Conference Report: No Fluff Just Stuff

26-28 August, 2005. Mark Windholtz

Overview

No Fluff Just Stuff, is a Java conference that travels around the U.S.
It provides high-quality speakers most of whom are authors of recent top-selling Java books. The number of attendees is limited to 200 so that everyone has access to the speakers. There were approximately 100 people in attendance. The conference is held over a weekend so that it does not interfere with consultant's billable hours. In all, it's a well thought out format.

This is the third time I've been to NFJS and it was also great to see everyone again.
There are 4 or 5 tracks, but I'll talk mostly about the sessions I attended.
My goal in writing this is to let readers get an idea of what happened at NFJS. If you have further in-depth questions about any of these topics contact me. Before the panel talk Joe O'Brian, (former xp-cinci member), currently at ThoughtWorks, gave a terrific endorsement of the www.xpcincinnati.org group.

Abstract

I concentrated on the Enterprise Java track, and attended sessions on Security, Performance, Service Oriented Architecture, AJAX, as well as a few sessions on Ruby. The information was very useful and I expect to immediately add many new the techniques to my consulting toolbox. The breadth of interest in Ruby surprised even me.

Most of the speakers had at least one slide relating their Java-techniques to the same thing in Ruby. And the questions from the audience at the Panel discussion were absolutely dominated by questions about *Ruby* and *RubyOnRails*. This is despite the fact that this was supposedly a *Java Conference!* At the grassroots, momentum is truly building for something less complex than Java.

Writing Secure Web Services (with Java and Axis) by Justin Gehtland

Justin talked about the 5 pillars of Internet Security

- ⇨ Confidentiality – Messages can't be intercepted during transportation on the Internet
- ⇨ Integrity – Messages can't be changed during transportation
- ⇨ Non-Repudiation – Messages can't be denied by the original sender. Sender can't say: *I didn't send that*.
- ⇨ Authentication – Prove the identity of the sender or receiver
- ⇨ Authorization – Once identity is know, what are they allowed to do on your system?

He then talked about implementations for Tomcat and the WS* specifications.

Ruby On Rails by Dave Thomas

Dave built a simple e-commerce web application while explaining it every step of the way in about an hour. This really amazed many of the Java programmers who would have expected to spend a week on similar set of features.

He explained the basic principles that make this kind of productivity possible.

- ⇨ Full Stack Framework – in J2EE one needs to know how to work in about 5 different languages to bring an application together. In Rails its one. This makes coordinating the layers much easier.
- ⇨ Don't Repeat yourself – Logic is placed in only one place in Rails. In J2EE defining a field in the database and screen means repeating the definition about 5 times in various formats.
- ⇨ Convention over Configuration – in J2EE there are lots of configuration files in various formats. Every architectural layer and each server seems to have its own configuration file. In Rails there is one config file and it is about 4 lines long per environment.

Performance Monitoring in J2EE applications, by Ramnivas Laddad

This one was a bit disappointing. He listed lots of tools and techniques, but it was not very in-depth for solving any problem in particular. I do plan to follow up on his tool recommendations though. I left early and went to Neal Ford's presentation on SAO.

SOA and ESB: Next Wave of Enterprise Development or return of CORBA? By Neal Ford

Neal admits from the beginning that SOA- *Service Oriented Architecture* is vaguely defined. In fact he calls it: *Service Oriented Ambiguity*. Services in SOA are business services that are linked together to implement a business process.

Business partners can use these services inside their own companies. SOA favors flexibility over efficiency.

There are two growing standards for SOA: WS* and REST.

- ⇨ WS* is a large set of specifications for a wide range of situations.
- ⇨ REST is a simple standard with clean readable URL encoding.

SOA focuses more on solutions rather than on building applications.

An ESB (*Enterprise Service Bus*) can be used to coordinate services with-in an enterprise.

Expert Panel

After lunch, a number of the speakers sat on the stage and took questions from the group. Surprisingly, the first 6 questions were all about Ruby and Rails. The moderator had to even cut off that line of questioning, and made the rule that questions had to concern Java. After that there was a noticeable pause. And then a 2 people offered half-hearted questions about Java.

Creating Killer Graphics and Professional PDFs with XML by Ben Galbraith

Ben talked about creating PDF files using Java, XSLT, and XML-FO. And walked though the format of the various XML files, needed.

AJAX: Creating next-generation, highly dynamic, off-line capable web applications with HTML and JavaScript, by Ben Galbraith

Ben explained the AJAX model and XMLHttpRequest. He also showed examples from Google maps and other sites of AJAX implementations that really helped the application experience. Interestingly, he demonstrated the capabilities of AJAX but showing a number of sites that were implemented in RubyOnRails (like TadaLists).

Unit Testing with Jython by Stuart Holloway

Stuart showed how to write smaller flexible unit tests with the dynamic language named Jython (A Java-Python bridge). He insisted that for many tasks a language with a dynamic meta-level was needed. He said he wanted to use JRudy (the Java-Ruby bridge) but did not think it was ready yet. After the session I introduced Stuart to a member of the JRuby team so that they could discuss his issues.

Guerrilla Web Techniques by Scott Davis

Scott had an interesting presentation on using old, low level technologies to build fresh behavior into web sites. One of his points is that the many Java frameworks have gotten so complex and so far from the real problem that sometimes it's easier to solve common problems underneath the standard frameworks.

Web Application Vulnerabilities by Neal Ford

Neal had a great presentation on the most common Internet security attacks done by hackers. He said his goal was to scare us. And he did. On the other hand he also provided simple approaches to avoiding many of the common routes of attack in our systems. Some of the attacks discussed were:

10. Insecure configuration management
9. Denial of service
8. Insecure storage
7. Improper error handling
6. Injection flaws
5. Buffer overflows
4. Cross site scripting flaws
3. Broken authentication and session management
2. Broken access control
1. Unvalidated input

Language Oriented Programming and Language Workbenches, by Neal Ford

Neal provided a real eye opener on where software development was headed.
He used MS-Excel as an example. What Excel is, is a language used by accountants to solve their problems. The development of Excel was the building of a language that was simple enough for accountants to use with little training. That is because it provides them with a language that they already understand. Accountants use rows and columns and sums etc..

What he suggests is that the future of software is that instead of hand coding applications, we are moving toward building Domain Specific languages for our users. He showed examples of how this looks in Java, XML, and Ruby. He showed how to build parsers for your own language. And showed some of the advanced workbench tools soon to be released.

The ways to build your own Excel like domain specific language are:

(1) Build a traditional parser
(2) Use one of the not yet released language workbenches
(3) Build your language in Ruby, because it's very high level already.

Conclusion

There are lots of good ideas in the Java space. But it is clear that the momentum for innovation is moving away from Java and toward Ruby, and specifically RubyOnRails. Many of the speakers referred to Ruby, many are even in the process of writing Ruby books. And most of the questions from the audience were about Ruby. It was truly surprising to see this much buzz around Ruby this year.

In the mean time, it is also clear that current infrastructures provide great support for Java and Java frameworks. It will be very interesting to see how these two technologies intermingle and compete.