June 2006

Innovations Affecting Us -- Do Web Applications Need to be Cleaned Up?

Kristen DeVoe

College of Charleston, devoek@cofc.edu

Follow this and additional works at: <https://docs.lib.purdue.edu/atg>

Part of the [Library and Information Science Commons](#)

Recommended Citation

DeVoe, Kristen (2006) "Innovations Affecting Us -- Do Web Applications Need to be Cleaned Up?," *Against the Grain*: Vol. 18: Iss. 3, Article 40.

DOI: <https://doi.org/10.7771/2380-176X.4988>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

PRODUCT PLACEMENT GOES ... WELL, COMICAL

by Bruce Strauch (The Citadel)

Product placement has long been rife in movies and TV. Now it comes to comic books. **Time Warner's DC Comics** has a new series boosting the **Pontiac Solstice**. **Marvel Entertainment** has the Nike swoosh on the t-shirts of the New **X-Men** and may feature the new **Dodge Caliber**.

Astounded? Comic readers now have a demographic reaching into their twenties and thirties and the \$450 million annual business is actually in competition with **Dennis Publishing's Maxim and Stuff**. **DC** intends to bring in advertisers pushing health and beauty care, shaving cream and razors.

See — **Brian Steinberg**, "Look—Up in the Sky! Product Placement!" *The Wall Street Journal*, April 18, 2006, p.B1.

CHAOS THEORY RIPPLES OUT OF THE USPTO

by Bruce Strauch (The Citadel)

Admin changes at the **US Patent and Trademark Office** have led to a flood of new patents and a tsunami of litigation. The **USPTO** became a "profit center" leading to pressure to grant more patents leading to patents on peanut butter and jelly sandwiches and a method of swinging on a swing invented by a five year old. While those might be harmless, lack of talent at the **USPTO** permitted patents on broad ideas related to mobile email without a whole lot of detail. And that led to the **BlackBerry** fiasco.

Now innovators are regularly being shaken down by holders of dubious patents and most don't have **BlackBerry** level money to ward off the attack. *But see the good news immediately below.*

See — **Adam Jaffe and Josh Lerner**, "Innovation and Its Discontents," *The Wall Street Journal*, March 21, 2006, p.A14.

THE END OF AUTOMATIC INJUNCTIONS IN PATENT LITIGATION

by Bruce Strauch (The Citadel)

The **Supreme Court** unanimously gave **eBay** a victory over **MercExchange** in the fight over the "Buy It Now" function, ruling that injunctions in patent infringement should only issue after weighing a variety of factors including the public interest. Complex systems use scads of minor inventions and can pay money damages for an infringement rather than shutting down the entire product as was threatened in the famous **BlackBerry** case.

Federal courts had been interpreting the patent act's "a court *may* issue an injunction" against an infringer into "*shall* issue." And that gave a green light to shake downs. Wait for a successful business to emerge, and then with the threat of shutting it down, rip off far more than the patent is worth.

See — **Jess Bravin**, "Ebay Ruling Changes Dynamic in Patent-Infringement Cases," *The Wall Street Journal*, May 16, 2006, p.B1.

IN SOFTWARE, PLUS ÇA CHANGE ...

by Bruce Strauch (The Citadel)

In the bad old days, acquiring software meant months and millions spent to license it, buy necessary hardware, and get your IT gang to set it running. Along came **SAAS — Software as a Service** promising a world of improvement. Now you use a Web browser to connect to the software.

But as the **SAAS** style takes hold, they are growing as complex as the old system they replaced. Once you're hooked, consultants, integrators and add-on pieces of software are suddenly necessary.

See — **Lee Gomes**, "Despite All the Hype, Getting Your Software On the Web Has Limits," *The Wall Street Journal*, April 12, 2006, p.B1.

Innovations Affecting Us — Do Web Applications Need to be Cleaned Up?

Column Editor: **Kristen DeVoe** (Electronic Resources Librarian, College of Charleston)
<devoek@cofc.edu>

Since their introduction Web based applications have rapidly undergone enormous change to improve functionality and the user experience. With the introduction of new Web development languages or Web applications, marked improvements have been made but user expectations continue to rise as functionality improves. With regards to efficiency, functionality, and user experience, the Web of today is stuck in the past in many ways. So what will the Web of the future be like and how will we get there? For some Web developers, the answer to those questions lies with a Web development technique frequently referred to as **Ajax**. **Ajax** and **Ajax**-based Web applications have quickly become the topic of frequent discussion among Web developers and is something to be aware of as the Web continues to evolve.

What is Ajax?

Asynchronous JavaScript and XML, frequently referred to as **Ajax**, is not a newly created language or technology. The sum of its

parts, **Ajax** is an incorporation of several familiar technologies working in kind to create a more interactive Web environment for users. The main technologies at play in **Ajax** include the following:

- **JavaScript** as the scripting language that **Ajax** applications are written in. **JavaScript** defines the user workflow of the **Ajax** application.
- **XHTML (Extensible Hypertext Markup Language)** and **CSS (Cascading Style Sheets)** for standards based content mark-up and presentation. These define the look and feel of the application. In **Ajax** Web applications the styling of a user interface can be modified interactively using **CSS**.
- **The DOM (Document Object Model)** presents the structure of Web pages as a set of programmable objects that can be manipulated with **JavaScript**. Scripting the **DOM** allows **Ajax** applications to modify the user interface quickly or "on the fly."

- **The XMLHttpRequest object** for asynchronous data retrieval. This allows Web programmers to retrieve data from the Web server as a background activity.

So What Can Ajax Do for Web Applications?

Ajax helps to unlock much of the unrealized potential of modern Web browser technologies. One way that **Ajax** overcomes a major limitation in traditional Web applications is by exchanging small amounts of data with the server asynchronously behind the scenes of the Web application. The intent of this design is to make Web pages feel more responsive, so that the entire Web page does not have to be reloaded each time the user makes a change. With traditional Web applications each time the user makes a change to the page, such as executing a search in an internal search engine or adding an item to an online shopping cart, the entire page must reload anytime the user requests new data.

continued on page 73

This classic page-based application model causes unnecessary resubmission of the same information (such as navigational links or images) to the Web browser even when only small changes need to be made, and the constant page refreshing interrupts the flow of the user's experience. In contrast, an **Ajax**-based Web application is designed to increase the Web application's speed, usability and interactivity because the entire Web page does not need to reload each time the user makes a change to the data or requests new data from the server. The result is faster, more intuitive Web applications that are comparable in usability to desktop applications.

Examples of Ajax on the Web

Industry giants such as **Google** and **Microsoft** have begun to experiment with using **Ajax** in some pretty big ways. One common example of how **Ajax** can improve a user's experience with a Web application is **Google Maps** (<http://maps.google.com/>). The search feature on this page functions as a normal Web application and refreshes the entire page, but the map itself is an **Ajax** application. The map is composed of small tiled images and if the user scrolls the map far enough (using the mouse) to expose a new tile, that tile will be asynchronously downloaded. The submission of new information does not interrupt the user's workflow because the rest of the page is not resubmitted to the browser. The only thing that changes is the information that needs to change. All of the contextual and navigational links, the zoom control, and the other page features are accessible to the user while the new map data is being gathered. **Google Maps** is not the only **Google** tool that utilized **Ajax** development techniques. **Gmail** (<http://mail.google.com>) also utilizes the technique to allow users to open several mail messages at a time and also to update mailbox lists automatically even if the user is typing a message. **Google Suggest** (<http://www.google.com/webhp?complete=1&hl=en>) is another **Ajax** based tool that searches the server as you type for the most likely completion to your query. Suggested search terms are displayed in a drop down below the search box and also include the number of results for each term.

Ajax can also be seen elsewhere on the Web. **Writely** (<http://www.writely.com>), a Web based word processor, is a powerful example of how an **Ajax** based application can improve functionality and even rival that of a desktop based application. **Amazon.com's** search engine, **A9** (<http://www.a9.com>) is another example of how **Ajax** is currently used to improve Web applications. While doing a search if you select any of the sources that you searched, new search results feed into the page through a dynamically added column on the search results page. You can easily toggle search areas on or off without reloading the page.

Microsoft is also stepping into the **Ajax** arena by developing software (as yet unreleased) called **Atlas** which is designed to ease the creation of **Ajax**-style applications. **Atlas** gives developers a more structured environment for building applications, providing time-saving services such as an object model and debugging.

How Are Libraries Using Ajax?

Some libraries have begun to implement **Ajax** based applications on their Websites. **Virginia Tech** has implemented **Ajax** into the keyword search function of their **OPAC** (<http://addison.vt.edu/>). This feature, called the **Addison Guesstimate**, previews the search as the user types and notifies the user of the approximation of the results. Further explanation of this feature is available on the library Website at http://addison.vt.edu/screens/help_index.html.

Georgia Tech is another school that has implemented **Ajax** in a different way. They have used it to facilitate their e-journals search (http://www.library.gatech.edu/search_locate/electronic_journals.html) and on their databases page (http://www.library.gatech.edu/search_locate/databases.php). To utilize the **Ajax** features of the e-journals search, begin typing a journal title in the journal title field. To utilize the features on the databases page, start typing a database name in the "search for databases" field. As you type, the **Ajax** features begin to work when the drop down menu appears based on what you type into the search box.

Another example of how academic institutions, although not specifically libraries, are implementing **Ajax** is the **University of Wisconsin's People Search** (<http://www.wisc.edu/directories/?name=>). When you begin to search for a name in this directory, the number of results and a list of matching names are displayed.

One more example of how a library is implementing **Ajax** is on **Curtin University of Technology's Single Search @ Your Library** page (<http://library.curtin.edu.au/cgi-bin/search/search.cgi?query=&submit=Sea>). This implementation allows the results from different sources to display as they become available instead of waiting for the entire set to return before the page is rendered.

Is this the Future?

In the Web development world it is not uncommon for new languages and technologies to pop up daily and then fade quickly into oblivion. **Ajax** however shows promise as a way for developers to redefine the way that users interact with the Web. Major online players have shown a keen interest in utilizing **Ajax** to improve Web applications, and users may come to expect such ease of use from library Web pages in the future as well. For online services such as **Google** and **Yahoo!** ease of use is a key feature in getting users to come back and use their tools again, and **Ajax** is one way that they are attempting to ensure this return. Ease of use or usability are issues familiar to librarians working in

continued on page 74

MAKING A BIG IMPACT

Frontiers in Ecology and the Environment

Over the past year, *Frontiers* has seen a doubling of its institutional subscriptions as well as receiving an excellent impact factor (3.362) for a journal that is only three years old.

With its timely, cross-disciplinary science, accessible writing style, four-color graphics and visually appealing format, the journal has been welcomed by scientists from a wide range of disciplines, as well as resource managers, educators, and students.

Visit:
www.frontiersinecology.org
and see for yourself.

Frontiers sets itself apart from other scientific journals, but delivers the same high-quality, peer-reviewed literature as the best of them. And at only \$220–230 for a U.S. institutional subscription, it's great value for your money.

Contact your subscription agent or ESA Headquarters at 1707 H Street NW, Suite 400 Washington, DC 20006 Fax: 202-833-8775

esa

Promoting the Science of Ecology

www.frontiersinecology.org

NISO Metasearch Initiative Issues First Set of Recommendations

by **Cynthia Hodgson** (NISO; Phone: 412-372-0694; Fax: 412-380-4025) <cahodgson@earthlink.net>, **Andrew Pace** (North Carolina State University) <andrew_pace@ncsu.edu>, and **Jenny Walker** (Ex Libris, Inc.) <jenny@exlibris-usa.com>

This is the second of a two-part article on the Metasearch Initiative of the National Information Standards Organization (NISO). Part 1, which appeared in February 2006, focused on the issues that prompted the creation of the Metasearch Initiative and reviewed NISO's plan of action. Part 2, reviews the first set of recommendations made by the NISO metasearch committees.

The **National Information Standards Organization (NISO)** established the **Metasearch Initiative** in late 2003 to enable metasearch service providers, content providers, and libraries to offer more effective and responsive metasearching. Part 1 of this article in the February 2006 issue of *Against the Grain*¹ introduced the initiative and its three task groups: Access Management, Collection and Service Descriptions, and Search and Retrieval.

All three **Metasearch Initiative** task groups presented their first set of recommendations at the **NISO Metasearch** workshop² held in September 2005 in Washington, D.C. The recommendations range from best practices to draft standards for trial use, **XML** protocols, and metadata element sets. All are designed to advance the use of metasearch technology and benefit both the providers and users of these services.

Authentication and Access Methods Recommendations

The **Access Management Task Group** issued a report on *Ranking of Authentication and Access Methods Available to the Metasearch Environment*³ in which they identified and evaluated 12 existing authentication methods, listed in Table 1, for their usability in a metasearch environment.

Innovations from page 73

the electronic environment as well. Many library users already expect the easy to use search box that they associate with many major search engines, and in the future they may come to expect features of **Ajax** in library related applications such as online databases, the library Website, e-journal portals, and the **OPAC**. With the growing use of **Ajax** on the Web, it is an innovative use of technology to keep an eye on in the future.

Table 1: Authentication Methods

Method	Description
Athens	an Access Management system for controlling access to Web-based subscription services
Cookies	a small bit of information that a Web server can store temporarily within a client computer for use by browser software
IP Filtering	a method for limiting access to a server based on the Internet Protocol (IP) address of the incoming connection
Kerberos	a network authentication protocol that utilizes secret-key cryptography
LDAP(Lightweight Directory Access Protocol)	an open-standard protocol for accessing X.500 directory services utilizing a simplified TCP/IP stack
NCIP (NISO Circulation Interchange Protocol)	a protocol for the exchange of messages between and among computer-based applications to enable them to perform the functions necessary to lend and borrow items, to provide controlled access to electronic resources, and to facilitate co-operative management of these functions
Proxy Server	an intermediary server that is used to provide additional security between a client and the end server by filtering or caching transactions in both directions
Referring URL	a method for enabling authentication based on the URL of the source which provided the link
Shibboleth	an architecture and policy framework for services to exchange information about their users in a secure, and privacy-preserving manner
SIP/SIP2(Standard Interchange Protocol)	a protocol to allow self-service machines in the library to exchange data with the library automation system
Username & Password	a method of authentication requiring the matching of a username with its associated password
X.509 Digital Certificates	a mechanism of utilizing public-key certificates for authentication

To evaluate these authentication methods, the group developed a comprehensive set of use cases that were then simplified to three metasearch specific cases: 1) In-Domain User;

2) Out-of-Domain User; and 3) Credentialed Access User. They also identified a set of environmental factors that are critical success factors in metasearch, as listed in Table 2.

Table 2: Environmental Factors Critical to Metasearch

1. Suitability/Effectiveness	7. Maintainability
2. Ease of Implementation	8. Robustness
3. Licensing Cost	9. Scalability
4. Implementation Cost	10. Simplicity of Understanding
5. Software Expertise Required	11. Acceptance/Preexisting Implementation
6. Security	

continued on page 75