

The SOA Magazine

Feature Article



AJAX: Bringing SOA to the Front Lines

by Mathew Quinn

Published: January 3, 2007 (SOA Magazine Issue III: January 2007, Copyright © 2007)

[Download this article as a PDF document.](#)

Abstract: A service-oriented architecture (SOA) can provide enterprises with significant benefits, including the ability to reuse application functionality and to interconnect heterogeneous applications to create new composite ones.

However, a critical component to the realization of SOA benefits is that users throughout the extended enterprise can efficiently access and interact with key resources. Otherwise you cannot fully leverage your infrastructure investment. Using AJAX rich internet applications (RIAs) as the presentation tier, however, can significantly enhance the impact of SOA.

This article explains how companies can link their employees, customers and partners, with a scalable, flexible interface to efficiently interact with service-oriented resources.

Introduction

"In order to maximize the vast potential of SOA, companies need to consider the best user-interface for the job."

Other than death and taxes, there's one other item that can be pegged as a sure thing: the staggering business value of the information that lies in a company's IT infrastructure. In today's fast-moving business environment, the ability to make smart, agile decisions requires unfettered access to that information and information context, no matter where it lies amongst the tangle of silo'ed applications and disparate platforms that make up a corporate infrastructure. To accomplish this, CIOs must find a way to unfetter the flow of business information throughout their IT infrastructures. The potential pay-off for success is huge, so it's not surprising that it has long been the Holy Grail of information system integration efforts.

But until recently, such integration has been much easier said than done.

It's not through lack of effort—technologists have tried many methods of solving this particular puzzle, though none lived up to their hype. But the latest buzz surrounding service-oriented architecture, or SOA, might finally herald a concept that can get the job done. More and more, CIOs are turning to SOA as a solution to the integration woes that continue to plague their information infrastructure. SOA provides the enterprise with the ability to reuse application functionality and to interconnect heterogeneous applications for the purpose of creating new, composite ones (as illustrated in Figure 1).

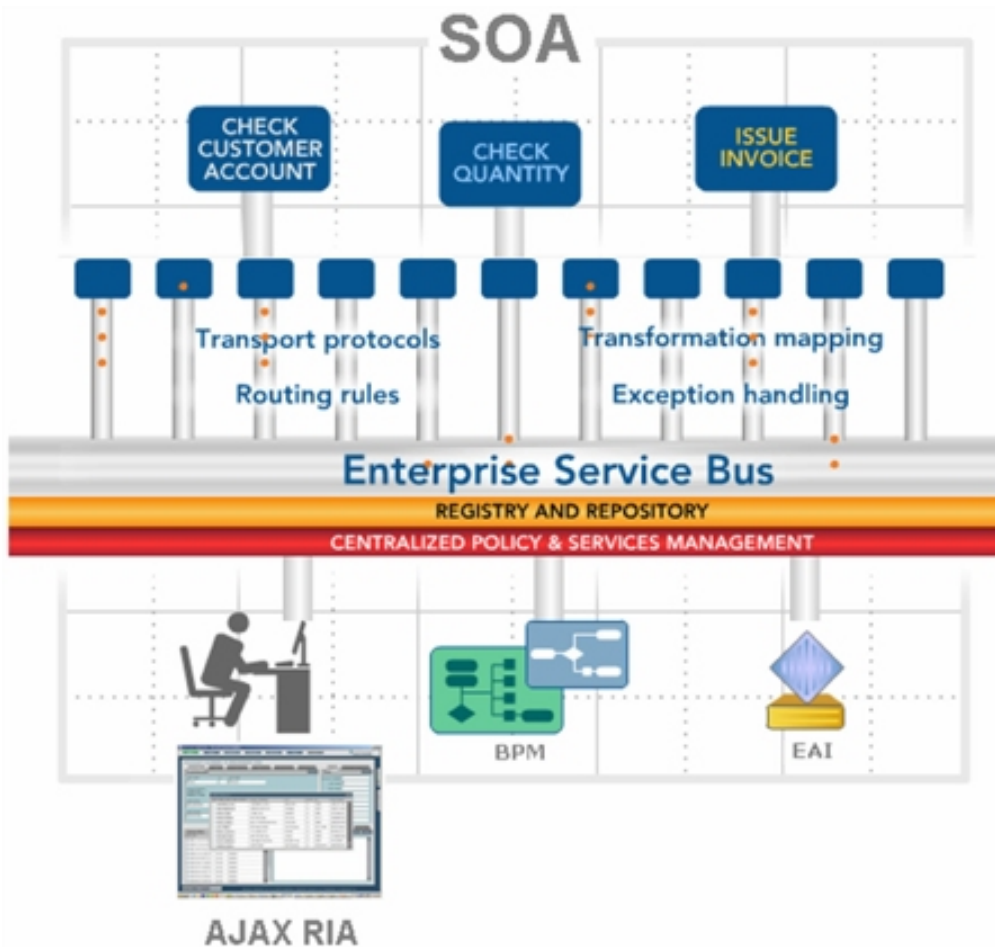


Figure 1: An SOA implementation, as managed by an Enterprise Service Bus, enables multiple solutions to be composed including AJAX Rich Internet Applications, Business Process Management, and Enterprise Application Integration.

A recent report from Forrester shows that satisfaction runs very high, with nearly 70% of SOA users saying that they will increase their use. The value of SOA is being validated and, if Forrester's sample is typical, will soon become the prevalent style of information technology architecture for the enterprise. The question is, how can companies attain SOA's full potential?

The Value in SOA

There are many ways of describing SOA, from the highly theoretical to more business-oriented definitions. In a nutshell, SOA is a technology agnostic architectural style—based on the concept of a service—that enables business agility. Both providers and consumers of services can interact in a loosely coupled manner via a consistent and neutral interface contract.

One of the biggest differentiators between SOA and architectures that have come before it is that the central concept of “service” can be defined along a spectrum of complexity, granularity and business focus. A service could represent a low level technical action such as a ‘log to file’ or a highly business focused and complex ‘provision customer’. This flexibility enables business and IT to communicate ideas and architectures that are consistent with the enterprise’s culture – and not be constrained by a single technology platform.

Because the services involved are platform and language independent, they can pull information to support business processes regardless of the underlying platform or application. SOA allows an organization to break down monolithic applications into discrete standards-based business services that can be shared and reused across the enterprise. By doing so, users can access information regardless of where it lies, or what platform or applications it resides in. By turning enterprise resources into services, they are easily accessible, easily reused, and easily integrated with other resources. Moreover, because enterprise class service-oriented architectures are event-driven, they can respond to asynchronous events as they happen, rather than more typical synchronous request/respond message exchange patterns.

Asynchronous events have causes that may lie outside the realm of the business or the IT implementation but which have effects within the business architecture. For example, a customer call may initiate a workflow, a stock split may

initiate a series of database updates and a disk drive failure may trigger a disaster recovery process. In this respect, events are really first-class citizens in an SOA.

The benefits to such a service-oriented architecture approach are considerable. Because a SOA is event-driven, companies can build services that automatically respond to certain asynchronous business processes. For example, a company could build a service that automatically activates extra customer care for orders over a certain monetary level. Services can also be reused over and over, which dramatically cuts development time, allowing companies to get to market faster. And the loosely coupled relationship between the services makes an SOA-based platform very adaptable, enabling the enterprise to swap services out as needs change. All in all, these add up to a very flexible architecture that is able to integrate information from sources across the enterprise. As such, SOA allows technology to quickly and flexibly adapt to meet the changing needs of many different kinds of business users.

Companies that have embraced SOA have reaped considerable business benefits from the results. Take the example of Con-way Transportation Services, who implemented a multiyear SOA project aimed at integrating every aspect of freight operations. As a result of this effort, Con-way operations personnel in over 400 service centers, along with sales personnel, customer service representatives and truck drivers have access to timely, accurate freight movement and status information. SOA has helped the company reduce its time to market by more than 40 percent for new business initiatives through reuse of existing services. There is zero latency in the flow of data, which gives real-time decision support to Con-way personnel. And event-driven integration with US/Canadian Customs processing has reduced border crossings from 2-3 hours to just minutes. In a business where time is money, that is a serious ROI.

The Role of AJAX in Enterprise SOA

Clearly, SOA is the undisputed hot ticket in IT trends. According to "Enterprise Service Bus and SOA Middleware," a recent report from Aberdeen Group [REF-1], nine out of every 10 companies surveyed are either implementing or have implemented service-oriented architectures. But there are some remaining hurdles that prevent companies from achieving its full benefits. One of the key challenges is how companies expose services to the people who will ultimately use them. Organizations need a usable front-end, or interfaces, that allows business users to fully exploit the full benefits of a service-oriented framework. This is no small issue, as it is critical for users across the extended enterprise to be able to easily access and interact with the resources within an SOA in order to make a difference in how they work. For that, they need a user-interface that does what they need from a business standpoint, but is also flexible and cost effective from an IT standpoint (Figure 2).

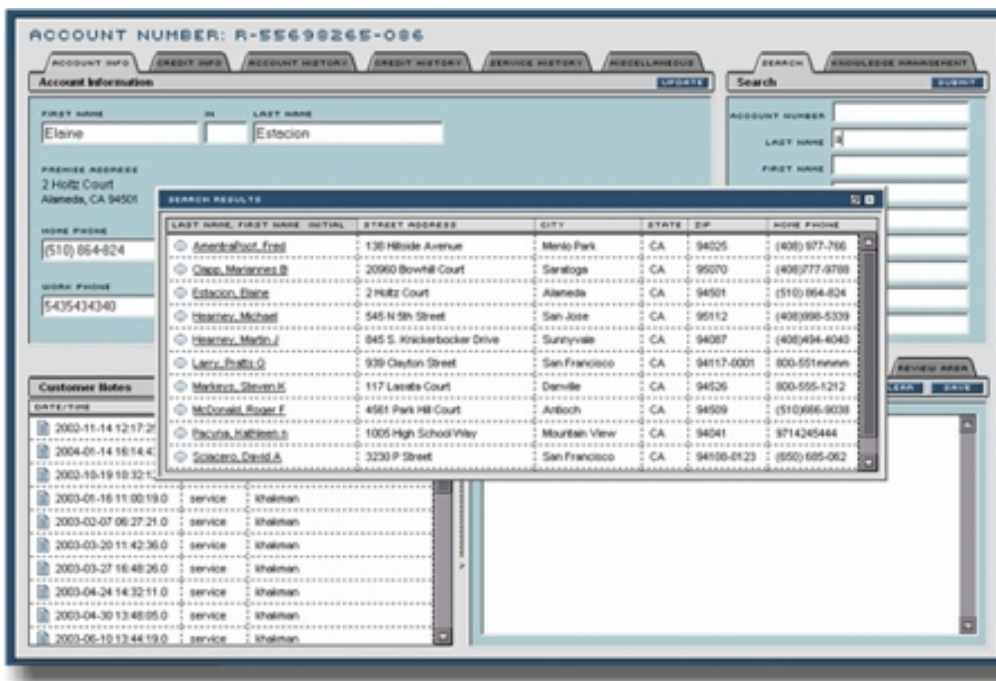


Figure 2: Integrated Call Center Customer Service Representative screen provides a 360-degree customer view by compositing ("mashing-up") information from systems exposed across an SOA, resulting in a zero-footprint, high productivity, browser-based application.

Right now, the common choices are desktop-based GUIs and Web browsers, and both have drawbacks. Traditional desktop-based user-interfaces are deep and interactive, but are expensive and time consuming to deploy and maintain. Moreover, such software is not compatible with all platforms, and not all users of SOA services will reside

within the enterprise. As such, many may lack the resources to use such an interface.

Web-based interfaces require no client installation, and access is available anywhere the Internet goes. It's also relatively simple and cheap to maintain. However, such interfaces frequently lack the interactivity and robustness of desktop interfaces. As companies move beyond initial, simple SOA projects into more complex services that require more interactivity, this will be a problem. For example, companies are now developing composite applications based on a number of services drawing from different back end applications. The best presentation options for a composite application is a browser with rich capabilities, but historic attempts to deliver more interactive browser interfaces have required supplemental browser technologies, which come with their own challenges.

Faced with this issue, companies are turning towards AJAX rich internet applications (RIAs) as the user presentation tier for SOA. While this technology has hit the mainstream market through such well-known RIAs as Google maps, its value to the enterprise is very significant. David Linthicum, co-author of "Real-World Ajax", agrees that AJAX is not only becoming the de facto standard dynamic interface for the Web, but it's also a good fit for SOA. He writes, *"In essence, AJAX provides edge technology for SOAs, or the top layer of technology dealing with the user-interface. See, AJAX can extend visual service to a true interactive dynamic interface that's more attractive and functional for the end user."* [REF-2]

Building rich internet applications using AJAX gives users a simple front-end that combines the best of both browser and desktop based GUI technology. The interface layer exists to make services available to users, and AJAX can directly access services through its asynchronous devices and send the pertinent information back to the user.

Moreover, AJAX RIAs solve these issues by utilizing capabilities already existing in the browser, and don't require add-on technologies such as applets or plug-ins. Their flexibility facilitates truly user-centric interfaces that can more closely match user requirements, because they aren't constrained by the inherent limitations of server-generated HTML pages. As a result, companies using AJAX RIAs can pull together series of services to create more complex and interactive applications and processes.

In essence, this is the next step in the thin client vs. thick client debate. With SOA, the clear heir should be the rich client, which can access and integrate information from back end Web services, and pull that information into a presentation that looks like a native application. Using rich client technology within the SOA framework allows better, more interactive interfaces that can exploit the full power of a SOA approach.

Let's take the example of a financial services customer service representative (CSR) trying to answer a client's question about his account. Using AJAX RIAs in front of a SOA framework, the company could write services that automatically pull data from the customer's savings, checking and mortgage and retirement accounts. Data services can then pull pertinent historical information about the customer from customer lists, while another service can run a risk assessment against that data. All of this could then be aggregated and presented in one application interface that the CSR can manipulate and interact with, rather than facing a static screen of data.

What to look for in AJAX RIA development tools

AJAX is a fairly recent appearance on the Web development scene, and even newer to enterprise IT shops. That said, there are a few tools targeted towards corporate use that are beginning to appear. Here's what to look for with this class of tools:

- Look for tools that work with existing Web browsers.
- These tools should not require the installation, authorization, or downloading of any plug-ins, applets or desktop executables, because that would defeat one of the key benefits of an AJAX RIA – high availability and no client installation.
- They should be able to work with existing web application servers and other HTTPS accessible sources in order to leverage existing infrastructure resources.
- Look for features such as Drag-and-drop visual authoring tools for standard XML and XSD, SOAP, and WSDL communications, as well as HTTP/S GET and POST operations.
- In order to reduce development time and flatten the developmental learning curve, these tools should make use of familiar technology such as traditional APIs, as well as reusable components.
- As this will be a learning experience for most IT staffs, make sure that the tool you choose comes with good support, particularly online resources and developer communities.

Today, this is called a “mashup”, but these types of composite applications have been a mainstay of those pursuing SOA before the Web 2.0 buzzword machine kicked in.

A Case Study: Iconix Pharmaceuticals

Take the case of Iconix Pharmaceuticals, a pioneering company in chemogenics, which uses chemistry and genomics to draw insights from the mountains of data required to bring drugs to market. In order to succeed, Iconix must be able to integrate data that typically exists in separate databases in order to analyze it and find compelling results. If that information is not available from a single source, it becomes exponentially more difficult to make the connections that will lead to important discoveries. Iconix used an AJAX RIA to serve as a front-end for the company’s DrugMatrix software, which delivers its chemogenics database. The GUI uses standard browser technology, but can consolidate reams of data and present that information in numerous ways. Using AJAX to develop this GUI has allowed the company to integrate more than 200 million complex data points in a visually rich and interactive environment (much like the one shown in Figure 3).

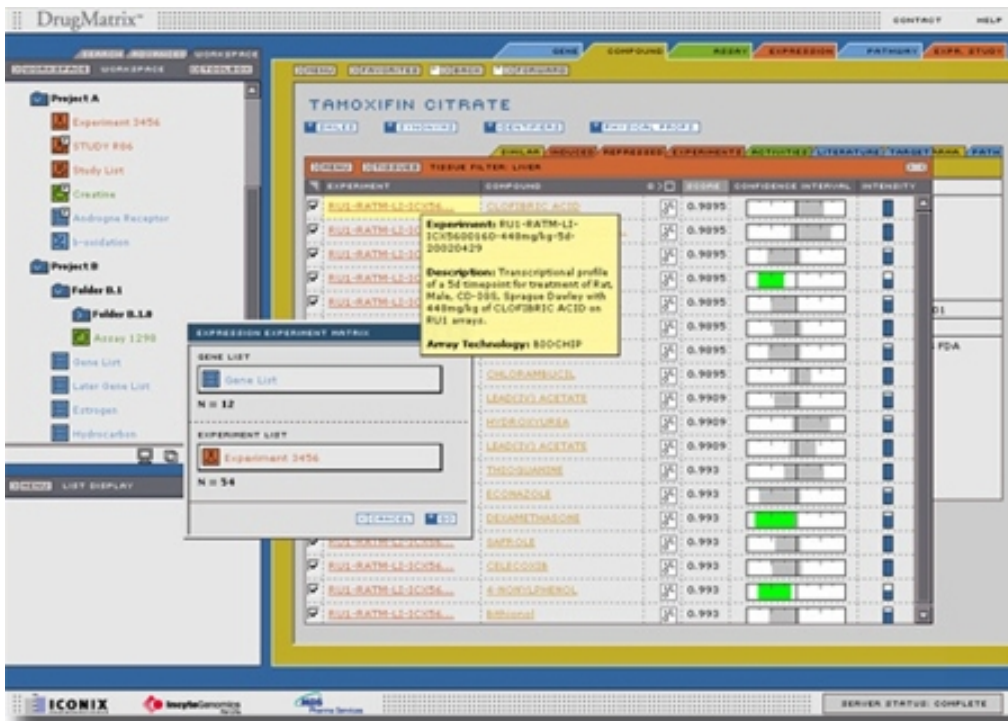


Figure 3: Iconix Pharmaceuticals’ DrugMatrix solution exposes more than 200 million drug research and related gene pathway data points through simple XML-driven services, and makes that information accessible to scientists through an Ajax RIA user-interface which interacts with those same services.

Iconix’ implementation is a service-oriented solution with simple XML services – not full scale SOA, but a good evolutionary path for getting started with services. As companies move beyond this initial stage into more mature SOA solutions, the use of services will become more complex and interactive. We can go back to Google maps to illustrate this. Behind its rich interactive AJAX GUI lies a mapping service, which enables the plethora of reuses into “mashups”. The reusability of the service and the richness and reusability of the GUI together are a great example of the power of service-orientation. AJAX RIAs used within a service-oriented architecture scheme offer tangible benefits on the IT side as well, such as the following:

- Development times and costs are slashed dramatically, as companies take advantage of the reusability of services as well as the ability to avoid the problems inherent in thick client deployments.
- The ability to use the same interface technology regardless of user location or platform. Since AJAX is platform agnostic, it works regardless of what the user is on, be it Windows, Linux or the Mac. This makes deployment much simpler, as well as guaranteeing that the entire user population has access to the same rich interactive services.
- Companies implementing AJAX in an SOA framework can finally use the full interactive capabilities of services without having to deal with the constraints of Web browsers. AJAX RIAs provide Web-based applications that function as well or better than traditional desktop interfaces.

Conclusion

In the final analysis, SOA projects are attracting so much attention because they finally offer a realistic framework for allowing an enterprise to make full use of the treasure trove of data that resides on its IT infrastructure. The notion of getting fast and easy access to real-time, accurate data across the entire enterprise is certainly enough to grab any CIO—or CEO's attention. But while SOA has the promise, don't forget about running the final yard—getting the information from the services to the user, and doing so in a way they can fully exploit. In my opinion, AJAX rich application interfaces are clearly the best tool out there for putting together a front-end that lets companies really go to town with a service based approach to integrating online information.

References

[REF-1] "Enterprise Service Bus and SOA Middleware", Aberdeen Group, June 2006, (available at http://www.aberdeen.com/summary/report/benchmark/RA_IT_ESB_PK_3170.asp)

[REF-2] "Real-World AJAX: Secrets of the Masters", Dion Hinchcliffe et.al.; ISBN 0977762203

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL



[home](#) [past issues](#) [contributors](#) [official book site](#) [legal](#) [rss](#)

Copyright © 2006-2007 SOA Systems Inc. All Rights Reserved