wharton: Currently Flash Video is only a solution for videos that are viewed online. It's not an option for downloadable videos; it doesn't have a digital rights management [DRM] scheme built

into it. Is that something you're going to add?

Lynch: Well, it's actually a little bit more nuanced than that. You *can* do what's called progressive download of Flash onto your machine and then display it from your own machine. Most people don't do that, but it's technically possible to do. And those videos don't have any digital rights management around them -- which is perhaps why many aren't distributed that way.

We are looking into what alternatives we might be able to provide for people to manage their content. But we're also very sensitive to the rights of individuals who are viewing that content. We want to be very careful about doing the right thing for all parties.

And anything we do inside the Flash Player basically has to live forever inside Flash. So we're being cautious about what we add in terms of DRM capabilities.

But, yes, it's absolutely something that we're looking at and talking about actively with our partners.

Knowledge@Wharton: Historically Adobe, and before that Macromedia, has been very focused on the mobile device space. You recently introduced a new version of Flash Lite. But you're competing against Java which has a greater installed base within the U.S. How do you compare the two technologies, and how do you plan to compete on mobile devices?

Lynch: Java does have greater reach on mobile phones today. They started earlier than we did with Flash on mobile phones and they've done a very good job of getting distribution.

Although it started later than Java, the Flash Player -- called Flash Lite on mobile phones -- is now also having incredible distribution. We've signed up the top handset manufacturers, the top operators, to distribute Flash on mobile phones. And the experiences that you can build with Flash are really incredible.

One of the things that happened with Java is that their early move into mobile was done through many different implementations of Java on phones. So, it's actually not a consistent implementation.

The result now is, yes, there's a lot of Java out there but it's highly variable. To create Java applications that work on mobile [devices] you must implement many, many times over in order to deploy on these telephones -- like over a hundred different times. So that's not very sustainable.

We're trying to be more consistent in our deployment of Flash -- which takes a little bit longer, but we hope will build a stronger base for people to create content.

Knowledge@Wharton: Is this part of the challenge? Don't the carriers have a vested interest in *not* providing a carrier- and device-independent platform?

Lynch: Well, the carriers that we work with are very focused on their relationship with their customers and the expression of their brand. And whatever technologies they can use to accomplish those things is what they want to do.

Flash can help them show their brand better and reduce the cost of creating the brand on the devices. We're seeing a lot of interest from them.

And Java and Flash aren't head-to-head competitors on mobile devices. They're included in many mobile

phones together. It's not an either/or situation. There's a potential for people to use those technologies together. My guess will be you'll see more people starting to do that over time as they realize the strengths of each technology and how you might use them together.

Knowledge@Wharton: What's the biggest technical challenge facing Adobe today?

Lynch: We've been working on our biggest technical challenges which are transitioning our [software] runtime [environments] from being primarily for animations to being for applications.

And this is a technical challenge we faced head-on. We spent the past few years re-writing the virtual machine, the Flash Player, which was a major effort. And a big bet on that transition that's happening.

It's critical to the success of our company that we are really successful expanding Adobe's business beyond design tools and web development tools to web application development tools and runtimes and servers.

And that's what we're starting to see play out right now. I think we've done a good job on the new Flash player [along with a new version of] Flex.

That has been our biggest challenge and I think we've executed well on it. And we look forward to seeing how it gets used by our customers.

This is a single/personal use copy of Knowledge@Wharton. For multiple copies, custom reprints, e-prints, posters or plaques, please contact PARS International: reprints@parsintl.com P. (212) 221-9595 x407.