

# Bay Area Software Symposium 2006

W Hotel Silicon Valley - Newark, CA

October 13 - 15, 2006

(session listing as of 10/13/2006)

Friday, October 13, 2006				
	Great Room 2	Studio 2	Studio 1	Strategy
12:00 - 1:00 PM	REGISTRATION			
1:00 - 1:15 PM	WELCOME			
1:15 - 2:45 PM	JavaServer Faces: A Whirlwind Tour David Geary	Working with Rules Engines Venkat Subramaniam	Spring Intro Justin Gehtland	Effective AOP Ron Bodkin
2:45 - 3:15 PM	BREAK			
3:15 - 4:45 PM	JSF: State of the Art David Geary	Open Source Tools for Agile Development Venkat Subramaniam	Spring Dependency Injection Justin Gehtland	Power Regular Expressions in Java Neal Ford
4:45 - 5:00 PM	BREAK			
5:00 - 6:30 PM	Ajaxian Faces David Geary	SOA: Next Wave of Enterprise Development or Return of the Son of CORBA? Neal Ford	Spring Security with ACEGI Justin Gehtland	Refactoring your code - a key step in agility Venkat Subramaniam
6:30 - 7:15 PM	DINNER			
7:15 - 8:00 PM	Keynote: Open Source from the Inside - Jason Hunter			

Saturday, October 14, 2006				
	Great Room 2	Studio 2	Studio 1	Strategy
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Ajax Architecture Justin Gehtland	Testing with Selenium Neal Ford	Groovy for Java Programmers Venkat Subramaniam	Shale: Turbo-charge your JSF Apps David Geary
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	JavaScript for Ajax Programmers Justin Gehtland	Clean Up Your Code: 10 Java Coding Tricks, Techniques, and Philosophies Neal Ford	Get Groovier with Grails Venkat Subramaniam	Essential db4o Ted Neward
12:30 - 1:30 PM	LUNCH			
1:30 - 3:00 PM	AJAX Performance and Scalability Ron Bodkin	Prototype: Ajax and JavaScript++ Justin Gehtland	Java Annotations: From Definition to Consumption Ted Neward	Real World Web Services Scott Davis
3:00 - 3:15 PM	BREAK			
3:15 - 4:45 PM	The Productive Programmer Neal Ford	JRuby Justin Gehtland	Guerrilla Web Techniques Scott Davis	Practices of an Agile Developer Venkat Subramaniam
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS			

Sunday, October 15, 2006				
	Great Room 2	Studio 2	Studio 1	Strategy
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Java5: The Language, The Libraries, The VM Ted Neward	Comparing EJB 3 with Spring and Hibernate Chris Richardson	Holistic Testing Scott Davis	Pragmatic Tracer Bullets Jared Richardson
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	Pragmatic XML Services Ted Neward	Developing Rich Domain Models Chris Richardson	Groovy: Greasing the Wheels of Java Scott Davis	Agile Software Testing Strategies Jared Richardson
12:30 - 1:15 PM	LUNCH			
1:15 - 2:00 PM	EXPERT PANEL DISCUSSION			
2:00 - 3:30 PM	Extreme Web Caching Jason Hunter	Software Development Techniques Jared Richardson	Effective Enterprise Java: State Management Ted Neward	Easing into Agile Scott Davis
3:30 - 3:45 PM	BREAK			
3:45 - 5:15 PM	Forgotten Algorithms Jason Hunter	Enhance Design Patterns with AOP Nick Lesiecki	Java6: Exploring Mustang Ted Neward	Continuous Integration with Cruise Control Jared Richardson

# Bay Area Software Symposium 2006

W Hotel Silicon Valley - Newark, CA

October 13 - 15, 2006

(session listing as of 10/13/2006)

## Comparing EJB 3 with Spring and Hibernate by Chris Richardson

The limitations of EJB2 led to the development of the extremely popular Spring and Hibernate frameworks. These frameworks replaced the cumbersome EJB2 programming model with a nimble, non-invasive Plain Old Java Object (POJO) #based model. But, now, the EJB3 specification has embraced many of the ideas made popular by Spring and Hibernate including POJOs, transparent persistence and dependency injection.

## Developing Rich Domain Models by Chris Richardson

Object-oriented design (OOD) is good way to tackle the complexity of modern applications. Yet many complex, enterprise Java applications are written in a procedural style. One reason is because EJB2 created too many obstacles to using object-oriented design techniques. Fortunately, enterprise Java technologies have improved. Plain Object Java Objects (POJOs) and object/relational mapping frameworks such as Hibernate, JDO and EJB3 led to the revival of OOD.

## Ajaxian Faces by David Geary

JavaServer Faces is a perfect platform for implementing Web 2.0 interfaces with Ajax. This session explores how you can use these two potent technologies--JSF and Ajax--together to create applications that look and behave like desktop applications but run in the browser.

## JSF: State of the Art by David Geary

In 2005, JSF hit its stride, as evidenced from overwhelming support from both vendors and the open-source community. JSF 1.0 had plenty of holes, but open-source projects have arisen to address those needs. This session takes a look at three of those projects: 

- Tomahawk (MyFaces component library)
- Facelets
- Seam

## JavaServer Faces: A Whirlwind Tour by David Geary

JavaServer Faces (JSF) has arrived. The standard Java-based web application framework based on Struts, JSF really took off in 2005. Embraced by developers, vendors, and open-source projects, JSF has started to hit its stride. If you haven't come up to speed on JSF basics, this is the place to start.

## Shale: Turbo-charge your JSF Apps by David Geary

JavaServer Faces is a well designed user interface framework, but it lacks a number of features you might otherwise expect out of the box; for example, JSF does not explicitly provide support for client-side validation. So, from the folks that brought you Struts, comes Shale, a collection of useful enhancements to JSF. A top-level Apache Software Foundation project, Shale adds some really cool features to vanilla JSF, including: 

- Web flow: script dialog flow
- Remote Method Calls: easily call JavaBean methods from JavaScript
- Tapestry-like views: code views in pure HTML
- Use Apache Commons Validator validators on the client or server, or both
- JSF testing framework: mocks for easy JSF testing

 There's a lot of cool stuff in Shale that makes JSF a much more compelling proposition. Come see what it's all about.

## Agile Software Testing Strategies by Jared Richardson

Creating and maintaining a solid automated test suite is critical to an Agile strategy, but often we're just told to "Do it." In this talk we'll look at several pragmatic strategies for creating and building your suite.

## Continuous Integration with Cruise Control by Jared Richardson

Continuous Integration is increasingly recognized as a vital practice in an Agile software shop. Traditionally it's been difficult to set up and administer. Today, that's no longer the case.

## Pragmatic Tracer Bullets by Jared Richardson

Are your product designs hit or miss? Do you have trouble building a loosely coupled system? Is your code incestuous? Refactoring not an option with your code base? Tracer Bullets help keep your project out of the fire. Tracer Bullet Development: 

- \* helps you create great software
- \* lends itself to an iterative cycle
- \* can be used for demos early and often
- \* is easily refactored
- \* allows your teams to work in parallel
- \* makes a very testable system

## Software Development Techniques by Jared Richardson

Throughout our software careers we learn habits from our coworkers, from books we've read, and

occasionally, from conferences we attend. Much of our competence comes from the tips and tricks we pick up as we go.

### **Extreme Web Caching by Jason Hunter**

Web Caching is very important for high traffic, high performance web site but few people know all the professional-level strategies. In this talk I'll share some of the tricks of the trade, including advanced tips from Yahoo's Mike Radwin. We'll start with the basics: using client-side caches, conditional get, and proxies. Then we'll talk about more advanced features: how best to handle personalized content, setting up an image caching server, using a cookie-free domain for static content, and using randomization in URLs for accurate hit metering or sensitive content.

### **Forgotten Algorithms by Jason Hunter**

There are many interesting and useful algorithms that people just don't remember or never learned. The Boyer-Moore string search algorithm is one prime example. The randomized skip list is another. Both solve common problems with wonderful flair and finesse -- and performance-wise they blow the pants off brute force solutions. This session covers these two algorithms plus several others. It's like your college algorithms course but with a practical bent and absolutely zero proofs. Extra bonus: The Google PageRank algorithm.

### **Open Source from the Inside by Jason Hunter**

Open source isn't about a license, it's about human interaction and individual motivation. I've seen open source from all sides. I've been an individual contributor and a project leader. I've worked on commercial and open source efforts, and have both helped commercial projects go open and designed ways for open projects to absorb commercial codebases. I've been on the front lines in the Apache/Sun negotiations on open source Java that ended on the JavaOne keynote stage with Scott McNealy. In this talk, I'd like to share my favorite stories in and around open source and the lessons they teach us.

### **Ajax Architecture by Justin Ghtland**

Ajax applications have unique architectural challenges and opportunities. This presentation will show you how to take advantage of the Ajax's strengths, and work around its quirks.

### **JRuby by Justin Ghtland**

JRuby is not one, but two great technologies: the Ruby language, and the Java Virtual Machine and libraries. In this talk you will learn the basics of programming JRuby, and how to integrate JRuby code into existing Java projects.

### **JavaScript for Ajax Programmers by Justin Ghtland**

This presentation covers JavaScript from the perspective of an Ajax programmer. We assume that you may be using an Ajax toolkit, but still need to be able to read, modify, and test the JavaScript code in your application. You will learn the common idioms of JavaScript by looking at working code from the Ajax toolkits themselves.

### **Prototype: Ajax and JavaScript++ by Justin Ghtland**

Learn to simplify Ajax development with Prototype through a series of real-world examples. Along the way, learn to code in Prototype's modern JavaScript style, taking advantage of Prototype's extensions to JavaScript's object model

### **Spring Dependency Injection by Justin Ghtland**

Dependency Injection (DI) is the cornerstone of Spring. The core concept is quite simple, but (surprise!) actual practice can become complex. To take full advantage of Spring DI, you need to understand not only the basics on configuration, but also the container lifecycle model and the various hooks provided by the framework.

### **Spring Intro by Justin Ghtland**

The Spring framework is one of the fastest growing open source frameworks. New job postings are gaining rapidly, and many customers are adopting Spring instead of heavier alternatives. In this session, we'll introduce Spring. You'll see how Spring can give you much of the power of EJB, without the complexity or pain. Spring uses concepts like dependency injection and aspect oriented programming to ease standard enterprise development. Spring developers write plain, ordinary Java objects (POJOs), instead of sophisticated components. In this session, you'll see a basic Spring application. You'll also see some details about some of the enterprise integration strategies, including: # Spring AOP # Transactions # Persistence # Model/view/controller When the session is over, you won't be an expert, but you should have a much clearer understanding of what Spring does, what it doesn't do, and why it's growing so rapidly.

### **Spring Security with ACEGI by Justin Gehtland**

Spring offers developers a simpler, more robust method for configuring applications. These benefits extend to security through the ACEGI framework. ACEGI makes the otherwise daunting task of securing your application logical and straightforward. More importantly, through its support for single sign-on provision through Yale's CAS system and its ability to provide instance-level authorization, Spring extends the common security model of most J2EE apps beyond what they are traditionally capable of.

### **Clean Up Your Code: 10 Java Coding Tricks, Techniques, and Philosophies by Neal Ford**

This session delivers 10 techniques for improving your code, whether you are freshly graduated or a grizzled veteran.

### **Power Regular Expressions in Java by Neal Ford**

Regular expressions should be an integral part of every developer's toolbox, but most don't realize how important it is. Regular expressions have existed for decades, but many developers don't understand how to take full advantage of this powerful mechanism, either through command line tools and editors or in their development.

### **SOA: Next Wave of Enterprise Development or Return of the Son of CORBA? by Neal Ford**

Is Service Oriented Architecture the next wave of distributed computing or just the same old crap in a shiny new package? This session provides an overview of what most people agree is the definition of SOA. I talk about SOA, ESB, CORBA, your MOM, and a bunch of other acronyms.

### **Testing with Selenium by Neal Ford**

This session describes the use and workings of Selenium, the open source web user interface testing tool.

### **The Productive Programmer by Neal Ford**

This session shows you how to become a more productive programmer every day by using tools that you didn't know you already had.

### **Enhance Design Patterns with AOP by Nick Lesiecki**

Design patterns have long been part of the experienced developer's tool chest. However, design patterns can affect multiple classes and this makes them invasive and hard to (re)use. This presentation will discuss how AOP solves this problem by fundamentally transforming pattern implementation. The class will examine examples of various traditional design patterns (including some of the famous GoF patterns) and discuss the practical and design benefits of implementing them with aspect-oriented techniques. This session will be of interest to anyone who has struggled with design patterns. It is also the perfect session for a programmer interested non-trivial applications of AOP, or who wishes to see aspect-oriented design in action.

### **AJAX Performance and Scalability by Ron Bodkin**

You know AJAX can improve Web application usability, but only if designed properly. How do you deploy and manage responsive rich client networked applications that scale? If you are using services from multiple sources, how can you identify bottlenecks?

### **Effective AOP by Ron Bodkin**

Aspect-Oriented Programming (AOP) allows better application architectures by centralizing scattered and tangled code for problems like security, error handling, testing, policy enforcement and feature variations across a product line. The recent release of AspectJ 5 and announcement of Spring 2.0 illustrate the maturing options for Java projects.

### **Easing into Agile by Scott Davis**

How do you get started with an Agile development methodology? Everyone has been talking about eXtreme Programming for years, but how do you get it introduced to your team? Many times, you're not simply transitioning from one methodology to another -- you're introducing a methodology for the first time. Adding structure to a previously unstructured endeavor. Adding a touch of discipline where programmers once roamed free.

### **Groovy: Greasing the Wheels of Java by Scott Davis**

This is the year of the dynamic scripting language. Ruby (and Rails) has won the hearts and minds of many independent software developers. JavaScript is experiencing a renaissance thanks to the wild success of AJAX and websites like Google Maps. And Groovy (JSR-241) brings the same level of excitement and "scripting goodness" to the Java platform.

### **Guerrilla Web Techniques by Scott Davis**

Frameworks? We don't need no stinkin' web frameworks. OK, so maybe that's overstating the case. Web frameworks do plenty of good things, but sometimes they can also be golden handcuffs. Too many web developers fall into the trap of thinking, "If it can't be done by my web framework, then it simply can't be done."

### **Holistic Testing by Scott Davis**

Mark Twain once said, "Everyone talks about the weather, but nobody does anything about it." Do you feel the same way about Unit Testing? Are you actively testing your code, or are you just thinking about testing your code... some day... once you get some more free time...

### **Real World Web Services by Scott Davis**

In this talk, we'll survey the web services exposed by leading websites (Google, Yahoo, Amazon, eBay) and discuss how they are driving the AJAX revolution. You'll see examples of RESTful, SOAP, and JSON web services, as well as the strengths and weaknesses of each.

### **Java5: The Language, The Libraries, The VM by Ted Neward**

Java5 introduced a whole slew of new features, including annotations (JSR 175), new language features (the enhanced for loop, generics, static imports, and more), new library support (java.lang.instrument, among others), and some interesting enhancements to the virtual machine itself.

### **Effective Enterprise Java: State Management by Ted Neward**

Managing state--both transient state (like your shopping cart) and your durable state (like your order placements, your inventory management forms, and so on)--is tricky in an enterprise application. In this talk, we'll examine some of the trickiness, both high-level and low-.

### **Essential db4o by Ted Neward**

Tired of writing object-table mappings? For years, Java developers have wrestled with the problems of storing objects into relational format and retrieving them back again; for all that Hibernate and JDO and other O/R tools make it easier (though not easy) to do, isn't there another way? In this presentation, we'll explore an alternative approach, real object persistence, using the db4o toolkit ([www.db4objects.com](http://www.db4objects.com)).

### **Java Annotations: From Definition to Consumption by Ted Neward**

Want to get the soup-to-nuts story on Java annotations? In this presentation, we'll first talk about what annotations provide to the Java language. After setting ourselves a conceptual basis to operate from, we'll look at the language definition for Java annotations, from how to use them to how to define them. Finally, we'll take a look at the other side of annotations, consuming them at source-level (using "apt", the annotation processing tool), class-level (using a bytecode toolkit such as BCEL), and at runtime (using enhancements to the Reflection API made in Java5).

### **Java6: Exploring Mustang by Ted Neward**

Mustang, the forthcoming Java6 release, is just around the corner, and even if you're not looking to adopt the new platform right away, it's important to know what's there so you can start to plan for it. In this presentation, we'll go over the major new features of the Java6 platform, including the new integrated XML services capabilities (JAX-WS and JAXB), dynamic/scripting language support (javax.script), new JVM "attach" capabilities, new annotations supported by the javac compiler, and more.

### **Pragmatic XML Services by Ted Neward**

There's a lot of talk about web services, and most of it falls into one of two categories: lots of low-level talk about vendor-specific tools and extensions, or lots of high-level talk that never shows you a line of code. XML services aren't that hard, and in this talk, we'll see how, why and when to do one.

### **Get Groovier with Grails by Venkat Subramaniam**

Inspired by the Ruby on Rails project, Grails brings the ease of web development and "convention over configuration" to the Java platform. We will learn how to create web applications using Grails, how to integrate it with Hibernate, and how to Ajax it, all using the built in features of Grails. This section assumes that you are familiar with Groovy or you have attended the #Groovy for Java Programmers# session. The session will be example driven with live coding where we will build a web application from scratch.

### **Groovy for Java Programmers by Venkat Subramaniam**

Object-oriented scripting languages, or agile dynamic languages, as some like to call those, are gaining programmers' attention. Groovy bring this excitement to the Java platform with its ability to generate byte

code. You can use Groovy instead of Java for some parts of your application. By learning it, you can switch between the languages where you consider fit.

### **Open Source Tools for Agile Development by Venkat Subramaniam**

As a Java developer, you have taken the time to learn the basics of the language and relevant parts of its rich API. However, you need more than that to develop serious industrial strength applications. In this presentation, the speaker will introduce you to a number of open source tools which you can use to improve your application quality and your development process.

### **Practices of an Agile Developer by Venkat Subramaniam**

You have worked on software projects with varying degree of success. What were the reasons for the success of your last project? What were the reasons for those that failed? A number of issues contribute to project success - some non-technical in nature. In this presentation the speaker will share with you practices in a number of areas including coding, developer attitude, debugging, and feedback. The discussions are based on the book with the same title as the talk.

### **Refactoring your code - a key step in agility by Venkat Subramaniam**

Refactoring is one of the core practices in Agile Software Development. Refactoring is based on some core principles that apply to more than writing good code. But, what's refactoring? Why should you do it? How do you go about doing that? What tools are available to successfully refactor your App?

### **Working with Rules Engines by Venkat Subramaniam**

Rule based programming allows us to develop applications using declarative rules. These can simplify development in applications where such rules based knowledge is used for decision making.