Happy Birthday, Ajax4jsf! A Progress Report

By Max Katz, Senior Systems Engineer, Exadel

Ajax4jsf is turning one soon and what a year it will have been. It was an amazing ride for all of us here at Exadel. The popularity of Ajax4jsf is evidenced by the size of the community, available resources, examples, blog entries, and articles. Also, a significant number of organizations today use Ajax4jsf in production.

Ajax4jsf is a popular open source extension to JavaServer Faces (JSF) that adds AJAX capability to JSF applications without requiring the writing of any JavaScript. Ajax4jsf is also a year old now. In terms of comparable years for a person's life, that's quite a long time, so, after wishing Ajax4jsf and those who have worked on it a heartfelt "Happy Birthday," let's take the time to reflect on this whirlwind of a year.

The Road to Ajax4jsf

First, we'll look at the technological situation preceding the birth of Ajax4jsf. In doing this, we'll look at the "jsf" piece, then the "Ajax" piece, and finally the "4" piece.

JSF

JavaServer Faces is a standard component-based user interface framework for building Java-based Web applications – with the emphasis on *component-based*. It's now about three years old.

In the past year, interest in JSF has dramatically increased. At Exadel, we've seen one example of this in our own experience with users of our IDE. Exadel provides an Eclipse plug-in for visual and source development that includes support for JSF, Struts, and Hibernate. In the past year, roughly 90% of all downloads of the plug-in were for JSF development.

It's not surprising that JSF has become so prominent. First, it's a standard, developed through the Java Community Process, and a part of Java EE 5. Second, and more importantly, it brings component-based development to the Web. As a developer, you design the application from reusable user interface components –

those of you with Swing or ASP.NET experience should be familiar with this approach. These UI components, also known as controls, can be as simple as a submit button and as sophisticated as special trees, tables, and menus. A quick search on the Internet will provide abundant evidence for why JSF is the best technology to use today for building Web applications.

AJAX

In the past couple of years, a great deal of buzz has emerged over - AJAX. It's not about the cleaning stuff (or the soccer team from the Netherlands or the ancient Greek hero). In this case, it stands for Asynchronous JavaScript and XML. AJAX is basically a collection of existing technologies for radically changing user experience on the Web.

To create this richer user experience, AJAX lets you create rich Internet applications. These are applications that are rich, interactive, and fast. The basic technique behind rich Internet applications is that only the part of the page that has changed should be updated. There is no need to reload the whole page. Such applications sometimes are referred to as Web 2.0 applications. (I'm sure you have heard this term already.)

Rich Internet applications have grown tremendously in popularity in 2006, especially on the consumer side. Today, thousands of consumer Web sites use AJAX functionality in one form or another. (Google Maps is probably the most popular one.) Enterprises have now noticed the benefits that rich Internet applications bring to using the Web and are slowly adopting these types of applications.

While Rich Internet applications deliver superior benefits to users, it's very challenging for developers to incorporate AJAX in these applications. It requires a lot of low-level programming (read: JavaScript, XML, DOM) and developers with the necessary expertise in the area. And then, it is not uncommon to spend 80 to 90 percent of direct AJAX development battling JavaScript and browser incompatibilities.

Combining AJAX and JSF

Using AJAX also poses a challenge for the JSF community. Starting to code in JavaScript along side JSF UI components is not an option. It jumps outside of the JSF component model. A better solution is needed.

To figure out this solution, remember that JSF is based on components. Why not just have components that encapsulate all of the challenging AJAX functionality inside them? This way, using AJAX in development actually becomes pretty simple. This enables application development with extensive AJAX functionality without having to worry about low-level AJAX details like JavaScript or browser incompatibilities. Almost all such problems will have been ironed out by the component developer back when the component was tested and debugged.

Highlights on the Road to Ajax4jsf

To summarize the lead-up to Ajax4jsf, keep in mind these main points:

1. JSF is a standard component-based framework for building Web-based applications.
2. AJAX is a technique to make Web applications fast, rich, and more interactive.
3. JSF and AJAX is a perfect match because of the JSF component approach (adding AJAX functionality through additional standard JSF components that provide AJAX features).

Now that the scene is set, we can move on to the actual birth of Ajax4jsf.

Ajax4jsf

Ajax4jsf framework was created and designed by Alexander Smirnov. In early 2005, he was looking to add a "hot" new technology along with the associated experience to his resume. Roughly at the same time, the concept of AJAX was being introduced by Jesse James Garrett. Meanwhile, JSF was starting to pick up steam. Alexander figured why not just merge the two, so it would be easy to have AJAX functionality within a JSF application. He figured this would be an excellent addition to his resume. He started the project on sourcforge.net and called it Telamon (taken from the Shakespeare play, *Anthony and Cleopatra*). And, Ajax4jsf was born.

Alexander Smirnov Joins Exadel

In the fall of that same year, Alexander joined Exadel and continued to develop the framework. Alexander's goal was to create a tool that was easy to use and that could be used with any existing JSF component libraries. His marvelous understanding of the JSF framework and AJAX technology allowed him to architect and design a framework that is today used by thousands of people.

Exadel Visual Component Platform

The first version of what would become Ajax4jsf was released in March 2006. It wasn't quite a standalone thing, yet. Rather, it was part of a product called Exadel Visual Component Platform (now known as Exadel RichFaces). The initial idea was to build components based on Alexander's framework and package them with the framework. RichFaces consists of sophisticated, ready-to-use JSF AJAX components. Underneath these components, Ajax4jsf provided the foundation.

Ajax4jsf Splits Off

In late April, based on customer feedback, we realized that our customers wanted more flexibility and control in adding AJAX functionality to JSF applications. Customers liked RichFaces components, but having direct access to Ajax4jsf would give them even more muscle in developing rich JSF applications.

Responding to this feedback, we decided to move Ajax4jsf outside of RichFaces and make it a standalone open source project. The project was moved to the Java.net site, an incubator for open source projects sponsored by Sun. After setting up the project and getting approval from Sun, the project was launched at the http://ajax4jsf.dev.java.net Web site as an open source project sponsored by Exadel.

Popularity of Ajax4jsf

On being spun off, Ajax4jsf started to become wildly popular. People were pleasantly surprised at how easy it was to add AJAX functionality to the standard JSF components in their applications with an open source product. They did not have to redo or recreate anything just because they wanted to add AJAX. As word spread, traffic on the Ajajx4jsf site has been increasing each month and traffic on the mailing list has increased as well.

Ajax4jsf has made Java.net's "Top Ranked" lists more than once. Ajax4jsf has appeared in the Top 10 projects lists for mail traffic, accesses, and CVS commits numerous times. The number of articles, Internet resources, and blog entries where people talk about Ajax4jsf has also grown significantly.

Initial Problems

However, with popularity comes some vulnerability. As more and more people have started using Ajax4jsf, a significant number have put Ajax4jsf into production applications. Ajax4jsf has been put through many different usage scenarios, configurations, and environments, so, of course, bugs have surfaced. Many people have taken advantage of the mailing list to submit reports and ask a variety of questions about using the product.

Alexander Smirnov and the rest of the Ajax4jsf team were somewhat stretched at first between fixing user-submitted bugs and introducing new features. It was not always easy to satisfy a growing open source community. While critical bugs had to be fixed first, other bugs had to be fixed later.

Nevertheless, having people report bugs and tell us how we can improve Ajax4jsf is what has made it such an excellent piece of software today. I believe no QA department would be able to put Ajax4jsf through as many test cases as the users did by simply using it in their applications. This has made Ajax4jf a rock-solid product that can be used in many production environments.

Ajax4jsf's Current Feature Set

Over its year of existence, the feature set for Ajax4jsf has grown a lot. It now boasts 18 JSF components that can be used to add AJAX functionality to a JSF application. The most commonly used component is <a4j:support>, a general component that can be associated with any JSF component to invoke an AJAX request.

In addition to its component library, Ajax4jsf is also a framework that provides the following capabilities:

- Write your own custom rich components with built-in AJAX support
- Package resources with the application's Java classes
- Easily generate images on-the-fly
- Create a modern rich user interface look-and-feel with skins-based technology
- Test components, actions, listeners, and pages as you are creating them

Driving the Evolution of Ajax4jsf

Throughout the evolution of the feature set for Ajax4jsf, a few principles have been key. First, most of the changes and improvements that we make to the project are user-driven. Users request a particular functionality; the project team implements it. It's that simple. Another important guiding principle for this project is to allow Ajax4jsf to be used with any other custom JSF component library. We don't want someone to start using Ajax4jsf and then not be able to use anything else. Ajax4jsf should be usable with any JSF implementation and any custom JSF component library.

Ajax4jsf's Future

Ajax4jsf will continue to be driven by its key principles, but there is the possibility of a big change in the near future for Ajax4jsf. Right now, the JSF Expert Group is eyeing adding AJAX functionally in JSF 2.0 using the Ajax4jsf approach. This new JSF version might surface some time towards the end of 2008.

The Right Time

At the end of this story, the obvious question to ask is: Why was Ajax4jsf so successful? Ajax4jsf is indeed an excellent framework, powerful and easy to use, but excellence doesn't always lead to success.

In this case, Ajax4jsf arrived at the perfect time when the right strands of technology were developing in the right way. So, Ajax4jsf was born when it was needed the most. And, when there is a need to be filled and a product comes along that exactly fills that need, it's going to be used.

POSTSCRIPT: Ajax4jsf Becomes JBoss Ajax4jsf!

Soon after this article was first published, just before Ajax4jsf was about to celebrate its first birthday, Ajax4jsf was "reborn." Exadel formed a strategic partnership with Red Hat on March 5, 2007. All of the products that had been developed by Exadel were moved under Red Hat as open source (if they weren't open source already). Ajax4jsf is now JBoss Ajax4jsf although Exadel's programmers continue to work with Red Hat and JBoss in its development. With all of this support, Ajax4jsf is now on track to celebrate many more birthdays! Find out more from the JBoss Ajax4jsf home page:

```
http://labs.jboss.com/portal/jbossajax4jsf
```