

# Building Enterprise RIAs? Look Beyond AJAX

BY ROBERT H. HALSTEAD  
AND JNAN DASH

Get the most out of your Web 2.0 initiatives



*Robert H. Halstead, Jr., vice president and chief architect at Curl, Inc. spent more than 20 years researching programming languages and parallel computation in both academic and industrial settings. During his nine years as a faculty member at MIT he developed the “futures” programming language construct and implemented it in the parallel programming language Multilisp. Bert was a consulting engineer at Digital Equipment Corporation’s Cambridge Research Lab, where he guided early stages of the project that led to High Performance Fortran, a widely known parallel programming language endorsed by an industry-wide forum. His bachelor’s degree and doctorate are from MIT. rhh@curl.com*

*Jnan Dash is chief strategy officer at Curl, Inc. Jnan spent 10 years at Oracle Corporation and was the group vice president, systems architecture and technology until 2002. He was responsible for setting Oracle’s core database and application server product directions and interacted with customers worldwide in translating future needs into product plans. He played executive roles at IBM and currently sits on the boards of several new technology companies. jdash@sbcglobal.net*

With the growing popularity of consumer Web sites such as Google Maps, YouTube, and Flickr, the consumer sector is aggressively endorsing Web 2.0 technologies. Enterprises are increasingly recognizing the business value of the Web platform over the high cost of ownership and limited reach of existing client/server applications. The fastest way for enterprises to start reaping the benefits of Web 2.0 is via Internet (RIA) applications rather than wikis, blogs, tags, mashups, or social networks. RIAs offer the most immediate return on investment (ROI) for enterprises because the technology improves the user experience and lowers the total cost of ownership (TCO).

Enterprise 2.0 goes beyond the scope of just deploying Web 2.0 components. Unlike consumer-centric applications, enterprise applications must combine complex transactional capabilities with high scalability and extreme performance. While sites such as Google or Yahoo handle very large numbers of users doing mostly simple tasks, enterprise Web applications cater more to “power users” who need to interactively analyze and visualize large volumes of information stored in business-critical databases and interoperate with existing legacy applications. Whether created by porting existing applications to the Web or by starting fresh, this new generation of Enterprise 2.0 applications must deliver performance, processing power, and graphical sophistication equivalent to or better than what users experience with traditional client/server enterprise applications.

The widespread adoption of AJAX in the consumer space has made it an easy answer for business-to-consumer (B2C) application development. While adequate for the first wave of Web applications geared to consumers, AJAX doesn’t have the capabilities to meet all the requirements for the next generation of Web applications and more advanced software-as-a-service models. New RIA technologies have emerged to offer capabilities that go beyond the traditional functionality that

AJAX provides. As the RIA category continues to gain mindshare, lingering misconceptions about the range of available RIA technologies need to be addressed so companies and enterprise application developers can correctly answer their key question: “What exactly should we look for in an enterprise RIA platform?”

## Market Evolution

Figure 1 summarizes the evolution of enterprise software architecture toward RIA. Fortune 1000 companies, along with many others, spent a lot of resources during the 1980s and 1990s building client/server applications using the rich user interface of desktop clients such as Windows. In this way they were able to move beyond the limited user-interface capabilities of the earlier generation of “green screen” character-based applications. To enjoy the increased reach and lower TCO that Web 2.0 technologies make possible, enterprises now need to progress beyond this client/server model to the next evolutionary stage.

Starting with the introduction of the Internet browser in the mid-1990s, the pressure to move applications to the Web has built steadily as its advantages in reach and TCO have increased. Responding to this pressure, many enterprises and application vendors implemented Web 1.0 interfaces to substitute for the old client/server applications (Arrow 2) but the loss in usability was often severe. Now many people hope that RIA technologies such as AJAX, developed originally for consumer-oriented Web 2.0 applications such as Google Maps, can be used to restore the lost usability (Arrow 3), but applications that need to interactively manipulate large data sets and provide complex user interfaces are just too hard to build this way. Many of these applications were too sophisticated to ever make the jump to the Web at all. What they need is an RIA framework powerful enough to support applications with the full richness, interactivity, and performance of native client/server applications (Arrow 4).

Applications in the upper-right quadrant of the chart represent a convergence offering the best features of both the client/server paradigm (notably the interactivity and richness of the user experience) and the Web paradigm (notably the broader reach and reduced TCO). Client-side platforms that enable such applications have been referred to as “Fit Clients” since they combine the virtues of “fat clients” (client/server) and “thin clients” (Web 1.0).

### Enterprise RIA Requirements

Enterprise application interactivity and performance is not just pleasing to see: it has a direct impact on the bottom line. Employees, partners, and suppliers are more productive, less error-prone, and make better decisions when they can use an application that lets them perform a task with an intuitive, visually rich interface with a minimum of mouse clicks and waiting time. Thus, a rich user experience that seamlessly includes complex graphics (even 3D), animations, and drill-down has real economic value. In this era of malware and Trojan horse attacks, coupled with online storage of increasing amounts of sensitive information, security is also a paramount concern for enterprise IT departments as well as their risk and compliance officers. See the sidebar for a summary of these enterprise RIA requirements.

### Scalability & Performance

The RIA usability requirements mentioned require a platform that has high performance on the client side and the scalability to perform large computations on large data sets. Enterprises have spent decades fine-tuning the performance of their applications and they (and their customers) cannot afford to lose this performance in exchange for the benefits of Web deployment, so latencies must be minimized for sub-second response times in most cases.

High throughput and fast response are the two critical metrics for transactional systems. These goals can be achieved through a combination of using a high-performance RIA platform and choosing a suitable application architecture. At the application architecture level, the division of work between the client and the server must be carefully architected to minimize round-trips, with the client side assuming most of the responsibility for user interaction and data caching. Solutions designed this way will be more scalable, which is important because the number of concurrent users can grow rapidly, especially in outward-facing B2B applications, as partners, suppliers, and buyers join the user community.

This type of application architecture is feasible only if the RIA platform is powerful enough

to operate on large volumes of cached data with high performance. Scripting engines such as the browser JavaScript interpreters used in the AJAX approach fall significantly short in this regard. While these engines may have built-in “canned” primitive functions that can be used to perform certain preprogrammed operations such as XML and JSON parsing, application-specific data analyses generally have to be written using the scripting language even for “inner loop” computations, suffering a major performance cost. This requirement has catalyzed a trend toward just-in-time (JIT) compilation technology that can optimize scripting performance on-the-fly. This technology is deployed in Adobe’s Flash and is starting to show up in browsers as well, with Google Chrome leading the way. Some RIA platforms, such as Curl, do not merely optimize the performance but actually compile applications to high-quality native code comparable to that produced by C and C++ compilers, yielding further significant improvements over those that have been achieved via JIT optimization of scripting execution. Enterprise application developers should evaluate their needs for client-side data processing and ensure that the RIA platform that they choose has the horsepower to do the needed computations with acceptable performance.

### Security Standards

Security is also a key criterion. Enterprises typically need to store large amounts of sensitive data whose disclosure could expose the enterprise to legal liability or give competitors a leg up. Every enterprise application, including RIA, has to address data and application protection via suitable encryption and authentication strategies. An ideal RIA will follow the principles of service-oriented architecture (SOA), where server-side functions

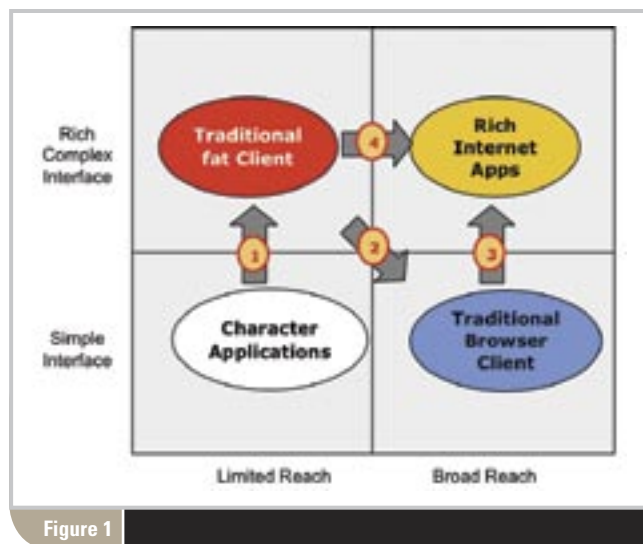


Figure 1

## Key Platform Requirements for Enterprise RIA

- Displays complex graphics and reports for users
- Handles large data sets
- Ensures very high performance
- Offers very high scalability
- Addresses strict security requirements
- Provides Fit Client architecture for online/offline operation
- Follows SOA and standards
- Facilitates migration from legacy apps
- Enables platform independence
- Includes rich development tools
- Eases manageability

are packaged as services and the client-side RIA acts as a consumer of those services. Historically, SOA discussions have centered strictly on server-side component assembly, with the consumers of services being other server-side components, but RIA technology allows the construction of clients that are powerful enough to consume the services directly, which is a more natural arrangement. Of course, client-side support for SOA standards and paradigms such as SOAP, WSDL, and REST is a prerequisite for smooth SOA interoperability.

### Online/Offline Capability

Increasingly, enterprises need to offer offline access to their applications for road warriors such as sales agents. To do this, enterprise RIAs must have access to a local persistent data store for offline processing that can be synchronized with the server when network connectivity is restored. Traveling salespeople can use this feature, for example, while visiting a customer, entering data into the client-side persistent data store that is later uploaded to a server. This kind of Fit Client architecture is very different from the browser-

centric thinking of the past, with its need to be tethered all the time to the Internet. The need for this sort of occasionally connected computing is growing and is essential for the next generation of enterprise RIA.

### Rich Development Tools, Platform Independence

To improve developer productivity, other important metrics to consider when choosing a technology for enterprise-scale RIA include the availability of rich development tools, the time it takes to build an application, lines of code, functionality, and ease of maintaining and extending applications. Since enterprise applications are often large, with dozens or more screens, the ability to manage large-scale code development with good modularity constructs, compile-time checking, test coverage analysis, and automated testing support is also essential. On top of that, tool suites must provide functions for performance monitoring and tuning. For many enterprises, flexible deployment and platform independence are also requirements.

### Evaluating RIA Solutions

A "one-size-fits-all" approach isn't wise when evaluating RIA technologies: some RIA platforms are better suited to B2C applications while others work best for complex enterprise applications. Consumer-grade RIA technologies aren't designed to support enterprise-class, data-intensive desktop applications. There's a substantial difference in both performance and ease of development depending on the platform chosen.

An independent benchmark study done last year showed how different RIA technologies vary in their capacity to meet the requirements for enterprise RIA. Conducted by Sonata Software, a Bangalore, India-based global systems integrator and solution provider, it evaluated Adobe Flex, Microsoft ASP.NET AJAX, and the Curl RIA platform based on four key categories: ease of learning, ease of design, runtime performance, and ease of development. The results found each platform to have unique advantages, with some better suited to B2C applications and others to enterprise applications.

The study's methodology was to implement the same modest-sized application in each of the three technologies studied, measuring various metrics in the resulting application as well as in the learning and development process. The study was time-boxed, with about two weeks allowed for developing the application in each technology. Several interesting trends emerged across the technologies. Various specific strong or weak points of the various technologies were also noted. The results of the study are further discussed in Figure 3.

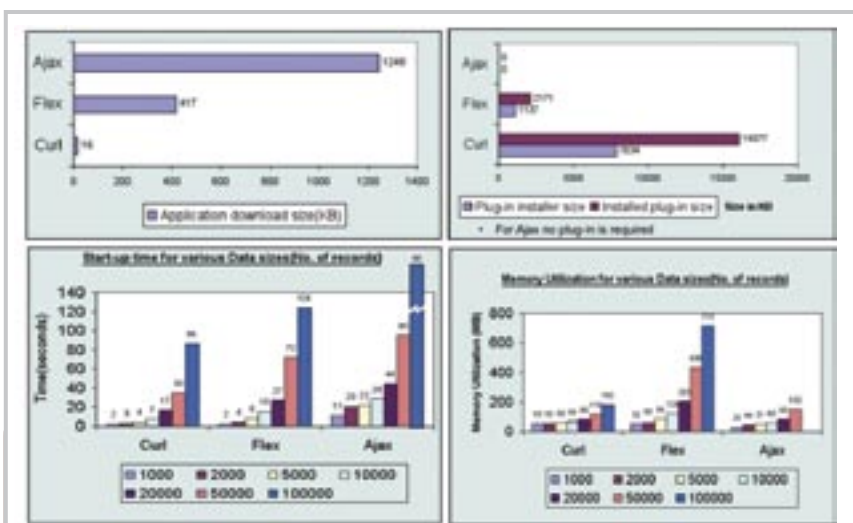


Figure 2

## Enterprise RIA Conclusions

Companies recognize the Internet as an essential service platform for a new class of highly interactive applications. The market for RIA continues to grow as companies experience significant savings in cost and improvements in their processes by moving to Web 2.0. In a recent April 2008 industry report by Forrester Research entitled, "Global Enterprise Web 2.0 Market Forecast: 2007 to 2013," G. Oliver Young predicts \$4.6 billion in spending by 2013 as large companies embrace Web. 2.0.

Today, there is a spectrum of choices for developing Rich Internet Applications. However, the reality for enterprise RIA is that the true choices are far fewer – as the requirements for building enterprise-scale rich Internet applications are more stringent than those that consumer-grade RIAs can support. At a minimum, any RIA platform for Web-enabling highly interactive client/server applications must be able to deliver high-performance processing, handle large data sets, support rich graphical content, and enable online/offline functionality. Depending on the type of application, ranging from simple B2C applications to complex enterprise applications, developers must match their application needs with RIA platform capabilities to get the most out of their Web 2.0 initiatives.

## Resources

- RIA Development Platforms: Curl, Flex and Ajax – A Comparison, Sonata Software, Fall 2007
- Web 2.0 Principles and Best Practices, Conference Presentation, Fall 2006, John Musser, with Tim O'Reilly and the O'Reilly Radar team. Published by O'Reilly Media, 2007, 100 pages.
- Measuring Rich Internet Applications, Forrester Research Report by Megan Burns, February 7, 2007
- Global Enterprise Web 2.0 Market Forecast: 2007 to 2013, Forrester Research Report by G. Oliver Young, April 2008

## Results of the Sonata Study

One interesting trend was the tradeoff between plug-in installer size and the size of the downloaded application. Paying the one-time cost of installing a larger plug-in that contains a richer set of APIs enables the subsequent application downloads to be dramatically smaller. At one extreme, AJAX required that no plug-in be installed, but the application was over 1.2MB because it needed to include the implementation of many library APIs, such as buttons and data grids. At the other extreme, the Curl installer was about 7.8MB, but then the Curl application was only 16KB in size – almost 80 times smaller – because the buttons and data grids are implemented in the plug-in. Flex fell between these two extremes (see Figure 2).

Similar variations, though less dramatic, were seen in performance measurements. The test

# “Developers must match their application needs with RIA platform capabilities to get the most out of their Web 2.0 initiatives”

applet begins by loading a large data set whose size can be tuned to observe performance trends. AJAX took the longest time to load the data sets, even failing completely when asked to load 100,000 records. Curl averaged nearly three times faster for the various data set sizes, with Flex falling in between. Curiously, Flex required much more RAM to store the data than either Curl or AJAX (see Figure 2).

Finally, the technologies were compared to gauge their ease of learning and ease of development and maintenance, with the number of lines

## RIA Vendor Checklist

Some of the top 10 questions companies should ask RIA vendors include:

1. Do you have enough functionality for creating dashboards for BI applications?
2. Can you construct transactional applications, similar to what we have in client/server today?
3. Do you provide functions such as drill-down, mouse-over pop-ups, and a rich library of charts and graphs?
4. Do you have just-in-time compilation at the client for super-fast performance? Otherwise, how do you minimize latency from the round-trips?
5. Can you run these applications offline for subsequent synchronization when connected? What's your data persistence approach at the client?
6. Do you have high-class IDE support for fast programmer productivity?
7. Do you have scalability? (i.e., no performance degradation with growth in users and workload)
8. Do you provide enterprise-class security? (i.e., sandbox, secure access to resources)
9. Can you handle a large volume of data with good performance? (i.e., 100K records processed at the client side)
10. Can you fit into the back-end ecosystem? (i.e., J2EE, Oracle, DB2, Weblogic, Websphere)?

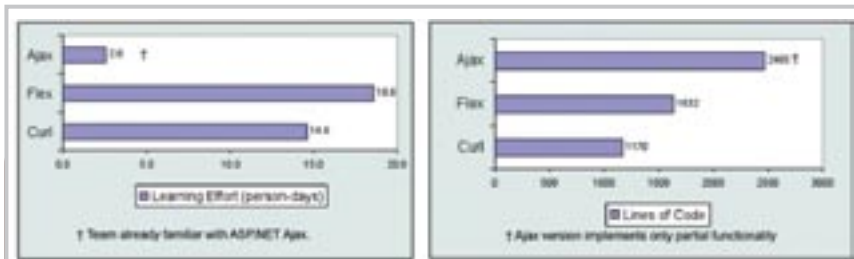


Figure 3

of code serving as an indication of the latter. AJAX was not directly comparable to the other technologies along these axes for a couple of reasons. First, the study team was already fluent in ASP.NET AJAX so did not require much time to “learn” it. Second, they weren’t able to complete all the features of the test application in AJAX in the time allotted. Curl scored best in terms of the number of lines of code, with Flex requiring 50% more code and even the partially implemented AJAX application requiring 2.5 times as much code as the Curl application. Curl was also rated superior to Flex in learning effort, requiring about 20% less time (see Figure 3).

### AJAX Geared to B2C Applications

ASP.NET AJAX emerged as the platform on which application development is the most expensive (judging by the number of lines of code that have to be written and the absence of key APIs such as chart controls), runtime performance is the lowest, and the application download size (and hence the loading time) is the highest. AJAX also lacks an offline execution capability. The great advantage of AJAX is that it doesn’t require any plug-in on the client machine in advance. This combination of properties points strongly toward the use of AJAX for B2C applications that don’t have very complex behavior (so the number of lines of code is tolerable) and don’t need to work on very large data sets. Many B2C applications fit into this category, and the fact that AJAX requires no software installation is a perfect match for the “casual” aspect of many B2C applications, where there isn’t a deep pre-existing relationship between

an application provider and an application user that can serve to motivate installing client-side software in order to use the application.

### Flex – Most Appropriate for Media Applications

Adobe’s Flex platform, as well as the accompanying development tools, stands out in their support for animated user interfaces and streaming media delivery. Since Adobe’s Flash player is already a widely installed browser plug-in, Flex is perceived as not requiring users to do a new plug-in installation and so is perceived as suitable for casual B2C applications. This combination of features makes Flex an attractive choice for media-intensive uses. The full Flex offering includes server-side components that streamline the process of connecting to Web Services and media sources, as well as client-side libraries, including attractively designed charts and data grids. However, Flex didn’t have an API for 3D visualization and Flex’ scripting-based programming model demonstrated weaknesses in performance and memory usage for handling enterprise-scale data sets on the client side.

### Curl Designed for B2B Enterprise Applications

The study found that Curl is most appropriate for moving internal enterprise applications (“business-to-employee” or B2E) and enterprise-class B2B desktop applications to RIA. The example application was quick to download and fast to start up. It offered good performance on large volumes of data and the Curl tool set included excellent performance analysis tools, supported multiple programming paradigms (visual drag-and-drop as well as code-based) and offered quicker edit-debug cycles thanks to the absence of any intermediate build step to translate source code into bytecode. The user interface responsiveness was good. The sample application required substantially fewer lines of code than AJAX or Flex. The learning curve for Curl also measured as competitive with, and in fact slightly superior to, Flex’ learning curve. ■



### Curl, Inc.

One Cambridge Center, 10th Floor  
Cambridge, MA 02142-1612

Telephone: 617-761-1200

Fax: 617-761-1203

E-mail: info@curl.com