

# The Google Web Toolkit Shines a Light on Ajax Frameworks

**Greg Goth**

**T**he Google Web Toolkit's public debut and rapid maturing have brought attention to the myriad tools and frameworks for facilitating Web applications, particularly those written with Ajax—Asynchronous JavaScript and XML (for more on Ajax, see the sidebar). One Ajax technology veteran, Charles Kendrick, says GWT's debut had significance for the industry far beyond Google itself.

"I think it was a lot like what Google Maps did for Ajax technology in general," says Kendrick, chief architect of Isomorphic Software, which develops the commercial Smart-Client Ajax framework. "It legitimized the idea of an Ajax component framework and made Ajax frameworks more visible. Google was out there saying, 'You don't want to roll your own.'"

Kendrick isn't alone in asserting that GWT's debut signals a new era of visibility for Ajax frameworks. Joshua Gertzen, lead developer for ThinWire ([www.thinwire.com](http://www.thinwire.com)), an open source Ajax framework, says GWT's release is a continuation of the Ajax trajectory Google followed with the creation of Google Maps and Google Mail.

"Something like GWT does have its merits, and I think Google is to be credited with bringing widespread Ajax adoption to the forefront," Gertzen says. "There's no question that Google Maps and Google Mail played a large part in that, a lot more so than the coining of the term Ajax itself."

## **New tool from deep pockets**

GWT (<http://code.google.com/webtoolkit>) debuted at the May 2006 JavaOne developers'

conference, and within hours some developers became true believers, while others took a more cautionary view of its Java-to-JavaScript paradigm. The fact that it came from Google, however, almost guaranteed close industry-wide examination and the subsequent spectrum of opinion.

"Everything Google releases gets a lot of scrutiny," says Bret Taylor, senior product manager for Google's developer group. "We love the feedback even if it's a little feisty at times. But I think really what I see is a lot of language warfare. Because it's a Java-centric product right now, folks who aren't big Java fans are probably not big GWT fans either. All these toolkits offer a lot of value, and it really comes down to the programming environment and languages you're most familiar with."

Taylor's observations perhaps illustrate the difficulty of empirically evaluating GWT and other Ajax frameworks side-by-side. Their capabilities are similar enough that they seem to be aiming toward the same end point—dynamic Web applications that are more easily developed. Those same capabilities are also divergent enough that getting a handle on what framework works best for which programming task is sometimes confusing. Rather than a clear-cut comparison of apples and oranges, Ajax developers face the subtleties of comparing what Kendrick concurs to be "apples and a different kind of apples."

ThinWire's Gertzen says you can roughly separate the current Ajax market into "two distinct groups—the public-facing applications and the business, enterprise-class applications. The public-facing applications have really gotten most of the limelight."

What could confuse developers trying to evaluate GWT's suitability, Gertzen says, is the seeming incompatibility between its development environment—pure Java, far more popular for server-side development than for Web applications—and its stated goal of helping developers more easily write Web applications with Java.

“GWT is interesting to me because it kind of wants to be enterprise, but it kind of doesn't,” Gertzen says. “GWT is using Java, trying to make it seem like it's a structured environment, but they also seem to be saying ‘We have no real significant components, we haven't thought through about what's needed to build more data-centric, data-input types of applications instead of Web pages types of applications.’ So I think there's still a little bit of room in terms of how things will pan out, but just the fact it has such a huge name behind it means it will be a key player.”

Isomorphic's Kendrick, too, says GWT's emphasis on Java seems incongruous considering the toolkit's likely environment.

“When we are in licensing discussions with ISVs [independent software vendors], other frameworks such as Tibco or Backbase will come up, but never GWT to date,” he says. “ISVs haven't shown much interest in it. To me, it's something that appeals to somebody who is already an expert Java developer and knows a lot of Java tools, which isn't your typical Ajax developer. A lot of Ajax developers like PHP or come from a Web-programming background. I think it appeals to those who see the necessity of Ajax for their applications but are a bit terrified of the browsers and JavaScript and that stuff, and GWT lets them stay on their home turf.”

Google's Taylor says the increasing complexity of Web-based application development will mandate that programming teams adopt a hybrid language approach—one that GWT is well suited to fill.

“For people who are very familiar with scripting languages, the prospect of going from a dynamic scripting language to a more strictly typed traditional language like Java might seem

## The Google Web Toolkit: Big Pea in the Ajax Pod

GWT might be the most visible manifestation of the Ajax framework, heralding more widespread use of the popular dynamic development model.

The term Ajax, for Asynchronous JavaScript and XML, was coined in 2005 by Jesse James Garrett, president of Adaptive Path, a technology strategy and design firm. Ajax development adds a layer of software in the traditional Web interface. But rather than slow down data retrieval, Ajax speeds it up by letting the application load only the updated elements of the desired Web page.

Ajax, as its name implies, isn't a single technology but rather a group of technologies that deliver a desired action. Its components include XHTML and CSS (cascading style sheets) for presentation, the Document Object Model for display and interaction, asynchronous data retrieval via XMLHttpRequest, interchange via XML and XSLT (Extensible Style Sheet Language Transformation), and JavaScript.

The performance premium that Web developers (and users) obtain with Ajax has been its main attraction. However, the intricacies of JavaScript development on the most popular browsers have caused many headaches; subtle incompatibilities from browser to browser delay application deployment. So, Google touts GWT as offering developers both the ability to compile Java code to JavaScript and browser-to-browser compatibility.

“The reason we released GWT was to improve the number and quality of Ajax applications,” says Google product manager Bret Taylor. “We run a fairly large search engine here and think what's good for the Web is good for Google. So we released it to give developers access to some of the knowledge and experience we've developed with Ajax.”

GWT is just one of dozens of open source and commercial Ajax frameworks. Its importance might increase now that it has become an open source project under the Apache 2.0 license. Already, developers are praising its ability to let them develop Web applications in tandem with tools from the open source Eclipse platform. With additional third-party tools and widgets being developed especially for GWT, its impact bears watching.

almost like going a little backwards,” Taylor says. “But the reason we released it is, when you're developing a really large product like Gmail or Google Maps and have a large development team, things like having types in your language, and well-defined interfaces, and modularity matter a lot more. In projects like this, you need to share code, automatically generate information about interfaces, and use code completion and things like that, that are virtually impossible with languages like JavaScript.”

### So who wants to play with it?

In the short time that developers outside Google have been using GWT, it has garnered kudos for several of its technical hallmarks. Taylor says two of

the things he's most proud of are the Java-to-JavaScript compiler and the amount of work Google developers have put into eliminating the idiosyncratic behavior that identical code often exhibited on different platforms before GWT's release.

“There were a lot of people in Silicon Valley who maybe were really familiar with the idiosyncrasies of JavaScript in browsers who felt very comfortable doing Ajax,” Taylor says. “And, I think there's also a class of professional developers who are familiar with languages like Java and were a little intimidated by the prospect of having to understand the idiosyncrasies of four different browsers and three different operating systems. So it's been popular with that group, which didn't want

# IBM and Universities in Open Source Collaboration

Terry Costlow

Industry-academia collaborations and basic research often get hung up by legal wrangling, but open source software's burgeoning popularity might break down some walls. In a move that some see as a boon for early-stage R&D, IBM has linked up with seven universities, pledging to make the results of joint projects available as open source software.

IBM and its university partners will be doing basic research in areas such as software quality, mathematical optimization, and privacy, exploring topics that have no commercial-product links or even expectations of short-term solutions. "These are long-term, evolutionary problems where we don't expect a single 'aha' moment to crack," says Stuart Feldman, IBM Research's vice president of computer science.

IBM's Open Collaborative Research program aims to simplify collaborative programs between the corporation and academia, eliminating intellectual-property rights by openly publishing results and offering them royalty free.

The company's partners are some of the leading research universities: Carnegie Mellon University, Columbia University, the Georgia Institute of Technology, Purdue University, Rutgers University, the University of California, Berkeley, and the University of California, Davis.

## Quick takeoff

Although the agreement is just a few months old, it's already bearing fruit. "This has opened up three-way collaboration. IBM T.J. Watson, Carnegie Mellon, and Purdue are working together on privacy and security issues. Researchers are very enthusiastic about how quickly things are happening," Feldman says. The three partnerships "cement our relationships," he added.

Feldman is bullish on the benefits, but he noted that the project holds some pitfalls. "There is the risk that we may give away something that's the equivalent of the patent for penicillin," he says.

The move is a big step for these universities and others that might follow their lead, and it's being viewed as a positive de-

velopment for the open source movement. IBM's move "recognizes open source as a legitimate licensing regime," says Michael Tiemann, a pioneer in the open source movement who's now Red Hat's vice president of open source affairs.

## Long-term focus

The move also underscores a shift from short-term, product-related research to investigations into long-term technologies. "IBM was hunkered down for a while, but now we're putting down a marker to say 'this is important,'" Feldman says.

As the software field evolves, leaders in academia and industry alike are beginning to focus more on end results, putting less emphasis on the software technologies used to provide answers. Open source software is increasingly becoming the base for solutions in many fields.

"The emphasis is shifting from software as being the key product of IT industry to information, knowledge artifacts, and services becoming the key product. Such a shift will result in new, challenging questions, such as the intellectual property of knowledge artifacts," says Elisa Bertino, a Purdue computer science professor.

As in other fields, early-stage research in software often yields results that can be years from becoming a marketable product. "We continually do research in computer sciences and software at a fundamental level. When we see an opportunity to put it into a new architecture or tool, we will make it a research project, though it may still be a couple years from being something truly useful to our company," says Rod Alferness, senior VP of research at Alcatel-Lucent's Bell Labs.

## Significant shift

If the move sparks a trend toward openness and collaboration, it will mark a significant change from the past couple of decades. In 1980, the Baye-Dole Act changed the way federally funded university research was handled, letting universities patent research results instead of the government.

to learn what comes down to memorization of esoteric knowledge, as opposed to programming methodology."

Software developer Rob Hanson, who was first intrigued by the Java-to-JavaScript compiler when GWT was released, has become a champion of the technology. He maintains the Sourceforge.net GWT widget library

(<http://gwt-widget.sourceforge.net>) and is writing *GWT in Action: Easy Ajax with the Google Web Toolkit* (Manning Publications), scheduled for release in the middle of this year. Hanson says while the compiler is indeed impressive, the ecosystem growing around GWT—including developers' tools such as the Eclipse Foundation's

Java debugger and integrated development environment—is becoming its true calling card.

"For me, the part that really makes it powerful is Java—not the language itself but everything that goes with it."

UK-based software development consultant Phil McCarthy, who wrote

Universities responded by patenting their IP in hope of turning a profit.

"Things were skewed after the Baye-Dole Act; every university was concerned about its own technology, and there was very little collaboration," says Mark Webbink, Red Hat's deputy general counsel.

As universities focused on patent rights, collaboration became increasingly more complicated. "It could take a year to get an agreement for a one-month interaction," Feldman says.

Universities might find that putting research in the open source world improves their bottom line. "Almost every university has lost money on licensing once you add in legal and management expenses," Feldman says. Some studies show the loss is as high as US\$40 for every dollar in revenue, Tiemann says.

Profit and loss is only one reason colleges and corporations collaborate. "It's important for us to interact with big companies. They have interesting problems to solve, and we need to make those relationships for our students," says Nila Bhakuni, Rice University's director of technology transfer.

Corporations, of course, like the fresh ideas outsiders provide, as well the comparatively low cost. "It's hard for companies to let freewheeling organic creative processes flourish. If they put in many rules, innovation withers," Tiemann says.

Still another benefit might come from students and faculty who decide to start companies. Open source software might open up opportunities.

"This gives students far greater opportunity without the roadblocks that come when code is closed and there are royalty issues with the universities," Webbink says.

### Fewer skirmishes

The open approach might also eliminate much of the infighting at universities, where debates over patent rights and royalty-free, open source software often get contentious. "In the computer science department, this is like a religious issue; a lot of strong dialogue goes on," Bhakuni says.

These issues also arise in corporations, even those such as

IBM that have embraced open source software as the basis for product development. Companies will always patent some technologies.

Although IBM is making a big commitment to opening up research, the firm won't stop working with universities to develop proprietary IP. "This won't replace the other things IBM does. If we want ownership, we know how to set that up, even if it's not always easy," Feldman says.

While open source software and globalization are both major trends in today's technical world, legal issues currently limit the Open Collaborative Research program to American schools.

"All our projects are with US universities so far; the lawyers had enough trouble getting things aligned here. We will look into other countries later," Feldman says.

However, Bell Labs facilities in foreign countries are doing a fair amount of collaborative research. "We have a lab in Ireland that has a university-government partnership. They also work with a consortium there, focusing on platforms and supply chain management," Alferness says.

Bell Labs also has a facility in China that partners with area universities, he adds.

Although many technical and legal issues are common to all universities and companies, the many trade-offs and individual views are perhaps more important than geographical concerns. These entities take different stances on IP licensing and the benefits of offering open source software based on their goals, which differ regardless of location. While some foreign universities freely work with corporate partners, some domestic universities are reluctant to partner with US companies to create open source software, still preferring to license their IP.

"A lot of universities have policies that are not very enlightened in terms of open source software," Webbink says. Change is coming faster at private universities than at many public schools because many public universities can't change financial policies without their state legislature's involvement, he adds.

an early review of GWT for the IBM developerWorks series ([www-128.ibm.com/developerworks/java/library/j-ajax4](http://www-128.ibm.com/developerworks/java/library/j-ajax4)), concurs with Hanson's assessment. He's most impressed with "the ambition and completeness of the GWT development environment. The real value of GWT is hosted mode (in which the application is run as Java

bytecode within the Java virtual machine and therefore able to take advantage of Java's debugging facilities and remain within an IDE) and the GWT shell, allowing you to interactively debug your entire Web-app codebase, client- and server-side, inside Eclipse."

Ironically, McCarthy wonders whether the very features that make

GWT attractive to developers comfortable with Java might also be its greatest downfall.

"I'm concerned that Java developers will plunge into GWT without some good basic knowledge of HTML, HTTP, the browser APIs, and JavaScript and that they are going to wind up in trouble when things go wrong," McCarthy says.

“Yes, when everything works as advertised, you can write GWT apps with little more than a knowledge of basic Java APIs. But there are plenty of things that can go wrong on the client side, especially in Internet Explorer, related to nasty, hard-to-trace subtleties like naming collisions between element IDs and built-in objects, overzealous security settings, caching problems, and so on.”

One of GWT's other outside champions is developer and ZDNet blogger Ed Burnette (<http://blogs.zdnet.com/Burnette>), author of *Google Web Toolkit: Taking the Pain out of Ajax* (Pragmatic Programmers). Burnette also says the seeming ease of coding in GWT might hide some problems.

“If there are any shortcomings, I'd say that GWT may make your job a little too easy, and you may get complacent,” Burnette says. “There are still some ‘gotchas’ to worry about—for example, CSS [cascading style sheet] differences between browsers—so you still have to test your application on different browsers to verify it looks and works okay.”

Another aspect of GWT that concerns both McCarthy and ThinWire's Gertzen is the potential security shortcoming.

“You can make the argument that GWT in its current state might scale on a public-facing environment to 100,000 users more easily than ThinWire,” Gertzen says, “because it resides more on the client, but you're exposing significantly larger portions of your application. GWT exposes multiple services and endpoint calls to the server, and each one is a potential security risk. ThinWire channels everything through a specific endpoint that only accepts requests that are valid based on the current state of the application.”

“Any code that executes in a Web browser can be tampered with, or bypassed completely, by a malicious user,” McCarthy wrote in his post-release review. “GWT provides a high level of obfuscation that mitigates the problem to a degree, but a secondary point of attack remains: any HTTP traffic that travels between your GWT client and its services.”

Google's Taylor says that, as Ajax technologies mature, security issues will be discovered and addressed over the entire ecosystem, including GWT.

“Ajax is really about using technologies in ways they weren't originally intended to be used,” he says. “For example, the HTML DOM [Document Object Model] is being used to display very graphically intense applications, whereas it was originally intended to display documents. And JavaScript is being used to implement very complex high-availability, high-performance applications, when it was really designed as a scripting language originally.”

“I think we are exposing a lot of security concerns in the process, and it will be an interactive process between the makers of toolkits and makers of browsers to find the right balance between user experience and security. All the toolkit makers have been doing a pretty good job of trying to reduce the possibility of introducing cross-site scripting attacks and things like that. There's certainly no easy solution for everyone, but everyone's aware of them.”

### **Culture, not binaries, might be the key**

If any one factor spells success for GWT, it could be when Google made GWT an open source project under the Apache 2.0 license in December 2006. Some developers' early criticism and hesitancy to use GWT were due to its original proprietary nature.

Taylor says the plan was to release it as an open source project all along, but the GWT team wanted to make sure the platform had reached a certain level of maturity before opening the source to a global audience.

“When we first released it, there was a lot of stuff under the covers we wanted to change very rapidly,” he says. “We knew once we released the source code, people would be building on it at different levels, and your ability to refactor the entire thing changes a little bit. We released Mac OS X support before open-sourcing it, and that was one of our bigger milestones. Once we did that, we were very comfortable having other peo-

ple build on top of the source code.”

Hanson acknowledges that developers were concerned with a void of information in the first few months GWT was available. However, he says that the GWT team has made few false steps in communicating with the developer community during the process of opening the project.

“They weren't ready for all this interest in the product, and there was a long delay between releases where there were a lot of bugs in the system,” he says. “People were not getting a lot of answers as to why the bugs and why so long between releases. Now that it's completely open, all the answers are there. We know what's going on on a daily basis, every bug that's being worked on, every patch. Beyond that, the community is spinning patches themselves. It's really becoming a true open source project.”

Isomorphic Software's Kendrick isn't sure GWT will emerge as a dominant player, even with its open source status.

“I would probably say the market will shape up into one or two major open source players, and if I were placing bets I'd say it would probably be Dojo (<http://dojotoolkit.org>), and then some commercial vendors at the high end,” he says. “I see GWT as a bit of reaction against how scary Ajax is, and I think that given the trend of Ajax tools rapidly getting better, and browser vendors paying more attention to developers' needs, the desire to stay in Java when working with Ajax is going to increasingly look like a reclusive reaction, and that will cause GWT to get a bit marginalized.”

Burnette says GWT will likely shake up the Ajax framework market to some extent but will probably be neither a behemoth nor an Edsel-like footnote.

“You can look at GWT one of two ways. Either it threatens all other Ajax frameworks because it largely obviates the need for them, or it's complementary to all other frameworks because, through the JavaScript Native Interface, you can make calls from GWT Java code into existing libraries. The truth is probably somewhere in between.”