

Beyond AJAX:

Accelerating Web Applications with Real-Time Event
Notification

Dr. Rohit Khare
August 2005

Executive Summary

A set of technologies dubbed 'AJAX' has triggered a surge of interest in dynamic interfaces to Web services such as email, maps, and calendars. At its core, however, AJAX is still locked into the Web's limited request/response architecture. That prevents servers from pushing real-time alerts such as incoming service calls, market data, or news headlines to Web browsers.

However, it *is* possible to stream events over today's Web infrastructure. And it's possible without installing any new software in the browser, a key part of the appeal of AJAX. KnowNow is already shipping such technology— and has been for five years.

This white paper will discuss potential of AJAX-style Web applications, as well as the fundamental limits of pull-based architectures. It will also discuss how KnowNow's technology works and its advantages for accelerating Web applications with real-time event notification.

1. Introduction

The rapid spread of the term AJAX (Asynchronous JavaScript And XML) — from a Weblog [9] to the *Wall Street Journal* [10] within weeks! — might lead developers to assume it's a breakthrough that heralds the death of desktop applications. There's certainly a kernel of truth in that: a slew of new Web applications under the AJAX banner have re-defined end users' expectations of what's possible within a Web browser by offering smooth scrolling, incremental updates, and more responsive input forms.

Nevertheless, so-called 'fat clients' that run on PCs still retain one fundamental advantage over the Web: real-time event notification. AJAX alone does not address instant messengers, stock tickers, or other collaborative applications that require "push" data streaming.

This seemingly-arcane distinction isn't merely about appearance or speed. Who really needs to know a fraction of a second sooner that another invitee has responded to a meeting request? The fundamental advantage of event notification over polling is that it can loosen the coupling between components in an application. Using a layer of "Web middleware" to dynamically route information about relevant changes to interested components can reduce software development, maintenance, and integration costs.

In this paper, we will explore the fundamental limits of browser-based applications and explain how AJAX solves some of the remaining problems — and which ones it doesn't. From there, we'll show how real-time event notification can be added to today's Web and how products such as KnowNow's are helping users take full advantage of technologies like AJAX and beyond.

2. Beyond the Desktop

Surveying the desktop landscape, only a few applications remain without any Web-based equivalents. Graphics-intensive CAD tools and games may demand the sheer horsepower of PCs and workstations, but even office productivity applications can be written entirely in Dynamic HTML: HalfBrain already hosted word processing, spreadsheets, and presentation tools five years ago!¹ [7, 11]

Nevertheless, desktop applications retain some advantages over Web applications in areas such as security, multimedia, and mobility. By design, browsers don't offer direct access to hardware such as security keys, cameras and microphones, or files on disk. Installing popular browsers plug-ins helps circumvent these limits, though.

For example, Adobe's latest Acrobat Reader plug-ins provide such rich public-key encryption, digital signature, and rights management services that San Jose's public library

¹ Part of that team regrouped to form Oddpost, the AJAX-style Webmail application acquired by Yahoo! as its response to Google's Gmail. Laszlo also ships a Flash-based interactive Webmail demo; all of these use polling to check for new messages, as discussed later in §4.

website can now lend out e-books to its patrons [26]. Macromedia's Flash Player plug-in powers podcasting startup Odeo's web-based multi-track recording studio; Flash can even support video recording [4, 8]. And `userData`, an obscure feature of Microsoft Internet Explorer's Cascading Style Sheets (CSS) implementation [5, 17, 18], enabled Salesforce.com to simulate off-line access to its web application by storing customer records on disk [25]. Furthermore, innovations such as cross-domain user scripting using Greasemonkey in the Mozilla Foundation's Firefox browser point towards an even richer future for the browser as a user interface platform [21].

3. AJAX Cleans Up

These plug-ins generally support the page-at-a-time paradigm of Web applications. By contrast, the key insight of AJAX-style applications is simple: decomposing jarring transitions that download an entire new Web page into a series of smaller, more frequent transactions.

Rather than retrieving complete HTML pages, AJAX emphasizes retrieving smaller, structured XML elements in response to user interface actions that, in turn, are expanded into an HTML rendering within the browser. For example, it dramatically reduces the perceived latency of scrolling a map to move around already-loaded image tiles, while only downloading the newly-visible tiles. Similarly, automatic completion of input form fields can take advantage of quick calls to retrieve data from the server while the user is still typing.

Given the sheer bulk of pages generated by modern applications servers and HTML template languages, the bandwidth savings of merely avoiding retransmitting portions that haven't changed can be quite dramatic. Judiciously-chosen URLs for identifying fragments of application state that can be cached safely automatically take advantage of browser- and proxy-caches to further accelerate the user experience.

AJAX is considered "asynchronous," but could be described more precisely as "non-blocking," because user input doesn't lead to an hourglass cursor. At the same time, users are conditioned to expect that submit buttons lead to blank intervals; the AJAX approach requires providing users alternative visual feedback that something is happening in the meantime. This is the most charitable explanation of why a fair bit of "eye candy" such as wipes, fades, bounces, sliders, and all manner of animation trumpet the self-consciously hip design aesthetic of so-called "Web 2.0" sites sailing under the AJAX banner.

4. Rewiring the One-Way Web

In the middleware community, the formal definition of asynchrony requires the ability to send a message at any time, *in either direction*. This helps reduce coupling between the presentation layer and application logic; event loops are the basis of the Model-View-Controller (MVC) style of GUI application development, for example.

Consider the challenge of developing an AJAX webmail application. Clicking on a subject line to delete a message that promptly sends a message to the server (and immedi-

ately disappears from view) is a vast improvement over 'plain HTML' webmail. Checking for the arrival of new messages still requires regular polling, though.

A desktop application built using MVC would reduce delays and bandwidth by letting the server send a notice as soon as new message arrives, and no sooner. HTTP would appear to make server-initiated transmission impossible. At its core, the Web is built to pull data from central servers, not to push it out peer-to-peer.

An AJAX email application must poll a server repeatedly to check for new messages, rather than letting the server send a notice as soon as that event occurs. This wastes bandwidth and increases delay for the user. This sort of pseudo-push using frequent polling is what gave the "Push" buzzword such a black eye in 1996-98 [6].

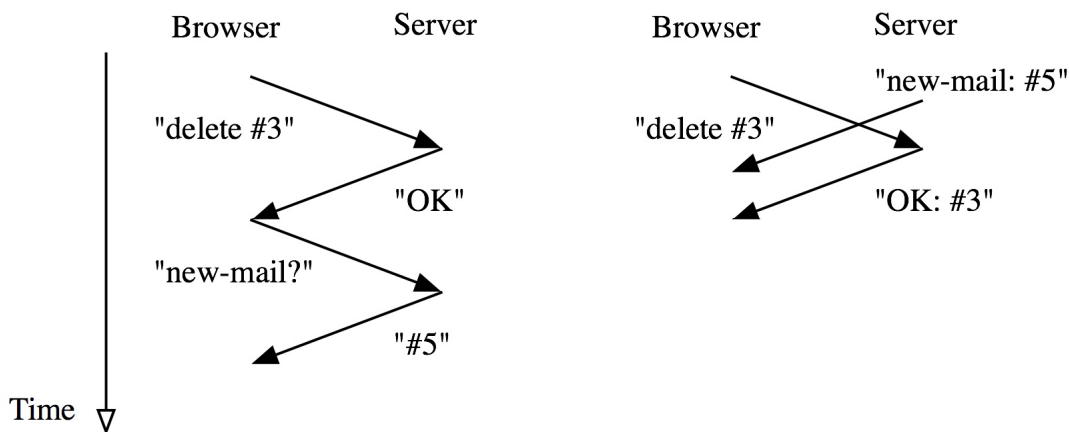


Figure 1: While AJAX's characteristic communication pattern is still a saw-tooth, true client/server applications can send overlapping messages, in either direction, at any time.

5. Beyond AJAX

Ambitious developers are already taking advantage of commercial and open-source toolkits to go beyond AJAX to power genuinely asynchronous Web applications — without installing any new software on the client.

Adding genuine asynchronous messaging to the Web is only possible due to some clever interaction between HTML, Javascript, and Web browsers' rendering algorithms. One of the earliest advances in browser design was to incrementally display a page while it was still downloading. With the invention of `nph` ('no-parse-headers') Common Gateway Interface (CGI) scripts in 1994, Web developers have had ways to 'pipeline' data to clients as soon as the server could compute it.

The simplest way to exploit this is to turn the browser into a 21st-century "green screen" dumb terminal. Manuel Kiessling's open-source A Really Simple Chat ([ARSC](#), [14]) uses AJAX techniques to send input lines upstream, while a modified HTTP server that can hold open thousands of simultaneous connections re-broadcasts the chat stream to other users. As each line of HTML eventually streams into the browser, it triggers the

execution of a new bit of JavaScript that updates the display to insert new lines into the transcript.

It takes a lot more to take advantage of incremental rendering to power “real apps” that traffic in structured data, not just adding lines to a display. What began as a convenience for dealing with slow dialup modem links, KnowNow has turned into a clever facility for adding real-time XML event notification over standard HTTP by using hidden HTML FRAMEs to trigger events in the enclosing application. Easier said than done — connecting multiple open windows within the confines of the JavaScript security sandbox had confounded many developers before KnowNow’s invention of the “microserver.”

6. The Potential of the Two-Way Web

The subtler and broader implication of combining AJAX with asynchronous event notification is that it can extend publish-and-subscribe application integration across the Internet to desktop PCs, mobile phones, and other devices. Within the enterprise, it’s easy to assume that different components can support peer-to-peer messaging; beyond the firewall, it’s often been assumed impossible to run a “server” process that listens for events from the Internet.

Assuming one did have a miniaturized server, though, it has far more utility than merely broadcasting information from a server to dumb terminals, as the applications in the previous section did. It affords developers a chance to give a name to shared concepts, and thus decouple programs from having to explicitly command each other to display or retrieve information. Rather than forcing a server to invoke “redraw” on each client that may be displaying a stock price, it only has to publish to the shared Web address <http://myServer.com/myStocks/FOO> ; other clients need only subscribe to the same URL, and the messaging fabric takes care of all the scheduling, transformation, and security constraints to connect them together. Next month, another developer can write an entirely separate application that visualized stock prices in a different way, without further coordination with the server; the month after that, a wireless carrier can offer to filter major price movements and send alerts to brokers’ phones. All of this is made possible by viewing the Web as a fundamentally two-way repository of ever-changing information, not the UI tricks of AJAX.

Several open-source platforms provide powerful abstractions for connecting fully-interactive Web UIs to enterprise applications and Web services. [Nevow](#) [22] and [Pushlets](#) [29] extend the event loop familiar from model-view-controller GUIs for Python and Java, respectively. [Mod_PubSub](#) is designed as an event bus that uses URL pathnames as topics to implicitly invoke programs written in a wide range of languages [13]. Historically, Mod_PubSub was the prototype for KnowNow’s award-winning [16, 19, 20, 28] commercial product, [LiveServer](#), which provides enterprise-class [15, 30] scalability, reliability, and security.

7. Development Tools for Web 2.0

The emergence of an ecosystem of industrial-strength tools, services, and solution providers bodes well for the revolution dubbed “Web 2.0.” Tim Berners-Lee’s original vision

for the Web to become “the universe of all network-accessible information” goes far beyond the browser, and so will this movement.

For example, KnowNow’s LiveServer can even deliver real-time feeds into ordinary Excel spreadsheets (as well as track any updates users type in). It includes connectors for major database packages; and parses several flavors of Web syndication formats. Just as behind-the-firewall Enterprise Application Integration (EAI) vendors found, unlocking the power of an Internet-scale event bus requires a well-maintained flotilla of connectors and adaptors to work in a real world full of legacy applications. To complete the cycle, developers deserve platforms that support them in developing, debugging, testing, and deploying large-scale, high-quality supportable, portable applications that are well-integrated with existing internal systems.

Part of that challenge is recognizing that not all systems are ready for publish-subscribe integration, either. Transactional Remote Procedure Calls (RPC) still have their place. Recall our earlier webmail example: it’s reasonable to express the user’s desire to see all new messages as soon as possible as a persistent subscription to their INBOX. But users would be justifiably upset if a newly composed message were sent twice, or not at all. For such commands, developers are still free to use AJAX-style XMLHttpRequests, or better yet, KnowNow’s integrated SOAP support for gatewaying transactions.

Underneath the covers, KnowNow still provides the advantages of loosely-coupled communication by transforming an RPC-style SOAP request into pub/sub so that other applications can monitor the status of a transaction. For the developer, though, KnowNow provides a unified approach to either style of software architecture.

The clear benefits of migrating desktop applications to the Web in terms of maintenance, security, and scalability must be weighed against the costs of slower response times, limited interactivity, and less-than-beautiful graphical interfaces. With AJAX, multimedia plugins, and push technology like KnowNow’s, the Web is closer than ever to becoming a viable default platform for application development.

All the News that's Fit to Push

Our founding a-ha! moment came to us when we asked “What do you need to know, now?” — the realization that most of business life is reacting to news flashes from the front lines. Five years ago, Really Simple Syndication (RSS, [23]) was an obscure technology for websites to plug into Netscape's Netcenter portal, but it quickly became the star of KnowNow's demos to customers and investors.

Specifically, while newspaper websites insisted on offering separate feeds for each brand and section, businesspeople needed to monitor *any* occurrence of a partner or competitor across *all* of those channels. In 2000 KnowNow's prototype code — since open-sourced as Mod_Pubsub [12] — only had hundreds of feeds to monitor; today the award-winning [1] technology at [PubSub](#) [27] monitors tens of millions of sources on behalf of hundreds of thousands of users.

Recently, these two companies joined forces to add PubSub's “premium, controlled access to non-traditional outside sources of information and the blogosphere” [2] to KnowNow's flagship AJAX-style application, [SpeedReader](#) [3]. PubSub's consumer offering requires installing a customized, binary plugin for certain Web browsers that connects using the Jabber instant-messaging protocol [24]. By connecting their “search-the-future” service to KnowNow's Enterprise Syndication Solution (ESS), organizations can deliver immediately actionable news feeds, blog content and critical communications to desktops without polling or client-side software installation.

“Headline news” includes a lot more than stock prices and traffic incidents — changes in customer relationships, systems going down, or an out-of-stock situation are just as “newsworthy” within an organization. When time is of the essence, KnowNow's real-time event streaming technology trumps AJAX-style polling.

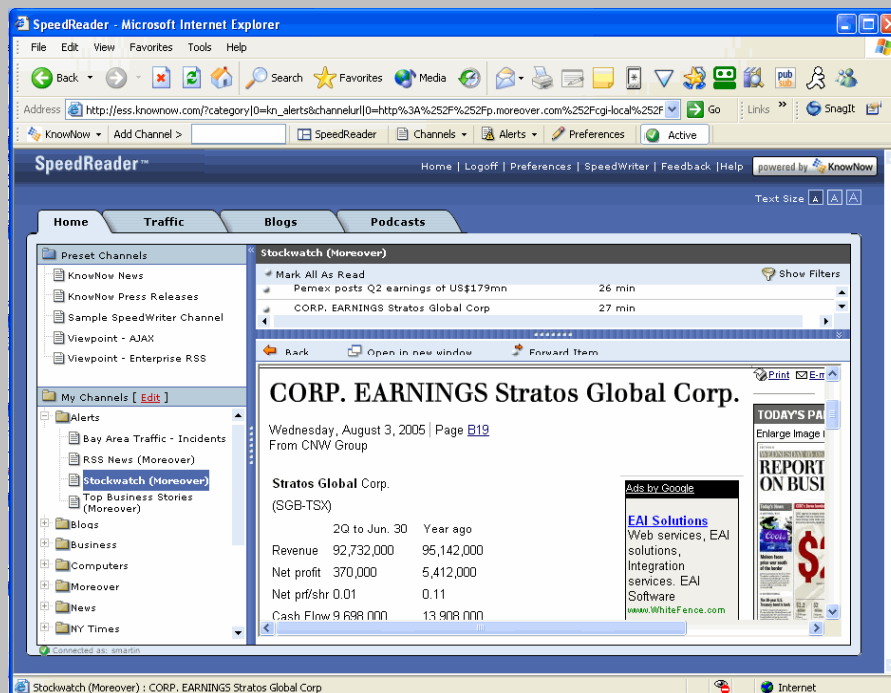


Figure 2:

KnowNow developed *SpeedReader*, a rich, interactive, and *real-time* RSS news aggregator in AJAX style — an application without any software installation on the client.

References

- [1] *First Annual Horizon Awards: PubSub.com in Computerworld*, (forthcoming) September, 2005. <http://www.pubsub.com/press/2005-07-25.php>
- [2] *KnowNow and PubSub Team to Provide Live, Persistent Search Feeds for KnowNow Enterprise RSS Customers*. Business Wire (Press Release), 15 August 2005. <http://www.pubsub.com/press/2005-08-15.php>
- [3] *RSS Goes Corporate: Information overload also affects companies, and they're more likely to pay for a cure in Red Herring*, 18 July, 2005. <http://www.redherring.com/Article.aspx?a=12664>
- [4] Calcanis, J. *Odeo podcasting software demo by Evan Williams*. BloggingETech.com, 17 March 2005. <http://etech.weblogsinc.com/entry/1234000687036533/>
- [5] Cox, K. *Inside Technique : User Persistence Features of IE 5*. SiteExperts.com, 19 November 1999. <http://www.siteexperts.com/ie5/tips/ts01/page1.asp>
- [6] Doan, A. *Pushing back: The growing popularity of push technology is creating problems and generating solutions in Infoworld*, 4 August, 1997. <http://www.infoworld.com/cgi-bin/displayStory.pl?features/970804push.htm>
- [7] Einstein, D. *Halfbrain beats Microsoft, Sun brains in Forbes*, 29 November, 1999. <http://www.forbes.com/1999/11/29/feat.html>
- [8] Everett-Church, J. *Update from Flash Forward NYC*. 7 July 2005. <http://justin.everett-church.com/index.php/2005/07/07/update-from-flash-forward-nyc/>
- [9] Garrett, J. J. *Ajax: A New Approach to Web Applications*. Adaptive Path, 18 February 2005. <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [10] Gomes, L. *Google Pioneers Use Old Microsoft Tools In New Web Programs*. *Wall Street Journal*, New York, NY, 14 March 2005, p. B1. <http://online.wsj.com/article/0,,SB111075227698078072-email,00.html>
- [11] Guttman, S. *Clients, Employers & Projects (Halfbrain.com)*. 2005. <http://www.gothisway.com/projects.htm>
- [12] Khare, R., Rifkin, A., Sitaker, K. and Sittler, B. *mod_pubsub: an open-source event router for Apache*, 2002. <http://mod-pubsub.sourceforge.net/>
- [13] Khare, R. and Taylor, R. N. *Extending the Representational State Transfer (REST) Architectural Style for Decentralized Systems*, in *26th International Conference on Software Engineering (ICSE)*, (Edinburgh, Scotland, 23-28 May 2004).
- [14] Kiessling, M. *ARSC: A Really Simple Chat*. 25 April 2005. <http://manuel.kiessling.net/projects/software/arsc/>
- [15] KnowNow. *LiveServer 3.1 Developer's Guide*, ver. 3.1, Sunnyvale, CA, 2005. http://www.knownow.com/products/docs/user_docs/Rev-29-LiveServer_Developers_Guide.pdf
- [16] McKellar, H. *2004 Promise and Reality Winners (KnowNow) in KMWorld: Content, Document, and Knowledge Management*, 1 January, 2005. vol. 14.

- http://www.kmworld.com/publications/magazine/index.cfm?action=readarticle&Article_ID=1951&Publication_ID=125
- [17] Microsoft. *Persisting Form Data*. 1997.
<http://msdn.microsoft.com/workshop/author/persistence/howto/formdata.asp>
- [18] Microsoft. *userData Behavior*. 1999.
<http://msdn.microsoft.com/workshop/author/behaviors/reference/behaviors/userdata.asp>
- [19] O'Neill, L. T. *2003 Readers' Choice Awards: The winners are in (KnowNow)* in *Infoworld*, 25 July, 2003.
http://www.infoworld.com/article/03/07/25/29FErcamain_1.html?s=feature
- [20] Open Systems Advisors. *Crossroads 2002 A-List Award Winner: KnowNow*. 7 January 2002. <http://www.crossroads-osa.com/research/company/Knownow.html>
- [21] Pilgrim, M. *Dive Into Greasemonkey: Teaching an old web new tricks*. 9 May 2005. <http://diveintogreasemonkey.org/>
- [22] Preston, D. *Nevow: A Web Application Construction Kit*. DivMod.org, 2005.
<http://nevow.com/>
- [23] Quain, J. R. *Fine-Tuning Your Filter for Online Information (RSS)*. *New York Times*, New York, NY, 2004.
<http://www.nytimes.com/2004/06/03/technology/circuits/03basi.html>
- [24] Saint-Andre, P., Hildebrand, J. and Wyman, B. *Transporting Atom Notifications over the Extensible Messaging and Presence Protocol (XMPP)*. 14 January 2005. <http://www.xmpp.org/drafts/draft-saintandre-atompub-notify-02.html>
- [25] Salesforce.com. *Offline Edition Data Sheet*. San Francisco, 2004. 1pp.
<http://www.salesforce.com/us/pdf/datasheets/offline-edition.pdf>
- [26] San Jose Public Library. *San José Public Library Launches New Digital Library*. 10 March 2004. http://www.sjlibrary.org/about%5Cnews/story_04-03-10.htm and <http://ebooks.sjlibrary.org/>
- [27] Smith, S. *The New Business of Content Alerts (PubSub & MessageCast/MSN Alerts)*. *EContent Magazine*, 18 April 2005.
<http://www.econtentmag.com/Articles/ArticlePrint.aspx?ArticleID=7854>
- [28] Udell, J., Gillmor, S. and Eds. *2002 Technology of the Year Award: Publish/Subscribe Technology: KnowNow 1.5* in *Infoworld*, January 24, 2003.
http://www.infoworld.com/article/03/01/24/2002TOY-sb_1.html
- [29] van den Broecke, J. *Pushlets - Whitepaper*. Pushlets.com, 6 August 2002.
<http://www.pushlets.com/doc/whitepaper-all.html>
- [30] Windley, P. J. *KnowNow shows off integration agility* in *Infoworld*, 19 November, 2004. http://www.infoworld.com/article/04/11/19/47TCknownow_1.html

About the Author

Rohit Khare is the Director of CommerceNet Labs, which is investigating decentralized electronic commerce. Prior to that, he co-founded KnowNow in 2000 with Adam Rifkin based on his doctoral research at UC Irvine. There, he studied the development of application-layer Internet protocols and architectural styles for decentralized systems with Prof. Richard N. Taylor, for which he won an ACM SIGSOFT Distinguished Paper Award.

Dr. Khare's participation in Internet standards development with world-renowned technical teams at MCI's Internet Architecture group and the World Wide Web Consortium (W3C) at the MIT Laboratory for Computer Science, where he focused on security and eCommerce issues, led him to found 4K Associates, a standards-strategy consultancy, as well as editing the World Wide Web Journal (W3J) for O'Reilly & Associates. Rohit received his B.S. in Economics and in Engineering and Applied Science with honors from Caltech in 1995 and his Master's and Ph.D. in Software Engineering from UC Irvine in 2000 and 2003, respectively.