

# Rocky Mountain Software Symposium

Denver Marriott South @ Park Meadows

May 29 - 31, 2009

<http://www.nofluffjuststuff.com/conference/denver/2009/05/index.html>

## Fri, May. 29, 2009

|                 | Salon A  | Salon B   | Lonetree  | Littleton   |
|-----------------|--|---|---|---|
| 12:00 - 1:00 PM | REGISTRATION   |   |   |   |
| 1:00 - 1:15 PM  | WELCOME  |   |   |   |
| 1:15 - 2:45 PM  | Emergent Design & Evolutionary Architecture<br>Neal Ford | Effective Concurrent Java<br>Brian Goetz        | The Amazing Groovy Weight-loss Plan<br>Scott Davis  | The Busy Java Developer's Guide to Java7<br>Ted Neward                      |
| 2:45 - 3:15 PM  | BREAK  |   |   |   |
| 3:15 - 4:45 PM  | Real-world Refactoring<br>Neal Ford                      | The Java Memory Model<br>Brian Goetz            | Groovy XML Ninja Skills<br>Scott Davis  | The Busy Java Developer's Guide to Java Platform Security<br>Ted Neward     |
| 4:45 - 5:00 PM  | BREAK  |   |   |   |
| 5:00 - 6:30 PM  | Test Driven Design<br>Neal Ford                          | Are All Web Applications Broken?<br>Brian Goetz | Dim Sum Grails: A Sampler of Practical Non Database-Driven Grails Applications<br>Scott Davis | The Busy Java Developer's Guide to Advanced Platform Security<br>Ted Neward |
| 6:30 - 7:15 PM  | DINNER   |   |   |   |
| 7:15 - 8:00 PM  | Keynote: by Neal Ford                                    |   |   |   |

## Sat, May. 30, 2009

|                  | Salon A                                   | Salon B   | Lonetree   | Littleton  |
|------------------|---|---|--|--|
| 8:00 - 9:00 AM   | BREAKFAST                                 |   |  |  |
| 9:00 - 10:30 AM  | JSF 2.0: An Introduction<br>David Geary   | The Productive Programmer: Mechanics<br>Neal Ford | Garbage-collector-friendly programming<br>Brian Goetz        | Groovy Testing<br>Scott Davis                                    |
| 10:30 - 11:00 AM | BREAK                                     |   |  |  |
| 11:00 - 12:30 PM | JSF 2.0: Advanced Topics<br>David Geary   | What's New in Spring 3<br>Ken Sipe                | Inside the JVM<br>Brian Goetz                                | Web 2.0 Checklist: Deconstructing Modern Websites<br>Scott Davis |
| 12:30 - 1:30 PM  | LUNCH                                     |   |  |  |
| 1:30 - 3:00 PM   | Flex for Java Developers<br>David Geary   | Architecture and Scaling<br>Ken Sipe              | The Busy Java Developer's Guide to Collections<br>Ted Neward | Seven Habits of Highly Dysfunctional Teams<br>Nathaniel Schutta  |
| 3:00 - 3:15 PM   | BREAK                                     |   |  |  |
| 3:15 - 4:45 PM   | Mastering Maven 2.0<br>Matthew McCullough | So you want to be an Architect<br>Ken Sipe        | IZero: Starting Projects Right<br>Stuart Halloway            | JavaScript: the Good, the Bad, and the Ugly<br>Nathaniel Schutta |
| 4:45 - 5:45 PM   | BIRDS OF A FEATHER SESSION                |   |  |  |

## Sun, May. 31, 2009

|                  | Salon A   | Salon B   | Lonetree                                     | Littleton                                      |
|------------------|---|---|--|--|
| 8:00 - 9:00 AM   | BREAKFAST   |   |  |  |
| 9:00 - 10:30 AM  | Taking Agile From Tactics to Strategy<br>Stuart Halloway                      | 7 Habits of Highly Effective Developers<br>Ken Sipe                       | Programming Scala<br>Venkat Subramaniam      | GWT: An Introduction<br>David Geary            |
| 10:30 - 11:00 AM | MORNING BREAK   |   |  |  |
| 11:00 - 12:30 PM | REST : Information-Driven Architectures for the 21st Century<br>Brian Sletten | Open Source Debugging Tools, A 90 Minute Bootcamp<br>Matthew McCullough   | Effective Java<br>Venkat Subramaniam         | GWT: Advanced Topics<br>David Geary            |
| 12:30 - 1:15 PM  | LUNCH   |   |  |  |
| 1:15 - 2:15 PM   | EXPERT PANEL DISCUSSION   |   |  |  |
| 2:15 - 3:45 PM   | SPARQL: Querying the Data Web<br>Brian Sletten                                | Hacking Your Brain for Fun and Profit<br>Nathaniel Schutta                | Hacking - The Dark Arts<br>Ken Sipe          | Java.next #1: Common Ground<br>Stuart Halloway |
| 3:45 - 4:00 PM   | BREAK   |   |  |  |
| 4:00 - 5:30 PM   | Semantic SOA : Meaningful Service Strategies<br>Brian Sletten                 | iPhone Objective-C integration to Java Web Services<br>Matthew McCullough | Cleaning up Code Smell<br>Venkat Subramaniam | Programming Clojure<br>Stuart Halloway         |

# Rocky Mountain Software Symposium

Denver Marriott South @ Park Meadows

May 29 - 31, 2009

<http://www.nofluffjuststuff.com/conference/denver/2009/05/index.html>

## **The Amazing Groovy Weight-loss Plan by Scott Davis**

"The central enemy of reliability is complexity." (Dr. Daniel Geer) Java is a powerful programming language. A smart developer can do nearly anything with Java. So the next question is, "How quickly can it be done? How many lines of code does it take to do common tasks?" Groovy greases the wheels of Java by decreasing the complexity of the language while preserving the raw power. At first glance, you might think that this talk is simply about how Groovy drastically reduces the lines of code you need to write. What this talk is really about is bringing simplicity, clarity, readability, and yes, beauty to your source code.

## **Groovy XML Ninja Skills by Scott Davis**

"XML is like violence: if it doesn't solve your problem, you aren't using enough of it." (Anonymous) XML is everywhere. Whether you are dealing with local configuration files (web.xml, struts-config.xml) or remote web services (SOAP, REST, RSS, Atom), the modern software developer needs to be able to request, slice, and dice XML with ease. That requires a set of razor-sharp tools that reduce the inherent complexity of the problem, not multiply it. Once you see XML tremble in fear at the awesome power of Groovy, you'll wonder what you ever did without it.

## **Dim Sum Grails: A Sampler of Practical Non Database-Driven Grails Applications by Scott Davis**

"The proof of the pudding is in the eating. By a small sample we may judge of the whole piece." (Miguel de Cervantes Saavedra) Most Grails tutorials demonstrate how easy it is to build simple CRUD (Create/Retrieve/Update/Delete) applications. While skinning a database with a web front-end is undeniably one beneficial aspect of Grails, it isn't the only thing Grails is good for. As you'll see here, Grails can be used to build a wide variety of web applications. You won't see a single HTML table with "edit" and "delete" links, I promise.

## **Groovy Testing by Scott Davis**

"Tests don't break things; they dispel the illusion that it works." (Anonymous) In this era of "Test-First" and "Test-Driven" development, the modern software engineer knows that testing is no longer an optional part of the process. You need to have the best tools at your fingertips: a set of utilities that maximize your results with a minimum of effort. Groovy offers Java developers an optimal set of testing tools.

## **Web 2.0 Checklist: Deconstructing Modern Websites by Scott Davis**

"The challenge of modernity is to live without illusions and without becoming disillusioned." (Antonio Gramsci) There are plenty of sarcastic "Web 2.0" checklists out there -- be perpetually in BETA, when in doubt add rounded corners, etc. While we can all laugh at the superficial aspects of the Web 2.0 revolution, there are plenty of serious aspects to it as well. Is your website mash-up friendly or hostile? Do you tell your visitors when things change (via RSS or Atom syndication), or do you expect them to check in daily for updates? Is your website a silo or a part of a larger ecosystem?

## **Emergent Design & Evolutionary Architecture by Neal Ford**

Most of the software world has realized that BDUF (Big Design Up Front) doesn't work well in software. But lots of developers struggle with this notion when it applies to architecture and design. Surely you can't just start coding, right? You need some level of understanding before you can start work. This session describes the current thinking about emergent design & evolutionary architecture, including both proactive (test-driven development) and reactive (refactoring, composed method) approaches to discovering design. The goal of this talk is to provide nomenclature, strategies, and techniques for allowing design to emerge from projects as they proceed, keeping you code in sync with the problem domain.

## **Real-world Refactoring by Neal Ford**

Refactoring is a fine academic exercise in the perfect world, but we don't really live there. Even with the best intentions, projects build up technical debt and cruffy bad things. This session covers refactoring in the real world, at both the atomic level (how to refactor towards composed method and the single level of abstraction principle) to larger project strategies for multi-day refactoring efforts. This talk provides practical strategies for real projects to effectively refactor your code.

## **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.

### **Keynote: On the Lam from the Furniture Police by Neal Ford**

When you were hired by your current employer, you may think it's because of your winning personality, your dazzling smile, or your encyclopedic knowledge of [insert technology here]. But it's not. You were hired for your ability to sit and concentrate for long periods of time to solve problems, then placed in an environment where it's utterly impossible to do that! Who decides that, despite overwhelming evidence that it's bad for productivity and people hate it, that you must sit in a cubicle? The furniture police! This keynote describes the frustrations of modern knowledge workers in their quest to actually get some work done, and solutions for how to gird yourself against all those distractions. I talk about environments, coding, acceleration, automation, and avoiding repetition as ways to defeat the mid-guided attempts to sap your ability to produce good work. And I give you ways to go on the lam from the furniture police and ammunition to fight back!

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

Developers from the 1980s would be shocked at how inefficiently developers use their computers because of the advent of graphical operating systems. This talk describes how to reclaim productivity afforded by intelligent use of command lines and other ways of accelerating your interaction with the computer and bending computers to do your bidding. Stop working so hard for your computer!

### **JSF 2.0: An Introduction by David Geary**

This session introduces JSF 2.0 fundamentals, with emphasis on new features in JSF 2.0. **Prerequisite:** *Familiarity with JSF, or other component-based frameworks*

### **JSF 2.0: Advanced Topics by David Geary**

This session covers advanced aspects of JSF 2.0. **Prerequisite:** *Familiarity with JSF, or other component-based frameworks. Familiarity with Ajax. This session builds on demos shown in the JSF 2.0 Introduction talk, so it is helpful, although not required, if you attend the intro talk before coming to this session.*

### **Flex for Java Developers by David Geary**

An introduction to Flex for Java developers. **Prerequisite:** *Familiarity with Flex and at least one other web application framework*

### **GWT: An Introduction by David Geary**

An introduction to Google Web Toolkit. **Prerequisite:** *Familiarity with a component-based framework, preferably a desktop application framework*

### **GWT: Advanced Topics by David Geary**

Learn to do really cool stuff with GWT. **Prerequisite:** *The GWT: Introduction session is not a prerequisite for this session, but it will help if you have some familiarity with GWT.*

### **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.

### **The Java Memory Model by Brian Goetz**

What's the worst thing that can happen when you fail to synchronize in a concurrent Java program? Its probably worse than you think -- modern shared-memory processors can do some pretty weird things when left to their own devices.

### **Are All Web Applications Broken? by Brian Goetz**

Many developers believe that web frameworks "take care of" the details of concurrency, but this is only because most web applications make limited use of state. Stateful web applications also need to be careful

about hazards like races. This talk will use the Java Memory Model to analyze common patterns of state management in web applications. **Prerequisite:** *The Java Memory Model*

### **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.

### **Inside the JVM by Brian Goetz**

Ever wondered what happens to your bytecodes when they're executed by a Java Virtual Machine? This talk provides a peek "under the hood" of modern JVMs, exploring dynamic compilation, speculative optimization, garbage collection, and some hardware-specific optimizations.

### **IZero: Starting Projects Right by Stuart Halloway**

If an iteration is the heartbeat of an agile development process, then Iteration Zero (IZero) creates the heart. While you can (and should) retrospect and adjust throughout the software lifecycle, few things are as valuable as a good start. In this talk, you will learn how we run Iteration Zero at Relevance.

### **Taking Agile From Tactics to Strategy by Stuart Halloway**

Teams adopting agile should begin at a tactical level, but they shouldn't end there. The Agile Manifesto operates at many different levels. Learn to apply the principles of agile at a strategic level. Otherwise you can have a great agile ground game and still lose.

### **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>.

### **Programming Clojure by Stuart Halloway**

Find out why Clojure is Java.next: \* Clojure provides clean, fast access to all Java libraries. \* Clojure provides all the low-ceremony goodness you know and love from dynamic languages such as Ruby and Python. \* Clojure includes Lisp's signature feature: Treating code as data through macros. \* Clojure's emphasis on immutability and support for software transactional memory make it a viable option for taking advantage of massively parallel hardware.

### **Mastering Maven 2.0 by Matthew McCullough**

Maven has been on the Java build tools scene for quite a number of years, but the adoption rate in enterprises is now going through the roof. Maven can seem daunting, but this presentation will equip existing Maven users with more efficient techniques and tools to overcome the biggest perceived Maven hurdles and build issues with ease. We'll examine tools to help you find artifacts in central repositories, manage your corporation's internal Maven artifacts with a proxy tool such as Nexus, view and override dependency graphs, dependency management and multi-module best practices, create OS specific profiles, and leverage the latest Maven plugins for the top Java IDEs. **Prerequisite:** *Basic Maven knowledge*

### **Open Source Debugging Tools, A 90 Minute Bootcamp by Matthew McCullough**

Open Source is not just a suite of libraries you consume within your application, but now reaches into the space of tools to help you troubleshoot and improve your applications. This session will quickly survey a wide range of tools across the Java, Networking, Filesystem, SOAP, REST, HTML, CSS and JavaScript realms. We'll look at applications such as VisualVM, which help you analyze your heap and garbage collection cycles of both local and remote applications. Performance and load testing tools such as JMeter will expose bottlenecks, threading, and scalability concerns of everything from Java modules to Web Apps (even ones that don't use any Java). **Prerequisite:** *Basic knowledge of web services, core Java, web application design.*

### **iPhone Objective-C integration to Java Web Services by Matthew McCullough**

iPhone development is all the rage both in the mobile entertainment, social networking, and productivity application spaces. As a Java developer, prepare yourself to be a participant in aspects of this new breed and platform of development. Hop on board with a quick start to iPhone application coding in Objective C and integration with some of our favorite Java web service back-ends such as Axis, JSR311 Jersey, Grails, and more.

### **The Busy Java Developer's Guide to Java7 by Ted Neward**

Even though the Java 7 JSR has yet to be formed, some interesting things are beginning to emerge from Sun about what Java7 may include when its formal release contents are finally made public.

### **The Busy Java Developer's Guide to Java Platform Security by Ted Neward**

Permissions, policy, SecurityExceptions, oh my! The Java platform is a rich and powerful platform, complete with a rich and powerful security mechanism, but sometimes understanding it and how it works can be daunting and intimidating, and leave developers with the basic impression that it's mysterious and dark and incomprehensible. Nothing could be further from the truth, and in this presentation, we'll take a pragmatic, code-first look at the Java security platform, including Permissions, the SecurityManager and its successor, AccessController, the Policy class and policy file syntax, JAAS, and more.

### **The Busy Java Developer's Guide to Advanced Platform Security by Ted Neward**

So you know the platform security model, and now you want to use it in new and interesting ways, like creating a custom Policy implementation, a custom Permission, or create a custom security context in which code will execute. Perhaps you even wish to make certain objects accessible only to those with the right permissions, or cryptographic key. Nothing could be easier, despite Java security's reputation as a dark and arcane place. **Prerequisite:** *The Busy Java Developer's Guide to Platform Security*

### **The Busy Java Developer's Guide to Collections by Ted Neward**

For so many Java developers, the java.util.\* package consists of List, ArrayList, and maybe Map and HashMap. But the Collections classes are so much more powerful than many of us are led to believe, and all it requires is a small amount of digging and some simple exploration to begin to "get" the real power of the Collection classes.

### **Seven Habits of Highly Dysfunctional Teams by Nathaniel Schutta**

Being on a high performing team is a transcendent experience - unfortunately, many of us find more dysfunction than function. In this talk, we'll take a look at some of the common issues that face teams and discuss some ways of working towards a happy crew.

### **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.

### **Hacking Your Brain for Fun and Profit by Nathaniel Schutta**

The single most important tool in any developers toolbox isn't a fancy IDE or some spiffy new language - it's our brain. Despite ever faster processors with multiple cores and expanding amounts of RAM, we haven't yet created a computer to rival the ultra lightweight one we carry around in our skulls - in this session we'll learn how to make the most of it. We'll talk about why multitasking is a myth, the difference between the left and the right side of your brain, the importance of flow and why exercise is good for more than just your waist line.

### **What's New in Spring 3 by Ken Sipe**

The Spring Framework has led the industry in innovation for years. Starting with dependency injection and promoting testing through removal of framework dependencies. Spring 3.0 continues that innovation in a way that takes full advantage of the Java 5 platform. There are a number of significant changes to the framework. So whether you are new to the framework or an experience Spring developer, this is a great session to come up to speed on the latest from SpringSource. **Prerequisite:** *Java 5*

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

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

### **So you want to be an Architect by Ken Sipe**

This session is a quick look at all aspects of being a corporate software architect. Whether you are a developer looking to move into the role of architect, needing to have an understanding of what is expected or already in the role of software architect looking for new and interesting ideas, this session is for you.

### **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.

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

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

### **REST : Information-Driven Architectures for the 21st Century by Brian Sletten**

There is a shift going on in the Enterprise. While still used and useful, the promises of the SOAP/WSDL/UDDI Service-Oriented Architecture (SOA) stack have failed to live up to their promise. A new vision of linked information is enveloping online and Enterprise users. The REST architectural style is squarely behind this thinking as a way of achieving low-cost, flexible integration, increased data security, greater scalability and long-term migration strategies. If you have dismissed REST as a toy or are unfamiliar with it, you owe it to yourself to see what is so interesting about this way of doing things.

### **SPARQL: Querying the Data Web by Brian Sletten**

The human-friendly Web is about nicely-formatted, accessible content for users to browse. There is an emerging Data Web that relies on technologies from the Semantic Web stack to link increasingly rich connections between various data sources. SPARQL and RDF are the main tools for expressing and using this connectivity. This talk will introduce you to one of the practical and accessible aspects of employing these ideas on the Web and in the Enterprise. *Prerequisite: The Semantic Web: The Future, Now and Rich Web Pages : Publishing Semantic Content with GRDDL and RDFa would both be helpful but are not required*

### **Semantic SOA : Meaningful Service Strategies by Brian Sletten**

The goal for web services was always to reduce our burden by increasing the potential for reuse of business functionality. Somehow, we got lost along the way in a morass of confusing, unfulfilling and downright broken technologies. While we are interested in pursuing REST-based systems for managing information, we need some strategies for tying it all together sensibly. If we abandon WSDL, SOAP and UDDI, what do we replace them with? This talk will walk you through combining resource-oriented strategies with technologies from the Semantic Web to describe, find, and bind to services in dynamic, flexible and extensible ways.

*Prerequisite: The Semantic Web: The Future Now, Give it a REST and SPARQL : Querying the Data Web would all be helpful talks to have attended*

### **Programming Scala by Venkat Subramaniam**

Scala is a static fully object-oriented, functional language on the JVM. While taking advantage of the functional aspects, you can continue to make full use of the powerful JVM and Java libraries.

### **Effective Java by Venkat Subramaniam**

Java is a well established language, that has been around for more than a decade. Yet, programming on it has its challenges. There are concepts and features that are tricky. When you run into those, the compiler is not there to help you.

### **Cleaning up Code Smell by Venkat Subramaniam**

Projects often start out simple, but soon become complex and turn into a lose cannon. Organizations are struggling to maintain and evolve software. Poor code quality is a significant part of that problem. Improving the quality of code is critical to success of enterprise projects.