

Twin Cities Software Symposium

Hilton Minneapolis/Bloomington Hotel

March 14 - 16, 2008

<http://www.nofluffjuststuff.com/conference/minneapolis/2008/03/index.html>

(event schedule as of March 13, 2008)

| Fri, Mar. 14, 2008 | | | | | |
|--------------------|--|---|---|---|---|
| | Salon A | Salon B | Salon C | Washington | Jefferson |
| 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 | JavaServer Faces: A Whirlwind Tour David Geary | 10 Things Every Software Architect Should Know Richard Monson-Haefel | Give it a REST Brian Sletten | Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson |
| 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 | Facelets David Geary | Techniques 2008 Jared Richardson | REST - Live! Brian Sletten | Developing Rich Internet Applications Richard Monson-Haefel |
| 4:45 - 5:00 PM | BREAK | | | | |
| 5:00 - 6:30 PM | Grails for (Recovering) Struts Developers: A Groovy Alternative Scott Davis | Seam David Geary | Agile Software Testing Strategies Jared Richardson | Resource-Oriented Computing w/ NetKernel : Software for the 21st Century Brian Sletten | Understanding Open Source Licensing Richard Monson-Haefel |
| 6:30 - 7:15 PM | DINNER | | | | |
| 7:15 - 8:00 PM | Keynote: by Neal Ford | | | | |

| Sat, Mar. 15, 2008 | | | | | |
|--------------------|--|---|---|--|---|
| | Salon A | Salon B | Salon C | Washington | Jefferson |
| 8:00 - 9:00 AM | BREAKFAST | | | | |
| 9:00 - 10:30 AM | Evolutionary SOA Neal Ford | Test Driven Everything David Hussman | YSlow: Building Your Website for Speed Scott Davis | 10 Tips for Getting Your Project Back on Track Jared Richardson | Resource-Oriented Groovy Brian Sletten |
| 10:30 - 11:00 AM | BREAK | | | | |
| 11:00 - 12:30 PM | Test Driven Design Neal Ford | Rich Faces David Geary | Groovy, Grails and Google Maps: Mashups 101 Scott Davis | Shippers Unite! Jared Richardson | What's Going On? : Complex Event Processing w/ Esper Brian Sletten |
| 12:30 - 1:15 PM | LUNCH | | | | |
| 1:15 - 2:15 PM | EXPERT PANEL DISCUSSION | | | | |
| 2:15 - 3:45 PM | The Busy Java Developer's Guide to Concurrency (Part 1: Threads) Ted Neward | Introduction to JRuby Neal Ford | Filthy Rich Clients with the Google Web Toolkit, Part I David Geary | Project Smells Nathaniel Schutta | Transaction Design Patterns Mark Richards |
| 3:45 - 4:00 PM | BREAK | | | | |
| 4:00 - 5:30 PM | The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) Ted Neward | "Design Patterns" in Dynamic Languages Neal Ford | Filthy Rich Clients with the Google Web Toolkit, Part II David Geary | The 90-Minute Startup Michael Nygard | Enterprise Messaging Using JMS Mark Richards |

| Sun, Mar. 16, 2008 | | | | | |
|--------------------|---|--|--|--|--|
| | Salon A | Salon B | Salon C | Washington | Jefferson |
| 8:00 - 9:00 AM | BREAKFAST | | | | |
| 9:00 - 10:30 AM | The Busy Java Developer's Guide to Debugging Ted Neward | JavaScript: the Good, the Bad, and the Ugly Nathaniel Schutta | Agile, Smagile: What's Working? - What's Not? David Hussman | Tactical Continuous Integration with Hudson Andrew Glover | Java Persistence: Approaching the Silver Bullet Mark Richards |
| 10:30 - 11:00 AM | MORNING BREAK | | | | |
| 11:00 - 12:30 PM | The Busy Java Developer's Guide to Monitoring Ted Neward | Dojo: Getting Started Nathaniel Schutta | Architecture and Agility Are Not Mutually Exclusive David Hussman | Groovin' builds Gant get any easier Andrew Glover | EJB3 Core Specification (JSR-220) Mark Richards |
| 12:30 - 1:15 PM | LUNCH | | | | |
| 1:15 - 2:15 PM | EXPERT PANEL DISCUSSION | | | | |
| 2:15 - 3:45 PM | SOA Unplugged Mark Richards | Designing for Ajax, part 1 Nathaniel Schutta | Leading Agile Projects: Finding Your Groove in the First 4 Iterations David Hussman | Failures Come In Flavors (part 1) Michael Nygard | The Busy Java Developer's Guide to Native Code Ted Neward |
| 3:45 - 4:00 PM | BREAK | | | | |
| 4:00 - 5:30 PM | Easy BDD with Groovy Andrew Glover | Designing for Ajax, part 2 Nathaniel Schutta | Leading Agile Projects: Maintaining Sustainable Agility David Hussman | Failures Come In Flavors (part 2) Michael Nygard | The Busy Developer's Guide to Scala Ted Neward |

Twin Cities Software Symposium

Hilton Minneapolis/Bloomington Hotel

March 14 - 16, 2008

<http://www.nofluffjuststuff.com/conference/minneapolis/2008/03/index.html>
(event schedule as of March 13, 2008)

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.

Groovin' builds Gant get any easier by Andrew Glover

There's no question that Ant is the de facto standard for building Java applications; however, even its creator has acknowledged an inherent limitation with Ant's expressiveness due to its reliance on XML. Recently, the popularity of Ruby and the Rails framework has brought to focus Ruby's de facto build platform: Rake. Rake's expressiveness comes from its reliance on Ruby itself to define a DSL for software assembly. While Rake's ultimate focus is Ruby, there are a number of interesting projects that utilize expressive DSLs for building Java including Gant, which uses Groovy as a DSL format and builds upon Ant's existing cornucopia of tasks.

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.

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.

REST - Live! by Brian Sletten

You've read the articles, the books, the Ph.D. thesis and all of the meta-commentary about building RESTful APIs, but you're still not sure where to begin. This is an interactive session with almost no slides. You should come prepared to pair program with me and everyone else in the room. We will tackle any reasonable suggestions of what might benefit from a RESTful interface. We will generate working code and explore the iterative process by which good REST-oriented APIs are created. Bring your ideas for open source projects that we might want to expose through a resource-oriented model.

Resource-Oriented Computing w/ NetKernel : Software for the 21st Century by Brian Sletten

Imagine the simplicity of REST married to the power of Unix pipes with the benefits of a loosely-coupled, logically-layered architecture. If that is hard to imagine, it may be because the architectures available to you today are convoluted accretions of mismatched technologies, languages, abstractions and data models. NetKernel is a disruptive technology that changes the game. It has been quietly gaining mind share in the past several years; people who are exposed to it don't want to go back to the tired and blue conventions of J2EE and .NET. Not only does it make building the kinds of systems you are building today easier, it does it more efficiently, with less code and a far more scalable runway to allow you to take advantage of the emerging multi-core, multi-CPU hardware that is coming our way. Come see how this open source / commercial product can change the way you think about building software.

Resource-Oriented Groovy by Brian Sletten

Resource-Oriented Groovy Ok, we've got our agile methodologies, our test-driven development, our dynamic scripting languages and, what? A static data model? Relational systems that need to have the relationships

spelled out for them? What is wrong with this picture? Our information systems are increasingly dynamic, we shouldn't let our data slow us down.

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.

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.*

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. Two frameworks. And then you have to learn to use them together. Why do we have to learn two frameworks just to retrieve "Hello World" from a database and show it in a view. Isn't that crazy? Now you can use one framework, and use one component model. One. Isn't that nice? Seam, a framework built on JSF and EJB3.0, unifies the JSF and EJB component models. Seam is a steam roller, quickly gathering market share among JSF newbies and longtime believers alike. Come see what it's all about. **Prerequisite:** *Some knowledge of JSF is required. If you don't know what a managed bean is, for instance, then attend JSF Whirlwind before this session.*

Rich Faces by David Geary

This talk explores the RichFaces Ajax framework, which is really two frameworks: Ajax4jsf and RichFaces components. In this session you will see how to implement low-level Ajax functionality using Ajax4JSF, and how to use high-level Ajax components from RichFaces. **Prerequisite:** *Some knowledge of JSF is required, in addition to familiarity with Ajax.*

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.

Test Driven Everything by David Hussman

Why do we wait to test? Of course when you read this your thoughts went to testing code. While we still wait to test code and products early, we also wait to test ideas, projects, product direction, meetings and more. This session will show you (or challenge you) to think about test driven beyond the coding realm. You will be doing some thinking and talking and other things that involve more than just listening to someone blather with slides for 90 minutes.

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.

Leading Agile Projects: Finding Your Groove in the First 4 Iterations by David Hussman

Although there are many books about agile, but few provide a path for guiding you through the beginning of an agile project. Whether you are preparing for your first agile project, or taking the lead for the first time, this session will provide a guided tour filled with practical advice and a pile of anecdotes.

Leading Agile Projects: Maintaining Sustainable Agility by David Hussman

Once your agile project is rolling along, there are many bumps and roadblocks which can derail the train. Whether you are leading the project formally or informally, there are techniques you can use to keep the project alive and innovative. This session will cover skills and techniques for leading sustainable project communities. **Prerequisite:** *Leading Agile Projects: Finding Your Groove in the First 4 Iterations*

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.

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.

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.

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.

Shippers Unite! by Jared Richardson

An overview of the Agile software approach from the book Ship It! A Practical Guide to Successful Software Projects.

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 Messaging Using JMS 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 this session we will take a look at some of the basics of messaging, including sending and receiving messages, message types, message headers, request/reply, and pub/sub messaging. In the first part of this session I will go over the basics of messaging and the JMS API. Then, in the second part of the session, I will abandon the presentation slides and through interactive coding using OpenJMS I will demonstrate how to send messages, receive messages, use message properties, and how to use the different message types (Text, Object, Map, Bytes, and Stream). At the close of the session I demonstrate how to use the pub/sub model and show how to code durable and non-durable subscribers.

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.

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 defining and accessing session beans, dependency injection, declarative security, interceptors (aop), and Message-Driven Beans (MDB). For those of you who still like to write XML, I will also discuss and show how we can use XML rather than annotations within EJB3. 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.

SOA Unplugged by Mark Richards

Awareness about Service Oriented Architecture (SOA) has grown significantly in the past several years. Unfortunately, along with that growth has come a significant amount of confusion about what SOA really is. SOA has become such a ubiquitous buzzword that it now has many faces and means different things to different people. CIO's, managers, vendors, business users, architects, and developers all see SOA differently which creates a sea of confusion about what is and isn't SOA. In this highly interactive and thought provoking session we will look beyond the hype and marketure of SOA and explore SOA from an architecture and development point of view - in other words, SOA as an architecture pattern. During this session we will look at SOA use cases, services, integration, implementation, guiding architecture principles of SOA, and attempt to answer the following question: What is and isn't SOA?

The 90-Minute Startup by Michael Nygard

What do you get when you add agile programming, automated deployment, self-describing systems, and virtualization? In this session, Michael will create and deploy a fully-functional web site. By the end of 90 minutes, you will be able to access the site live on the 'Net.

Failures Come In Flavors (part 1) by Michael Nygard

The typical JEE application does not reach the fabled "five nines" of availability. Far from it. It's more like "double eights". Come see why enterprise applications and web sites are only serving users 88% of the time instead of 99.999%. Part 1 of 2

Failures Come In Flavors (part 2) by Michael Nygard

What can we do about the dismal uptime of typical applications? We are asked to provide "five nines", but only reach 88%, on average. Come learn how to prevent the Stability Antipatterns from biting you. Apply these Stability Patterns to contain damage, recover from shocks, and survive disasters. Part 2 of 2

Project Smells by Nathaniel Schutta

We all know that code can have a certain...odor but frankly so can projects. Everyone has their favorite horror story or tale of a death march. In this talk, we'll discuss common project smells and what you as a developer can do to maintain your sanity - and your hair line!

JavaScript: the Good, the Bad, and the Ugly by Nathaniel Schutta

Thanks to Ajax, JavaScript is cool again and developers are taking a second look at this much maligned language. This session will give you an overview of this misunderstood language as well as opening your eyes to some of the excellent tools available to ease the pain of developing in this dynamic language.

Dojo: Getting Started by Nathaniel Schutta

So you want to do some Ajax and you've rightly concluded that you don't want to build your own library. After some thought, you've settled on using Dojo - but you're not sure how to get going. This talk will introduce Dojo and discuss several ways that Ajax can improve your new or existing application.

Designing for Ajax, part 1 by Nathaniel Schutta

So you've convinced the boss that your new web application just has to have Ajax...but now what? With dozens of libraries making even the most blinkish of interactions trivial, how do you decide where to sprinkle the magic Ajax dust? This talk will give a plain old boring "web 1.0" an Ajax facelift with a focus on improving the user experience providing you with a game plan for introducing Ajax to your world.

Designing for Ajax, part 2 by Nathaniel Schutta

We'll pick up where Part 1 left off working in even more advanced approaches such as offline support with Google Gears.

Keynote: Ancient Philosophers & Blowhard Jamborees by Neal Ford

It turns out that ancient philosophers knew a lot about software -- did you know that Plato defined object-oriented programming? This keynote applies old lessons to new problems and old problems to new lessons. It describes why SOA is so hard, and why people in your company make bone-headed decisions. What other keynote includes Rube Goldberg, Aristotle, Dave Thomas, and Chindia?

Evolutionary SOA by Neal Ford

This session demonstrates that "Agility" and "SOA" complement each other quite well. Just because SOA is buzz-word compliant doesn't mean that you should throw good practices out the window. This session demonstrates how you can apply the principles of agility to building highly complex distributed enterprises.

Test Driven Design by Neal Ford

Most developers think that "TDD" stands for Test-driven Development. But it really should stand for "Test-driven Design". Rigorously using TDD makes your code much better in multiple ways.

Introduction to JRuby by Neal Ford

This session describes JRuby, the 100% pure-Java implementation of the Ruby programming language. It covers the basics of programming with JRuby and examples of how to integrate it into existing Java projects.

"Design Patterns" in Dynamic Languages by Neal Ford

The Gang of Four book should have been entitled "Palliatives for Statically Typed Languages", because the recipes it provides are cumbersome solutions to the problems it poses. Using powerful languages makes the

solutions in the GoF book look hopelessly complicated. This session shows how to solve the same problems concisely, elegantly, and with far fewer lines of code using the facilities of dynamic languages.

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.

Developing Rich Internet Applications 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 two years and has already done the research so you don't have to.

Understanding Open Source Licensing by Richard Monson-Haefel

What does GPL, LGPL, MIT, Apache licenses, copyleft, and dual licensing mean? Richard Monson-Haefel explains both the legal and technical implications of the major open source licenses in plain English. He explains when and how you can use open source in the enterprise and in the development of software products and how to protect your organization from abusing open source licensing.

Grails for Struts Developers: A Groovy Alternative by Scott Davis

Struts is the defacto web framework for Java web developers. It has been with us since 2001. Struts enjoys unprecedented success -- most surveys place its market share between 60% and 70%. It introduced a whole generation of web developers to the phrase 'MVC' (Model / View / Controller). Grails 1.0 was released in 2008. It marries the modern features of Rails with the need for legacy support for Struts. Grails is inspired by Rails, but it is not a simple port of the project to Java. It takes the ideas of Rails, but expresses them in familiar Java libraries like Spring and Hibernate. It also leverages a new dynamic language for the JVM called Groovy.

Groovy, Grails and Google Maps: Mashups 101 by Scott Davis

Groovy is a new dynamic language that dramatically speeds up Java development. Grails is a complete web framework in a box, including a web server and a database. Google Maps allows you to add maps to your webpage in a few lines of code. Put all three together and you are built for speed.

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.

Grails for (Recovering) Struts Developers: A Groovy Alternative by Scott Davis

Struts enjoys an unprecedented marketshare in the Java web development space -- 60%-70% according to most surveys. As newer, modern web frameworks come to the scene, very little attention is paid to the real costs of migrating an existing Struts application. This talk shows you ways to mix Groovy into a legacy

Struts application, dramatically reducing both the lines of code and the complexity. We'll also introduce you to Grails (a Groovy-based web framework) whose URL-mapping capabilities allow it to replace your Struts application without breaking legacy URLs.

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

This talk focuses on integrating Groovy with your legacy Java codebase in a way that wouldn't raise an eyebrow in the most conservative of organizations. We'll look at the dramatic reduction in line of code you can achieve by simply flipping your POJOs to POGOs (Plain Old Groovy Objects). We'll talk about calling Java classes from Groovy, and calling Groovy classes from Java. We'll look at Groovyc, the integrated compiler that manages Groovy/Java dependencies without a hiccup.

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.

Groovy, Grails and Google Maps: Mashups 101 by Scott Davis

Groovy is a new dynamic language that dramatically speeds up Java development. Grails is a complete web framework in a box, including a web server and a database. Google Maps allows you to add maps to your webpage in a few lines of code. Put all three together and you are built for speed.

The Busy Java Developer's Guide to Concurrency (Part 1: Threads) by Ted Neward

Java's threading capabilities took a serious turn for the better with the release of Java5, thanks to the incorporation of the `java.util.concurrent` packages, a set of pre-built components for thread pooling and execution, synchronization, and more.

The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) by Ted Neward

Java's threading capabilities have been a part of the Java platform since its inception, yet for many Java developers, using Threads still remain a dark and mysterious art, and synchronization beyond the use of the "synchronized" keyword is almost unknown. *Prerequisite: The Busy Java Developer's Guide to Concurrency (Part 1: Threads)*

The Busy Java Developer's Guide to Debugging by Ted Neward

Bugs? We all know your code has no bugs, but someday, you're going to find yourself tracking down a bug in somebody else's code, and that's when it's going to be helpful to have some basic ideas about bug-tracking in your toolbox. Learn to make use of the wealth of tools that the Java Standard Platform makes available to you--tools that your IDE may not know exist, tools that you can make use of even within a production environment.

The Busy Java Developer's Guide to Monitoring by Ted Neward

Crashes? Outages? Slow response? We all know that it's never your code that causes these things, but for some reason those pesky system administrators still insist on paging you at 4AM to come in and fix those problems, anyway. For some reason, they just keep expecting you to support this thing, even after QA said it was OK!

The Busy Java Developer's Guide to Native Code by Ted Neward

As much as the Java Virtual Machine and libraries provide a comfortable womb in which to write code, moments appear in every Java developer's life when they just have to call down to code that exists at the native, C-executable, level. Java provides a standard API for doing this--Java Native Interface--but its use is at once both darkly mysterious and... well... painful.

The Busy Developer's Guide to Scala by Ted Neward

Scala is a new programming language incorporating the most important concepts of object-oriented and functional languages and running on top of the Java Virtual Machine as standard "dot-class" files.