



The Four Quantum States of AJAX

By Kevin Hakman

Co-founder, TIBCO General Interface, TIBCO Software Inc.

Uh oh! Here comes yet another technology buzzword. This time it's "AJAX". In addition to being a household cleaner, a character from Greek mythology, and the name of a European soccer club, AJAX is Asynchronous communication, JavaScript, and XML. These technologies, which are already present in your browser, make it possible to retrieve information without refreshing a Web page.

Citing Google's use of AJAX, the *Wall Street Journal* noted that AJAX represents "a big step toward the Holy Grail of having the kinds of speed and responsiveness in Web-based programs that's usually associated only with desktop software."¹ And increasingly, developers, entrepreneurs, and industry analysts are grasping the opportunities that AJAX presents.

So what's all the excitement? The need and desire to provide richer functionality in Web-deployed applications, a.k.a. Rich Internet Applications (RIA), have been around for a while. Historically, attempts to deliver on this vision however have required supplemental browser technologies such as Java applets, Active-X controls, and browser plug-ins. But each of these approaches has challenges. There are security concerns around Active-X. The Java runtime version required by an applet may not be present, or applets may be blocked by firewalls. The right version of a plug-in may not be present, and locked-down enterprise desktops may prevent it from being updated. With an AJAX-enabled RIA, even the most cynical software engineer acknowledges that there is no need for supplementation. Developers and businesses alike are excited about AJAX because it leverages functionality that is readily accessible in the browser.

The base technology for AJAX has been available since 1999 when Microsoft put an XML parser into Internet Explorer and formalized, in its XML HTTP Request object, a method for background communications that eliminated the need for creative work-arounds, such as using hidden portions of a Web page. The early AJAX pioneers focused on business-to-business applications where Internet Explorer had a ubiquitous presence. However, with the recent addition of AJAX capabilities to other browsers, most notably Mozilla Firefox, developers are enhancing consumer applications with richer capabilities. Support across multiple browser types enabled Google to provide its much touted maps.google.com site to a world-wide audience. This support has also contributed to the popularization of the AJAX acronym, coined in early 2005 by Adaptive Path, a San Francisco-based business and Web consulting firm. Now the buzz is that any software start-up visiting a Sand Hill Road venture firm should have the term in its PowerPoint deck.

¹ Lee Gomes, "Google Bets on AJAX," *The Wall Street Journal*, March 15, 2005.

Deconstructing “AJAX”

A brief survey of ajaxian.com—a developer website that catalogs and facilitates discussions of all things AJAX—quickly reveals that AJAX is being applied to everything from simple communication scripts to full-blown RIA development and deployment frameworks. One of the problems with the term AJAX is that, strictly speaking, the concepts underlying the acronym do not encompass a graphical user interface. Tightly defined, AJAX simply describes a technology that transports information to and from the browser and not how that information is displayed. To purists, AJAX is about communicating asynchronously from the browser using JavaScript and XML, nothing more. To others, through their experiences with Google Maps and Yahoo!’s new email offering, AJAX represents a desktop-like GUI that leverages the pre-existing HTML, DHTML, and vector-based rendering capabilities of the browser. When they speak of AJAX, just as the *Wall Street Journal* did, they mean the rich user interactivity of the desktop—not just background communications. Either way, “AJAX” is a lot catchier than “DHVAJAX.” Accordingly, it’s likely that developers will continue to use it for a broad spectrum of uses. So, it’s important to understand the “quantum states” the term has taken on.

Four Quantum States

This article proposes four quantum states of AJAX in an effort to tease apart and classify the plethora of AJAX technologies available today.

1. Communication libraries
2. User Interface Components
3. Rich Internet Application frameworks
4. RIA Frameworks with robust visual tooling

Your unique needs will determine which of the following AJAX incarnations makes the most sense for you.

Communication Libraries – libraries that provide the communication elements named in the acronym and nothing more.

Recall that your desktop Web browser—assuming you’re not using an obscure one—has asynchronous communications, JavaScript, and XML built in. You can directly interact with the JavaScript APIs that Internet Explorer and Firefox provide with their XML and asynchronous communication objects. Of course, the rub is that the APIs for each browser are slightly different. Accordingly, libraries such as Prototype, SAJAX, and AJAX.NET handle the browser differences for asynchronous communications, while Google’s AjaXSLT provides a JavaScript implementation of XSLT and XPath that enables you to work with XML structures more easily across a variety of browsers.

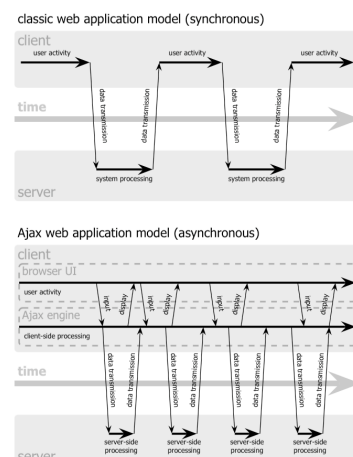


Image source:

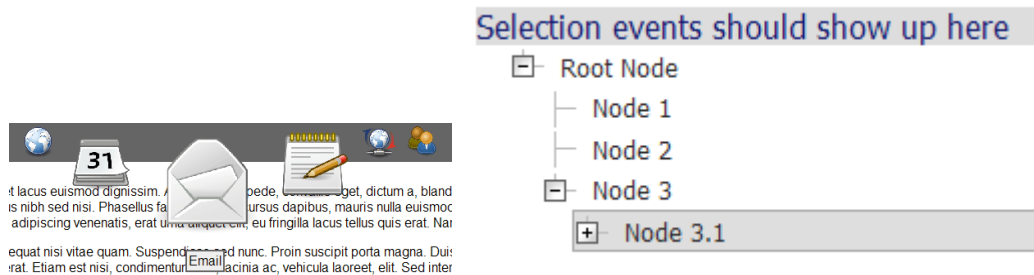
<http://www.adaptivepath.com/publications/essays/archives/000385.php> (c) AdaptivePath. Publisher will need to get permission to use form AdaptivePah or re-illustrate this concept.

Potential Users

If all you want to do is update the options in a select box, swap out content in a HTML page asynchronously, or perform other simple content updating tasks, communication libraries offer strong CSS and DOM capabilities. And if you love working with low-level objects, they are a good fit. If you'd prefer to have reusable GUI components at your disposal, read on.

GUI Components – graphical user interface components that are scripted to communicate asynchronously and display the resulting information without refreshing the entire Web page.

What you find in libraries offered by Dojo Toolkit, Rico, eBusiness-Apps, and others are out-of-the-box GUI controls that can retrieve data in the background and render it in the GUI control. Typically, one finds GUI controls that can be used individually or several at a time to enhance portions of a Web page. These components may or may not share common communication libraries or a common look and feel.



The image shows a screenshot of a web page with a tree view overlay. The tree view is titled "Selection events should show up here" and contains the following nodes: Root Node, Node 1, Node 2, Node 3, and Node 3.1. Node 3.1 is highlighted with a grey background. The web page content includes a calendar showing the number 31, an envelope icon, and some text.

Widgets: Part of the Dojo Toolkit

Potential Users

Developers with existing HTML pages can add GUI controls that enrich a page with these ready-made libraries. If you aspire to full-fledged browser-based applications rather than occasional enhancements, AJAX-based RIA frameworks (described later) will help you head off issues that arise from mixing and matching components that do not share a common infrastructure.

Rich Internet Application (RIA) Frameworks – AJAX-based RIA Frameworks that deliver ready-built, fully-integrated libraries of interactive GUI components and share common approaches to data acquisition, persistence, and communication. Some frameworks center on server-side models, while others are sophisticated client-side libraries.

In these frameworks, GUI controls share a common look and feel—in contrast to stand-alone, mix-and-match GUI components. In addition, in RIA frameworks, communications, data caching, and other core services are shared among GUI controls, usually in an extensible model that supports object inheritance.

There's also a semantic shift in the language used to describe objectives. For example, in RIAs, the notion of a "page" goes away and is replaced by application "screens," "controls," "modal dialogs," and other conventions of the client-server GUI world that predated the Web. If you think in these terms, then AJAX-based RIA frameworks such as Bindows, Isomorphic, BackBase, JackBe, TIBCO General Interface and others make sense.

Potential Users

RIA Framework users are likely to be development organizations creating both tactical and mission-critical solutions. Here, developers are creating complete RIAs, rather than simply enhancing an HTML page with a few GUI components. Thus, these desktop-like applications transcend simple enriched Web pages. At the same time, the capability still remains to place a component or two into an existing HTML page and enhance its functionality. In the latter case it may require a bit more overhead to get the component up and running. Often, though, the component definitions are cached at the client, making overhead less of an issue.

RIA Frameworks with Visual Development Tooling –

AJAX RIA frameworks accompanied by visual authoring capabilities via a custom development environment or plug-ins to popular IDEs.

Here, developers write significantly less code than if using a framework's APIs directly. In comparative terms, developers don't write much code using pure AJAX communication libraries or AJAXified GUI components, because their goals are usually relatively limited. Hence, in building full applications with a framework, it's powerful, but time consuming to write directly to the framework APIs. AJAX RIA frameworks such as BackBase, JackBe, TIBCO General Interface, and others provide rapid development tools that increase developer productivity and reduce the cost of owning AJAX RIA applications. In order not be left out of a hot development trend, Microsoft has also announced its intention to provide an AJAX framework code named Atlas that works with its .NET server-side framework and likely it's visual development tools. These tools complement frameworks through access to visual debugging, communication tracing, API references, code generation, and other time-saving features.

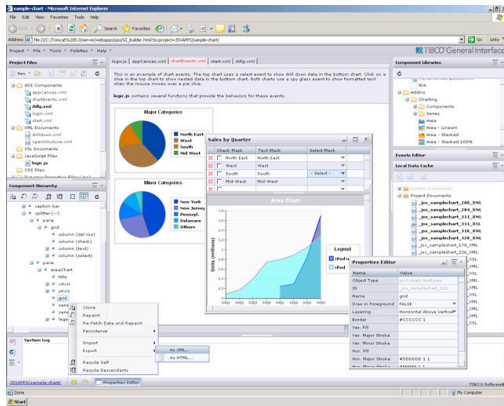


Image caption: TIBCO General Interface Builder is an AJAX application for building AJAX applications.

Potential Users

Savvy AJAX RIA tooling applies to much the same group of users as those considering an AJAX RIA framework. For many organizations, author time efficiency is paramount, so tools must make building applications faster and easier.

Wrapping it up:

While the strictest definition of AJAX addresses only communications, the term is being used to describe a new class of more richly-featured GUI component and a class of Rich Internet Applications that require no plug-ins, Active-X controls, applets, or other technology to supplement the browser while achieving functionality and performance that is comparable to desktop-installed applications.

Given that, when potential users ask whether AJAX is simply an ingredient in a no-plug-ins, no-Java applets, no-Active-X approach to RIAs, or a class of RIAs unto itself, the answer it seems is "all of the above."

A Snapshot of Tools Available for all AJAX States:

| Level | Tool/Product | Provider |
|-------|--------------|----------|
|-------|--------------|----------|

| | | |
|---|---|---|
| 0 | IE 5.0 Firefox AjaXSLT | Internet Explorer (www.microsoft.com) Mozilla Corporation (www.mozilla.com) Google (www.Google.com) |
| 1 | SAJAX Ajax.NET Prototype | ModernMethod (www.modernmethod.com) BorgWorX (www.borgworx.com) Conio.net |
| 2 | dojo 0.2 Rico Web ComboBox V3 - AJAX | Dojo (www.dojotoolkit.org) Rico (www.openrico.org) www.ebusiness-apps.com |
| 3 | BINDOWS™ SmartClient™ | Bindows (www.bindows.net) Isomorphic (www.isomorphic.com/) |
| 4 | Backbase AJAX JackBe Visual GUI Builder TIBCO General Interface | Backbase (www.backbase.com) JackBe (www.jackbe.com) TIBCO (developer.tibco.com) |

Author's note: The products in this table are a sampling of the plethora of AJAX-related libraries, frameworks, and tools evolving. A more complete and ever-growing list can be found at http://ajaxpatterns.org/Javascript_Multipurpose_Frameworks.