

# Northern Virginia Software Symposium

Sheraton Reston

November 7 - 9, 2008

<http://www.nofluffjuststuff.com/conference/reston/2008/11/index.html>

(event schedule as of November 5, 2008)

Fri, Nov. 07, 2008						
	Meeting Room 5	Meeting Room 6	Meeting Room 7	Meeting Room 8	Meeting Room 9	Meeting Room 10
12:00 - 1:00 PM	REGISTRATION					
1:00 - 1:15 PM	WELCOME					
1:15 - 2:45 PM	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	Java Persistence: Approaching the Silver Bullet Mark Richards	10 Tips for Getting Your Project Back on Track Jared Richardson	Java 7 Preview Alex Miller	Give it a REST Brian Sletten	Choosing the right RIA platform for your project Richard Monson-Haefel
2:45 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind Scott Davis	10 Things Every Software Architect Should Know Richard Monson-Haefel	Techniques 2008 Jared Richardson	Design Patterns Reconsidered Alex Miller	RESTlet for the Weary Brian Sletten	Enterprise Messaging Using JMS (Part 1) Mark Richards
4:45 - 5:00 PM	BREAK					
5:00 - 6:30 PM	Rapid Web Development with Grails and Ajax Scott Davis	Rise of the Fit Client Richard Monson-Haefel	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson	Exploring Terracotta Alex Miller	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	Enterprise Messaging With JMS (Part 2) Mark Richards
6:30 - 7:15 PM	DINNER					
7:15 - 8:00 PM	Keynote: by Jared Richardson					

Sat, Nov. 08, 2008						
	Meeting Room 5	Meeting Room 6	Meeting Room 7	Meeting Room 8	Meeting Room 9	Meeting Room 10
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Structuring concurrent applications in JDK 5.0 Brian Goetz	Enterprise Integration with Spring, Part 1 Mark Fisher	Restoring Agility: Getting Your Team Back on Track Jared Richardson	YSlow: Building Your Website for Speed Scott Davis	JavaServer Faces: A Whirlwind Tour David Geary	EJB3 Core Specification (JSR-220) Mark Richards
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Effective Concurrent Java Brian Goetz	Enterprise Integration with Spring, Part 2 Mark Fisher	Be Heard: Public Speaking for Techies Jared Richardson	GIS for Web Developers: Adding Where to Your Application Scott Davis	Facelets David Geary	Transaction Design Patterns Mark Richards
12:30 - 1:30 PM	LUNCH					
1:30 - 3:00 PM	Java.next #1: Common Ground Stuart Halloway	Architecture and Scaling Ken Sipe	Agile, Smagile: What's Working? - What's Not? David Hussman	Configuring Spring with Annotations Mark Fisher	Using Ajax4jsf David Geary	Garbage-collector-friendly programming Brian Goetz
3:00 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Java.next #3: Dispatch Stuart Halloway	7 Habits of Highly Effective Developers Ken Sipe	Architecture and Agility Are Not Mutually Exclusive David Hussman	High Performance Projects Srivaths Sankaran	Intro to Seam David Geary	Squashing bugs with FindBugs Brian Goetz
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION					

Sun, Nov. 09, 2008						
	Meeting Room 5	Meeting Room 6	Meeting Room 7	Meeting Room 8	Meeting Room 9	Meeting Room 10
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	How to Fail with 100% Code Coverage Stuart Halloway	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	A Thorough Introduction To Groovy Jeff Brown	Agile Management & Managing Agility David Hussman	Enterprise JPA - Tips and Tricks for JEE5 Persistence Pratik Patel	Easy BDD with Groovy Andrew Glover
10:30 - 11:00 AM	MORNING BREAK					
11:00 - 12:30 PM	Java Memory, Performance and the Garbage Collector Ken Sipe	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	Agile Test Driven Development With Groovy Jeff Brown	Agile Product Planning: Building Strong Backlogs David Hussman	Viva La Javolution! Brian Sletten	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock
12:30 - 1:15 PM	LUNCH					
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION					
2:15 - 3:45 PM	Your app is done - now what? David Bock	Real-world JEE performance tuning: Tips n' Tricks Pratik Patel	Powerful Metaprogramming Techniques With Groovy Jeff Brown	Iteration 0 Ken Sipe	Introduction to Hibernate Scott Leberknight	Tactical Continuous Integration with Hudson Andrew Glover
3:45 - 4:00 PM	BREAK					
4:00 - 5:30 PM	Surviving Middle Management David Bock	Hacking - The Dark Arts Ken Sipe	Grails - Agile Web 2.0 The Easy Way Jeff Brown	Shining a Light on Eclipse Srivaths Sankaran	Real World Hibernate Tips Scott Leberknight	Less is Always More Andrew Glover

# Northern Virginia Software Symposium

Sheraton Reston

November 7 - 9, 2008

<http://www.nofluffjuststuff.com/conference/reston/2008/11/index.html>  
(event schedule as of November 5, 2008)

## **Java 7 Preview by Alex Miller**

Possible library and language changes for Java 7 have been in discussion since early 2007 and many things have been proposed. This talk will focus on those items that are most likely to be included and some of the more controversial language changes that might be included in a future release. For a comprehensive list of almost everything being considered, see <http://tech.puredanger.com/java7>.

## **Design Patterns Reconsidered by Alex Miller**

The Design Patterns book launched a revolution in object-oriented design and provided a vocabulary for OO developers to communicate their ideas. However, in some cases, patterns used blindly can lead to awkward, confusing, or hard to maintain code. It is time for some common patterns used in Java to be reconsidered so that we can derive the benefits from patterns while minimizing their concerns. This talk will re-evaluate key patterns like Singleton, Template Method, Visitor, and Proxy. These patterns have downsides and in some cases, do more harm than good. Examples of each pattern will be given in Java and examined for clarity, testability, and flexibility. Important problems will be discussed and examples of alternate solutions will be given.

## **Exploring Terracotta by Alex Miller**

Terracotta is an open-source Java clustering technology. It creates a virtual, durable Java heap that is shared across a cluster of Java Virtual Machines. This is done by dynamically instrumenting bytecode at load time to intercept calls to read and write fields, and also to enter and exit monitor locks. Information about these calls is then transmitted to the Terracotta Server (which can also be clustered) and out to other nodes in the cluster as needed. The advantage of this approach is that many Java programs can be clustered without code changes by providing just external Terracotta configuration. Many performance optimizations are performed to minimize communication and locking costs. Terracotta is commonly used for session sharing in web applications, distributed caching, and distributed workflow processing. This presentation will give an overview of the Terracotta technology, how it's implemented, and common use cases that can benefit from the technology. We will look at some code and cluster some Java applications during the presentation.

## **Easy BDD with Groovy by Andrew Glover**

Behavior-driven development, or BDD, has attracted a lot of attention via RSpec in the Ruby community, but BDD's roots stem from JBehave, a Java based framework modeled off of the xUnit paradigm. But JBehave isn't the only framework available for Java developers-- with the advent of Groovy, new options are available for embracing BDD in the spirit of RSpec's innovative behavior based DSL.

## **Tactical Continuous Integration with Hudson by Andrew Glover**

This session will walk attendees through a series of iterations on a fictional Java project where an automated build system is created that facilitates compilation, testing, inspection, and deployment. This build system is then plugged into the Hudson CI server and as features are coded using Agile techniques like developer testing, attendees will ultimately see firsthand how a Continuous Integration process reduces risk and improves software quality.

## **Less is Always More by Andrew Glover**

The spectacular success and market penetration of the iPod, Agile software development principles, Twitter, and even Google are just a few examples of minimalism in action.

## **Structuring concurrent applications in JDK 5.0 by Brian Goetz**

JDK 5.0 is a huge step forward in developing concurrent Java classes and applications, providing a rich set of high-level concurrency building blocks.

## **Effective Concurrent Java by Brian Goetz**

The Java programming language has turned a generation of applications programmers into concurrent programmers through its direct support of multithreading. However, the Java concurrency primitives are

just that: primitive. From them you can build many concurrency utilities, but doing so takes great care as concurrent programming poses many traps for the unwary.

### **Garbage-collector-friendly programming by Brian Goetz**

To many developers, garbage collection is black magic. Accordingly, there are a lot of conflicting advice about what is good or bad for the garbage collector. In this talk, I look at how garbage collection is implemented in the HotSpot VM, and techniques for writing programs that exhibit good garbage collection behavior. Surprisingly, many of these techniques coincide with writing good, clean code.

### **Squashing bugs with FindBugs by Brian Goetz**

Does your program have bugs, despite unit tests, integration tests, and code reviews? You bet. Are you using static analysis as part of your QA process? If not, you're probably missing out on some bugs that can be caught before they bite your customers.

### **Give it a REST by Brian Sletten**

As developers, we sometimes get to make choices about the technologies we use, sometimes not. We base these decisions on personal experiences, recommendations from others and a general sense of where the industry is going. Web Services have been all the rage for several years now. We have been told time and again that we should be building systems around them; as an industry, we've never been more confused. Perhaps it is time to Give it a REST.

### **RESTlet for the Weary by Brian Sletten**

If you have started to take a look at REST as way of exposing web services or managing information spaces, you may be frustrated by the support offered by legacy containers. There is no direct support for REST concepts in the J2EE specs (yet). XML-based configurations are so 1990's. Come learn about Restlets, a little API that has caught the attention of many in the RESTafarian community. **Prerequisite:** *Give it a REST (unless you are very comfortable with REST)*

### **What's Going On? : Complex Event Processing w/ Esper by Brian Sletten**

We write very complicated software, don't we? In our systems, we detect when simple things happen. Customers log in, people buy things, a stock is sold at a particular price, inventory shifts locations... all of these events mean little things, but what about the larger picture? Complex events are particular patterns of simpler events that suggest something deeper is happening. Do you know how you'd discover these bigger picture occurrences? Come hear how the Esper open source software represents a new class of complex event processing (CEP) frameworks that can be added to even high volume, high transaction systems.

### **Viva La Javolution! by Brian Sletten**

You're a good Java programmer. You understand the JDK libraries and how to use them. The problem is that many fundamental APIs don't take the bigger performance picture in mind. Garbage collection can end up killing your app if you aren't careful. Concurrency problems and contention can keep your well-intentioned software from leveraging modern hardware architecture that support multi-core and multi-cpu systems. Who knew that simply using the standard library code the way it was designed was opening you up for performance problems in your apps? Don't worry, Javolution has your back.

### **Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools by David Bock**

How many times have you started a new project only to find that several months into it, you have a big ball of code you have to plod through to try to get anything done? How many times have you been the 'new guy' on an established project where it seems like the code grew more like weeds and brambles than a well-tended garden? With a few good structural guidelines and several tools to help analyze the code, we can keep our project from turning into that big ball of mud, and we can salvage a project that is already headed down that path.

### **Your app is done - now what? by David Bock**

After months of effort, your software is done. Or is it? Very few successful projects in our industry are really 'done'... The success of the software often breeds feature requests, spinoff ideas, scalability concerns, not to mention the continued maintenance of the hosting platform, security, data backup, and so on.

### **Surviving Middle Management by David Bock**

Most good developers eventually have the opportunity to be managers. Whether they call you the "project manager", "Technical Lead", "Lead Developer", or some other classic middle-management title, you become the 'goto' guy between management and developers. You're the guy who is expected to keep the project in-line, track a schedule, and occasionally answer the question "How's it going?", and perhaps still contribute at a technical level. So how do you do that?

### **JavaServer Faces: A Whirlwind Tour by David Geary**

In April 2005, annual growth rates for jobs in JavaServer Faces, Struts, and Ruby on Rails were all at about 0%. Today, Struts' growth rate still hovers around 0%, but JSF and Rails have taken off. At the end of 2007, both JSF and Rails were growing at a rate of between 400-500% annually (according to indeed.com). JSF has passed the adoption tipping point, and is now the Java-based framework of choice, as is evidenced by its ecosystem. From vendors such as MyEclipse and RedHat to open source projects such as Seam, Facelets, and Ajax4JSF, JSF is where the action is. Come see why JSF is so popular. In this code- and demo-intensive session, I'll show you the fundamentals of JSF. **Prerequisite:** *Some knowledge of Java-based web applications, such as Struts, is a plus, but is not required. If you have a significant experience with JSF, you probably already know most of what's covered in this session.*

### **Facelets by David Geary**

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet. It's popularity is well deserved, and in fact, much of what is in Facelets today will make its way into the JSF 2.0 spec due out in 2008. So not only can you come to this session and see some really cool demos that you can put to use in the real world, but you'll also be learning JSF 2.0 before it's even been defined! How's that for a ROI? **Prerequisite:** *Some knowledge of JSF is essential. If you're familiar with a templating framework, such as Velocity or Tiles, that's a plus, but not required.*

### **Using Ajax4jsf by David Geary**

Ajax4jsf makes it very easy to add Ajax to your JSF applications. Come to this presentation to see how.

### **Intro to Seam by David Geary**

Have you ever stopped to think that you need to learn two frameworks to develop a non-trivial, database-backed, web application? Struts and iBatis; JSF and Hibernate; Tapestry and EJB3.0.

### **Filthy Rich Clients with the Google Web Toolkit, Part I by David Geary**

The Google Web Toolkit (GWT) is truly a revolutionary framework that lets you develop Ajaxified web applications without knowing anything about Ajax or JavaScript. But the GWT goes way beyond basic Ajax by letting you implement desktop-like applications that run in the ubiquitous browser.

### **Filthy Rich Clients with the Google Web Toolkit, Part II by David Geary**

In the second part of this talk, you will learn how to extend the GWT by implementing custom widgets, including a scrolling viewport and a drag and drop framework. After discussing custom widgets, you will see how to integrate database access into your GWT applications, and how to deploy your GWT applications to external servers.

### **Agile, Smagile: What's Working? - What's Not? by David Hussman**

With the growth of agile comes the need to add a new line to the Agile Manifesto: Success over Dogma. The number of people who can say agile is growing faster than the number of people benefiting from agile practices. There are now many successful agile projects, yet there are also a growing number of projects claiming to be agile but not seeing any of the benefits agile methods provide. This session will discuss successful adoptions of agile, dumb things you can do to muck it up, and more.

### **Architecture and Agility Are Not Mutually Exclusive by David Hussman**

Being agile does not mean living life one iteration at a time. Agile projects without a long view can run into the common design problems of the past. Planning iteration by iteration is often foolish and feeds the myth that agile projects do not think beyond a few weeks. Successful agile projects plan within iterations and across iterations. The later planning is called release planning and it is the forum where agility first engages architecture and other cross cutting concerns.

### **Agile Management & Managing Agility by David Hussman**

Management and agility are not mutually exclusive. Many managers are already working in an agile manner as a means to improve, produce, or simply survive. Other managers hear about projects using agile methods and struggle to find a place in the project community. This session provides a new way to think about managing projects. Some managers will find that their existing practices and skills are supported and enhanced by the forums and metrics provided within an agile project while others will be challenged by some of the principles and practices.

### **Agile Product Planning: Building Strong Backlogs by David Hussman**

Agile methods have cut through the noise and lighten the burden of crafting requirements documents. While this is good, it also shows clearly see that defining and guiding the creation of software products is challenging work. Most agile projects use a product backlog as a place to hold anything that will improve the product. Creating strong product backlogs is less defined than many of the other agile practices. Backlogs contain many items: user stories, architectural spikes, investments in updating and maintaining development and other environments, and more. While it is clear that developers primarily code, it is often less clear who adds to and grooms the backlog.

### **10 Tips for Getting Your Project Back on Track by Jared Richardson**

Software projects fail over and over for many of the same reasons. We'll look at some of the more avoidable problems and some solid ways to fix them, or avoid them in the first place.

### **Techniques 2008 by Jared Richardson**

There are a number of great techniques you can use across technologies and projects. Come hear some of my favorites and contribute a few of your own. We'll discuss topics from DRY to creating a zone defense for your product.

### **Credit Card Software Development: Recognizing and Repaying Technical Debt by Jared Richardson**

Technical debt has long been recognized in technical circles for years, but convincing your manager to budget time to repay "technical debt" has always been problematic. Let's couch the term technical debt concept in language more familiar to our managers: credit card debt.

### **Career 2.0: Take Control of Your Life by Jared Richardson**

Has your career been a random product of your manager's whims or company's needs? Never rely on your company to keep your skills current and marketable. Take control of your own career with a proven strategy.

### **Restoring Agility: Getting Your Team Back on Track by Jared Richardson**

An agile team is first and foremost "a team". When that gets lost in the rush to get a product out the door, the people suffer as well as the products. It's bad for the company, but even worse for the team members. We'll learn how to defuse some of the more common problems you'll run into on dysfunctional teams.

### **Be Heard: Public Speaking for Techies by Jared Richardson**

Most people fear public speaking more than death, but you don't have to let it handicap you or your career. Learn solid techniques for managing yourself, your content, and your audience.

### **A Thorough Introduction To Groovy by Jeff Brown**

Groovy is an agile dynamic language for the Java platform. The language and its libraries bring many things to the table to ease the process of building applications for the Java platform. This session provides a detailed run through Groovy with lots of code samples to drive home the power of the language.

### **Agile Test Driven Development With Groovy by Jeff Brown**

Dynamic languages bring a lot of interesting elements to the table for teams interested in doing Test Driven Development (TDD). Groovy lends itself very well to TDD and this session demonstrates many features of the language and its libraries that help teams build more testable systems and build better tests.

### **Powerful Metaprogramming Techniques With Groovy by Jeff Brown**

Metaprogramming is a key component in building truly dynamic and flexible applications with Groovy. Groovy's metaprogramming capabilities bring great new possibilities to the table that would be very difficult

or just plain impossible to write with Java alone. This session will demystify a lot of the magic that seems to be going on inside of a Groovy application. **Prerequisite:** *A Thorough Introduction To Groovy*

### **Grails - Agile Web 2.0 The Easy Way by Jeff Brown**

Grails is a full stack MVC framework for building web applications for the Java platform. Grails makes web application development both fun and easy. This session covers all of the fundamentals of building web applications with Grails.

### **Architecture and Scaling by Ken Sipe**

Scale... what is scale... how do you applications which are scalable. How do you know if the application scales?

### **7 Habits of Highly Effective Developers by Ken Sipe**

Thoughts lead to words, words lead to action, actions lead to habits. In this session we'll sharpen the development saw in the process of understanding what makes a hyper-productive programmer. The focus will consist of developer habits and development processes.

### **Java Memory, Performance and the Garbage Collector by Ken Sipe**

You are using Java, whew!!! No need to worry about memory, the garbage collector will handle that. Those who have had a memory issue in Java are not so naive any more. Often memory utilization and heap sizes are an after thought and are not recognized until the application is in production, often caused by application uptime, production request volume or production sets of data. When the OutOfMemory Error occurs, often the science of development seems to brake down and knobs are turned. First the (-mx) maximum heap space gets adjusted... More is better right. The next OutOfMemory, heads start scratching, code reviews start in earnest, and Google gets several new hits. Did you know that it is possible to get an OutOfMemory error without running out of heap space?

### **Iteration 0 by Ken Sipe**

The success of an Agile / SCRUM project is a successful start. The first interaction is often referred to as iteration 0. Other iterations have a set of stories with clear acceptance certain which establishes the velocity of the team and its effort. What then is accomplished in iteration 0? How do we get an Agile process started.

### **Hacking - The Dark Arts by Ken Sipe**

A live Hacking demonstration exposing the tools and techniques used by Hackers.

### **Enterprise Integration with Spring, Part 1 by Mark Fisher**

The first part of this two-part session will focus on the essentials of Enterprise Integration with Spring. The discussion will cover the enterprise integration support libraries in the Spring Framework core within the context of well-established design principles such as loose coupling and separation of concerns.

### **Enterprise Integration with Spring, Part 2 by Mark Fisher**

The second part of this two-part session will introduce Spring Integration, a new addition to the Spring portfolio. We will begin with a high-level overview of Enterprise Integration Patterns as catalogued in the highly influential book of the same name. We will then embark on a demo-driven exploration of Spring Integration to see how it enables the development of applications based on those patterns. **Prerequisite:** *Enterprise Integration with Spring, Part 1*

### **Configuring Spring with Annotations by Mark Fisher**

In this session, we will take a deep-dive into annotation-based dependency injection with Spring 2.5. You will learn how to combine annotation and XML formats, how to customize component scanning, and how to leverage Java 6 annotations within a Spring application. Since there is no "one size fits all" solution to application configuration, we will wrap up the discussion with general guidelines to consider when employing this approach.

### **Java Persistence: Approaching the Silver Bullet by Mark Richards**

Java Persistence has come along way since the days of straight JDBC coding and custom framework development. We have at our disposal several outstanding open source frameworks such as Hibernate, Toplink, iBatis, and OpenJPA (just to name a few), and we now have a promising and emerging

standards-based solution called Java Persistence API (JPA). However, all too often we find in the Java persistence space that it is a world of one-size-does-not-fit-all. We continually struggle with traditional ORM solutions like Hibernate when it comes to reporting queries, complex queries, complex relationships, and stored procedures, and we also struggle with managing the enormous amount of SQL required for solutions such as iBATIS or JDBC-based frameworks. In this coding-intensive session we will take a detailed look at identifying and overcoming the challenges we face when using frameworks such as Hibernate, iBATIS, and JPA, and how to combine the various persistence frameworks to create an effective Java persistence solution that approaches (but of course does not reach) the silver bullet.

### **Enterprise Messaging Using JMS (Part 1) by Mark Richards**

The chances are good that at some point in your career you will need to use messaging to pass information between applications, subsystems, or external systems, particularly with service-oriented architecture on the rise. The Java Messaging Service (JMS) allows Java applications to implement messaging using a standard API, thereby removing the dependency on any particular messaging provider. In Part 1 of this session we will take a look at some of the basics of messaging, including sending and receiving messages, message types, and request/reply messaging. I will begin the session by going over the basics of messaging and the JMS API. Then, through interactive coding using OpenJMS I will demonstrate how to connect to JMS providers, send messages, receive messages, and use message properties. Please note that this is a two part session.

### **Enterprise Messaging With JMS (Part 2) by Mark Richards**

In Part 1 of the JMS session I covered messaging models, messaging basics, the JMS API, and point-to-point messaging. In this interactive code-intensive session I will cover some additional JMS topics such as browsing queues, load balancing, publishing and subscribing to messages within the pub/sub model, durable and non-durable subscribers, message selectors, and message filtering. I will also discuss and demonstrate message prioritization, persistent and non-persistent messages, and finally message expiration (expiry). Note that this is Part 2 of a two-part JMS session. **Prerequisite:** *Enterprise Messaging With JMS (Part 1) or some knowledge of JMS*

### **EJB3 Core Specification (JSR-220) by Mark Richards**

EJB3 (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will explore in detail some of the new features of the core EJB 3 specification. Included in this session will be a hands-on discussion and demonstration of session beans, dependency injection, interceptors (aop), and Message-Driven Beans (MDB). For the interceptors discussion I will be showing how to define interceptors for enabling a method trace, mocking objects, and sending JMS message notifications to be later picked up by the MDBs I will be creating. During the session I will demonstrate the new features of EJB 3 through interactive coding examples. Note: this session does not cover the new Java Persistence API (JPA) - only the core specification.

### **Transaction Design Patterns by Mark Richards**

Most web-based applications rely solely on the database to manage transactions, thereby freeing the developer from having to worry about transaction management. While this works in some circumstances, there are times when the use of transactions is vital to the integrity and operations of an application and its corresponding data. In this session I will demonstrate through real-world coding examples why transactions are such a critical part of the application development process. I will review the basics of both programmatic and declarative transactions, then introduce three transaction design patterns and explain when they should be applied, how to use them, and what problems they solve. By the end of this session you will see that by using transaction design patterns you can build an effective transaction management strategy for your application with very little effort.

### **Enterprise JPA - Tips and Tricks for JEE5 Persistence by Pratik Patel**

As with many technologies, the basics are easy. The hard part comes when the developer needs to do sophisticated integration, development, and testing as part of an enterprise application. A large enterprise application requires the developer to think of issues that affect the development, scalability and robustness of the application. This presentation will cover the advanced topics described below. A large enterprise application often will have several sub-projects that each contain their own JPA persistence unit. This opens up a number of questions around how to organize the persistence units and how the code between

sub-projects should interoperate. Developers will gain insight into these issues and will see a couple of solutions using live code examples.

### **Real-world JEE performance tuning: Tips n' Tricks by Pratik Patel**

Performance tuning any application is a black art that can consume much time. Fortunately, Java has many tools that can aid in this effort. There also are a number of basic tips that can help to analyze and fix performance problems. The Java memory model is usually something that you don't need to tune, but for high performance applications it is necessary to tweak. While there are a number of advanced things that can be done to performance tune an application, we'll discover that the simple, basic things are all that are usually needed to make your apps fly.

### **Choosing the right RIA platform for your project by Richard Monson-Haefel**

With literally hundreds of RIA products (e.g., Adobe Flash, Nexaweb, Backbase) and open source Ajax projects (e.g. Dojo, GWT, Prototype) to choose from. Picking the right RIA technology for the job requires months of research. Richard Monson-Haefel has been researching and writing about RIA alternatives for three years and has already done the research so you don't have to.

### **10 Things Every Software Architect Should Know by Richard Monson-Haefel**

An effective software architect understands that every application is different and requires unique choices regarding programming language, middleware, integration, data access, user interface design, etc. Richard Monson-Haefel has distilled knowledge from his own experience and from personal interviews with the World's best software architects to define 10 principles every software architect should know in order to be effective.

### **Rise of the Fit Client by Richard Monson-Haefel**

The world of desktop and web client software is about to change and for the better. Fit Clients, application platforms that combine the capabilities of Web 1.0, RIA, desktop widgets, and client/server technologies, are gaining mind share quickly and for good reason. Fit Client platforms such as Adobe AIR, Curl Nitro, Google Gears, and Mozilla Prism promise web developers the power of client/server applications, the ease of management and access of Web 1.0 applications, the cross-platform capabilities of RIA, and the holistic environment of desktop widget platforms.

### **Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy by Scott Davis**

There are wild-eyed radicals out there telling you that Java is dead, statically-typed languages are passe, and your skills are hopelessly out-of-date. Those extremists are the same ones who don't bat an eye at throwing out years of experience to learn a new language from scratch, pushing aside a familiar IDE for a new one, and deploying to a whole new set of production servers with little regard to legacy integration. While this "burn the boats" approach to software development might sound exciting to some folks, it's giving your manager the cold shakes right now. What if I told you that there was a way that you could integrate seamlessly with your legacy Java code, continue to use your trusty IDE and stable production servers, and yet take advantage of many of the exciting new dynamic language features that those fanatics keep prattling on about? You'd probably say, "Groovy!" I would, too...

### **Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind by Scott Davis**

This talk focuses on the ways that Groovy can turn a traditional Java developer's world-view upside down. We'll start by talking about how you can thumb your nose at The Man by leaving out many of the main syntactic hallmarks of Java: semicolons, parentheses, return statements, type declarations (aka Duck-typing), and the ever-present try/catch block. Then we'll look at features like operator overloading and method pointers that Groovy welcomes back into the language with open arms.

### **Rapid Web Development with Grails and Ajax by Scott Davis**

Grails is a Java- and Groovy-based web framework that is built for speed. First-time developers are amazed at how quickly you can get a page-centric MVC web site up and running thanks to the scaffolding and convention over configuration that Grails provides. Advanced web developers are often pleasantly surprised at how easy it is to break out of that coarse-grained navigation model using the native Ajax support baked into the framework.

### **YSlow: Building Your Website for Speed by Scott Davis**

How optimized is your website? YSlow, a FireFox/FireBug plugin, doesn't pull any punches. It gives any website an A, B, C, D, or F rating based on 14 individual analysis points. You'll be amazed (or depressed) at what YSlow thinks of your site. In this talk, we'll walk through these points step by step, learning what Yahoo! (the creator of this utility) does to keep its web properties running as quickly as possible.

### **GIS for Web Developers: Adding Where to Your Application by Scott Davis**

Based on the book GIS for Web Developers, this talk demonstrates how you can build your own Google Maps in-house using nothing but open source software. The Portland, Oregon Transit Authority recently migrated from a proprietary web mapping solution to the suite of 100% free and open source software discussed in this book. We look at Java-based clients, Java-based servers, and everything in between. We also discuss integrating free, public domain data from sources like the US Census Bureau and the USGS. If you're looking for real-world examples of AJAX in use, you'll find it here. If you're looking for real-world examples of web services in use, you'll find it here.

### **Introduction to Hibernate by Scott Leberknight**

This session introduces the Hibernate Object/Relational Mapping (ORM) framework, showing the basics of persisting Java objects to relational databases. No prior knowledge of Hibernate or ORM is assumed.

### **Real World Hibernate Tips by Scott Leberknight**

Hibernate is a very powerful object/relational mapping framework. With the vast amount of power also comes the responsibility to choose which features of Hibernate to use and how to use them, as well as things to avoid. We'll look at some real world Hibernate tips and tricks in this session.

### **High Performance Projects by Srivaths Sankaran**

Unsuccessful projects dominate the landscape of IT projects by a wide margin. Retrospectives are common occurrences and the advent of newer project delivery techniques hasn't proven to be the miracle cure. Some projects have lemon written all over them at the time of arrival and others just wither away for want of proper care and feeding. This talk introduces you to a variety of project problems and how to address them. You will learn techniques that will get you off on the right foot and habits that will ensure smooth progress throughout the project's lifecycle. The net result will be a project that hits its target and a customer that is happy.

### **Shining a Light on Eclipse by Srivaths Sankaran**

The Eclipse IDE has matured over several years to become quite a powerful tool with a robust plugin-based architecture. It gives commercial IDEs such as IDEA a run for their money. Yet, most developers barely scratch the surface of its capabilities. This talk will expose you to features that will greatly enhance your productivity. You will learn tips and tricks that will make you more efficient in your development activities and allow you to focus on solving the business problem at hand. Along the way you will be introduced to improvements introduced by the recent Ganymede release.

### **Java.next #1: Common Ground by Stuart Halloway**

In this talk, we will explore and compare four of the most interesting new JVM languages: Clojure, Groovy, JRuby, and Scala. Each of these languages aims to greatly simplify writing code for the JVM, and all of them succeed in this mission. However, these languages have very different design goals. We will explore these differences, and help you decide when and where these languages might fit into your development toolkit. For more information see <http://blog.thinkrelevance.com/2008/8/4/java-next-common-ground>.

### **Java.next #3: Dispatch by Stuart Halloway**

Dispatch takes many forms. Single dispatch, switch statements, pattern matching, and multiple dispatch all meet similar needs: Selecting runtime behavior in response to varying runtime conditions. Flexible dispatch is a key element of Java.next. All of the Java.next languages support dispatch strategies that are far more flexible than Java's single dispatch. In this talk (Part 3 of the Java.next series), I will explore how the Java.next languages (Clojure, Groovy, JRuby, and Scala) support dispatch.

### **How to Fail with 100% Code Coverage by Stuart Halloway**

Over the last few years, we have taken dozens of projects to 100% coverage, and there are still plenty of things that can go wrong. We will look at examples the various problems, and show how to prevent them from infecting your project.