

Rocky Mountain Software Symposium

Renaissance Suites Flatiron Hotel

May 04 - 06, 2007

<http://www.nofluffjuststuff.com/sh/2007-05-denver>

Fri, May. 04, 2007					
	Flatiron Ballroom	Flagstaff	Red Rocks	Chautauqua	Eldorado
11:30 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Ajax Design and Architecture Glenn Vanderburg	Implementing SOA Neal Ford	Groovy: Greasing the Wheels of Java Scott Davis	Annotation Hammer Venkat Subramaniam	Gradual Agile: The Secret to Introducing Agile Practices Jared Richardson
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	JavaScript Exposed: There's a Real Programming Language in There! (Part 1) Glenn Vanderburg	10 Ways to Improve Your Code Neal Ford	Groovy and Java: The Integration Story Scott Davis	Domain Driven Design Venkat Subramaniam	Agile Software Testing Strategies Jared Richardson
4:45 - 5:00 PM	break				
5:00 - 6:30 PM	JavaScript Exposed: There's a Real Programming Language in There! (Part 2) Glenn Vanderburg	Real World Grails Scott Davis	Shippers Unite! Jared Richardson	OSGi: A Well Kept Secret Venkat Subramaniam	Power Regular Expressions in Java Neal Ford
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Neal Ford				

Sat, May. 05, 2007					
	Flatiron Ballroom	Flagstaff	Red Rocks	Chautauqua	Eldorado
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary	Java Performance Myths Glenn Vanderburg	Debugging and Testing the Web Tier Neal Ford	Build Teams, Not Products Jared Richardson	Drizzling with Groovy and Rules Venkat Subramaniam
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary	Everything Old Is New Again Glenn Vanderburg	Continuous Integration with Cruise Control Jared Richardson	Agile Release Estimating, Planning and Tracking Pete Behrens	Advanced Selenium Neal Ford
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Spring 2.0: New and Noteworthy Ben Hale	The Enterprise Service Bus: Do We Really Need It? Mark Richards	The Google Web Toolkit, Part One David Geary	Agile Metrics and Measurements Pete Behrens	Introduction to JRuby Neal Ford
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Spring and Hibernate in the Middle Tier Ben Hale	The Zen of REST Scott Davis	The Google Web Toolkit, Part Two David Geary	Making Architecture Work Through Agility Mark Richards	Rails for JRuby Neal Ford
4:45 - 5:30 PM	BIRDS OF A FEATHER				

Sun, May. 06, 2007					
	Flatiron Ballroom	Flagstaff	Red Rocks	Chautauqua	Eldorado
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Java 6 Features, what's in it for you? Venkat Subramaniam	AOP and JMX: A match made in heaven Ben Hale	The Busy Developer's Guide to Rules and Rules Engines Using JESS Ted Neward	Intro to Java Persistence API (JPA) Mark Richards	The Grails Plug-in System: Plug into productivity Graeme Rocher
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Writing Good APIs Brian Pontarelli	Grails: Spring & Hibernate Development Re-invented Graeme Rocher	The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward	Advanced Java Persistence API (JPA) Mark Richards	get Fit Venkat Subramaniam
12:30 - 1:15 PM	LUNCH				
1:15 - 2:00 PM	EXPERT PANEL DISCUSSION				
2:00 - 3:30 PM	EJB3 Core Specification (JSR-220) Mark Richards	Acegi Security: The security framework with the funny name Ben Hale	The Busy Java Developer's Guide to Reflection Ted Neward	GORM - Object Relational Mapping with Hibernate De-mystified Graeme Rocher	Productive Programmer: Automation and Canonicity Neal Ford
3:30 - 3:45 PM	BREAK				
3:45 - 5:15 PM	Making The Right Persistence Framework Choice Mark Richards	Spring Web Flow Jumpstart Ben Hale	The Busy Java Developer's Guide to ClassLoaders Ted Neward	Mocking Web Services Scott Davis	Productive Programmer: Acceleration, Focus, and Indirection Neal Ford

Rocky Mountain Software Symposium

Renaissance Suites Flatiron Hotel

May 04 - 06, 2007

<http://www.nofluffjuststuff.com/sh/2007-05-denver>

Spring 2.0: New and Noteworthy by Ben Hale

Spring 2.0 has marked a major advance in the Spring Framework. While still maintaining backwards compatibility, this release adds quite a few new features. What are those features and how do they add value? Come by and see.

Spring and Hibernate in the Middle Tier by Ben Hale

To today's JEE developer, there are two indispensable tools for creating applications; Spring and Hibernate. Together these two frameworks comprise one of the most powerful and often used stacks in the industry. While it is possible to do amazing things it's not always obvious how best to use them to maximize value. This session aims to correct that.

AOP and JMX: A match made in heaven by Ben Hale

You're winding down project and you get that dreaded email from your project manager, "How hard would it be to add some performance monitoring to the system?" Well, after this session, you'll be able to respond, "No problem at all!" It turns out that with a pinch of AOP and a dash JMX, you can introduce amazing management and monitoring capabilities without changing you're mainline code one bit.

Acegi Security: The security framework with the funny name by Ben Hale

Security is one of the major requirements in modern day enterprise applications and yet it is also one of the weakest parts of most developers toolboxes. The problem is of course that security is HARD! It turns out that rather than reinventing the wheel for each application, developers can turn to a great security framework out there already; Acegi.

Spring Web Flow Jumpstart by Ben Hale

Have you ever developed a web application with a long user action based on form input? Did you curse the Java community for their inability to address this very common application type? Well, attend this session about Spring Web Flow and you'll curse no more.

Writing Good APIs by Brian Pontarelli

Writing APIs is fairly easy but writing an API that is usable and lives longer than a few days is hard. This talk discusses methodologies, tips and tricks for writing good APIs.

RAD JSF with Seam, Facelets, and Ajax4jsf, Part One by David Geary

In this session, see how you can get Ruby On Rails-like productivity on the Java side of the house with this compelling combination of technologies.

RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two by David Geary

A continuation of a 2-session presentation on Seam, Facelets, and Ajax4jsf.

The Google Web Toolkit, Part One by David Geary

Developing highly interactive web applications, for the most part requires knowledge of a wide array of technologies: HTML, CSS, JavaScript, XMLHttpRequest, JSP, JSF, etc. With the Google Web Toolkit (GWT), Google turns that notion of development on its head. Instead, you implement Ajax applications by writing almost entirely in Java. You use an AWT-like API, which the Google compiler compiles to JavaScript that runs on the client.

The Google Web Toolkit, Part Two by David Geary

The second part of a 2-session presentation on the Google Web Toolkit.

Ajax Design and Architecture by Glenn Vanderburg

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

JavaScript Exposed: There's a Real Programming Language in There! (Part 1) by Glenn Vanderburg

With the sudden importance of Ajax, it's time to take JavaScript seriously. That means learning it the right way: looking at the fundamentals of the language and surveying its strengths and weaknesses, instead of

just copying other people's poorly written examples.

JavaScript Exposed: There's a Real Programming Language in There! (Part 2) by Glenn Vanderburg

Building on part 1, this talk dives deep into JavaScript's object model. We'll see how it differs from more mainstream object-oriented languages, and why. We'll explore how to hide some of those differences, as well as the reasons you might not want to. Additionally, we'll cover useful tools for JavaScript testing, debugging, and profiling.

Java Performance Myths by Glenn Vanderburg

Performance myths about the Java platform abound, from the general "Java is slow", to the more specific "reflection is slow", "allocation is slow", "synchronization is slow", "garbage collection is slow", etc. Many of these myths have their root in fact (in JDK 1.0, everything was slow); today, not only are many of these statements not true, but Java performance has surpassed that of C in many areas, such as memory management.

Everything Old Is New Again by Glenn Vanderburg

The early years of computers -- the '50s and '60s -- were characterized by furious exploration of a huge variety of different ideas. Since then many of the hot topics of those days have moved to the fringe, largely ignored by the mainstream of software development. But some of them are being rediscovered, and a lot of what we think of as "new developments" are really just some old ideas returning to center stage.

The Grails Plug-in System: Plug into productivity by Graeme Rocher

Grails is more than just a web framework, it is a complete platform and API for runtime configuration. This talk, by Grails project lead Graeme Rocher, will demonstrate Grails' modular architecture and how to hook into runtime configuration to adapt your application based on its environment and/or the presence of other plug-ins.

Grails: Spring & Hibernate Development Re-invented by Graeme Rocher

Spring & Hibernate development, although hailed in the past as being "lightweight", is still an XML-centric, configuration heavy approach to web application development. In this talk, by Grails project lead Graeme Rocher, you will discover how you can leverage Spring & Hibernate without ever having to write a line of configuration!

GORM - Object Relational Mapping with Hibernate De-mystified by Graeme Rocher

Hibernate is an immensely powerful ORM layer with an array of features and mapping options which comes at the cost of complexity. The web application framework Grails ships with an ORM layer that builds on top of Hibernate, but eliminates much of the complexity through clever use of convention and the dynamic features of the Groovy language. Known as GORM, it offers a convention-based mapping strategy that hooks into the Hibernate configuration model and provides an abstraction layer above Hibernate with powerful features like Groovy builders for Criteria, dynamic finders and transaction management through closures.

Gradual Agile: The Secret to Introducing Agile Practices by Jared Richardson

Agile practices are popular because they work, but getting people to take that first step can be tricky.

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.

Shippers Unite! by Jared Richardson

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

Build Teams, Not Products by Jared Richardson

A great team builds great software, but how do you build a great team?

Continuous Integration with Cruise Control by Jared Richardson

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

The Enterprise Service Bus: Do We Really Need It? by Mark Richards

There has been a significant amount of buzz in the community and industry about the definition and role of

an Enterprise Service Bus (ESB), particularly within the area of Service-Oriented Architecture (SOA). In this product-agnostic high energy session we will take a step back and consider whether we really need an ESB. Through real-world application and architecture scenarios we will see where an ESB would be helpful and where it would be overkill. Then, using product-agnostic coding examples we will learn what an Enterprise Service Bus is supposed to do, then answer the question about whether the ESB is just a bunch of hype or if we really need it.

Making Architecture Work Through Agility by Mark Richards

As companies continue to change the way they do business, so must the IT systems that support the business. Changes due to regulatory requirements, competitive advantage, mergers, acquisitions, and industry trends require flexible IT systems to meet the demands of the business. Software Architects must therefore make their architectures more agile to meet the flexible demands of today's business. Through real-world examples and scenarios we will explore some of the challenges facing Software Architecture and discuss several concrete techniques for applying agility to both the architecture process and the technical architecture itself. We will also look at various architecture refactoring techniques, and discuss the pros and cons of each. By attending this session you will learn how to apply various agile techniques to improve your architectures and overcome some of the challenges facing software architecture in today's ever-changing market.

Intro to Java Persistence API (JPA) by Mark Richards

In addition to providing a simplified API, the new EJB3 specification (JSR-220) defines a standard ORM Java Persistence API (JPA) that is rapidly gaining in popularity. As you will see in this session, JPA bears a striking resemblance to popular ORM solutions like Hibernate and Toplink. In this session we will explore in detail the new Java Persistence API offered by JSR-220. We will start by discussing the overall design and architecture of the JPA and how the major components within JPA interact. We will then look at defining mapping objects (entities) and how to use the EntityManager to manage these entities. Through interactive coding examples we will investigate the pros and cons of detached entities and merging, how to map and use entity relationships (1-1, 1-N, N-1, and N-N), discuss Lazy Loading, and finally see how to use XML mappings rather than annotations. More advanced features of JPA will be covered in a separate session.

Advanced Java Persistence API (JPA) by Mark Richards

This session picks up where the Intro to JPA session left off and covers some of the more advanced topics in the Java Persistence API. Some of the topics covered in this session include switching persistence providers, versioning, compound keys, entity inheritance, and finally handling both simple and complex stored procedures. Some knowledge of JPA is recommended for this session as I will not be covering the basics of JPA (that is covered in a separate Intro to JPA session). Through a combination of slides and interactive coding I will demonstrate these advanced topics using both Hibernate and Toplink JPA.

EJB3 Core Specification (JSR-220) by Mark Richards

The new EJB 3 specification (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will discuss why EJB is still important, and 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, JTA transaction management, declarative security, and interceptors. For those of you who still like to code in 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) - that topic is covered in separate JPA sessions.

Making The Right Persistence Framework Choice by Mark Richards

Java Persistence has come a long way in the past 4 years. We have many viable options available now, including JPA, Hibernate, iBATIS, Toplink, and OpenJPA. With so many options available now it is difficult to know when to use which framework. In this session we will focus on native Hibernate, JPA, and the iBATIS framework, and discuss the main strengths and weaknesses of each approach and what the decision criteria is for using each of these frameworks. Knowing that it is not a one-size-fits-all situation when it comes to Java Persistence, through interactive coding we will take a look at how to use iBATIS together with native Hibernate or JPA and when this makes sense. By attending this session you will gain the knowledge necessary to make informed decisions about which Java Persistence Framework to choose for your current or next Java-based application.

Implementing SOA by Neal Ford

This talk avoids SOA hype and gets to the meat of the matter: how do you implement a Service-Oriented Architecture, what are the technological pitfalls, how do you test it, and what traps should you avoid. No marketecture: just implementation details.

10 Ways to Improve Your Code by Neal Ford

No one writes perfect code, and every developer eventually falls into a slump where they just crank out the same code day after day. This session illustrates 10 different ways to improve your code, covering sacred cows, good citizens, smells, and more.

Power Regular Expressions in Java by Neal Ford

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

Keynote: Polyglot Programming by Neal Ford

Java is dead; long live Java!

Debugging and Testing the Web Tier by Neal Ford

As our applications have spilled from the server across the wire to the web tier, we increasingly must debug and test in the browser. This session covers debugging and testing tools for clients, JavaScript, and Ajax.

Advanced Selenium by Neal Ford

This session discusses advanced Selenium techniques for testing web applications. It discusses techniques for both TestRunner and Remote Control Selenium, including data driven tests, creating branch points, testing Ajax applications, creating flexible tests, integration with continuous integration, and tons more.

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.

Rails for JRuby by Neal Ford

This session explains all the hype surrounding Ruby on Rails, in a context familiar to Java developers. It covers convention over configuration, ActiveRecord, controllers, views, Ajax, scaffolding, testing, and deployment...on the JVM, using JRuby.

Productive Programmer: Automation and Canonicity by Neal Ford

This session discusses how to use the Productive Programmer principles of automation and canonicity to become a more productive programmer. This session describes these principles, but the primary focus of this session is demonstration of these principles with real-world examples.

Productive Programmer: Acceleration, Focus, and Indirection by Neal Ford

This session discusses how to use the Productive Programmer principles of acceleration, focus, and indirection to become a more productive programmer. This session describes these principles, but the primary focus of this session is demonstration of these principles with real-world examples.

Agile Release Estimating, Planning and Tracking by Pete Behrens

Business leaders and stakeholders require accountability and accuracy in our software release projections and yet, as an industry, we have failed. However, many of these same leaders are not convinced that agile is any more than an excuse to avoid projections at all. While it is true that agility provides the framework to support change, it doesn't mean you can't provide accurate projections. In fact, a well-executed agile process actually provides more accurate results with less time investment than traditional methods. This session will demonstrate these agile project management techniques to manage 6-12 month projects.

Agile Metrics and Measurements by Pete Behrens

Are you being asked to measure your agility? How productive is your team? What is the quality of your product? How accurate are your estimates? Be careful, you will get what you measure. This session will evaluate metrics which attempt to measure productivity, quality, estimation accuracy, value, and return on investment within the context of an agile project. What measurements are teams using and why? We will explore various measurements used by session participants and discuss some of their pros and cons.

Groovy: Greasing the Wheels of Java by Scott Davis

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

"scripting goodness" to the Java platform.

Groovy and Java: The Integration Story by Scott Davis

I'm attracted to Groovy because of its spirit of inclusiveness. Because it extends my platform of choice, not replaces it -- include a single JAR in your classpath and you are Groovy-enabled. Because it offers full bidirectional integration with Java. Because it offers a nearly flat learning curve for experienced Java developers. Come see how you can use Groovy to augment your existing Java codebase.

Real World Grails by Scott Davis

Scott Davis is the Editor in Chief of aboutGroovy.com. The website, in addition to being, umm, about Groovy, is implemented in Grails. This talk shows you how to get started with Grails, but also talks about the experience of using it in a live, production web site.

The Zen of REST by Scott Davis

Google quietly deprecated their SOAP search API at the end of 2006. While this doesn't mean that you should abandon SOAP, it does reflect a growing trend towards simpler dialects of web services. Google joins a number of popular websites (Yahoo, Flickr, YouTube, del.icio.us) that offer all of the benefits of web services without all of the complexity of SOAP.

Mocking Web Services by Scott Davis

In this talk, we'll survey the web services exposed by leading websites (Google, Yahoo, Amazon, eBay) and discuss how they can be easily mocked up for testing purposes and to aid offline development. You'll see working examples of RESTful, SOAP, and JSON web services, as well as strategies for unit and functional testing your asynchronous, service-oriented architecture.

The Busy Developer's Guide to Rules and Rules Engines Using JESS by Ted Neward

If you've been keeping your ear to the ground, you may have heard some talk recently about "rules", "business rules" and "rules engines", but not necessarily any clear discussion on what they are, how to use or design them, or why they might be useful or important.

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

If you've never used Reflection (`java.lang.reflect`), you don't know what you're missing. In this presentation, we'll take a code-first, soup-to-nuts look at the Java Reflection APIs, from how to examine the class metadata that Reflection provides, to using annotations to enhance that metadata with your own information, even through the use of Java Dynamic Proxies to create flexible object "interceptors" that can layer services in front of ordinary method calls with nothing more complicated than an interface and a factory.

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

If you've ever gotten a `ClassCastException` and just knew the runtime was wrong about it, or found yourself copying `.jar` files all over your production server just to get your code to run, then you probably find the Java `ClassLoader` mechanism to be deep, dark, mysterious, and incomprehensible. Take a deep breath, and relax--`ClassLoaders` aren't as bad as they seem at first, once you understand a few basic rules regarding their operation, and have a bit more tools in your belt to diagnose `ClassLoader` problems. And once you've got that, and hear about `ClassLoaders`' ability to run multiple versions of the same code at the same time, and to provide isolation barriers inside your application, or even compile code on the fly from source form, you might just find that you like `ClassLoaders` after all... maybe.

Annotation Hammer by Venkat Subramaniam

Annotation is an interesting feature in Java. However, like any features, there are good uses and bad uses. When should you use Annotation? This presentation will answer that question for you.

Domain Driven Design by Venkat Subramaniam

Domain Driven Design (DDD) is an approach that places emphasis on the domain model and carrying it into implementation. DDD is mostly repackaging of fundamental OO Design. It brings new emphasis to what we should be already doing, but often find it hard and confusing given the realities and complexities of our real world. In this presentation we will take a close look at what DDD is and how to use it for agile development.

We will discuss several design options, and also look at some examples of good modeling and layering.

OSGi: A Well Kept Secret by Venkat Subramaniam

In this presentation we will introduce OSGi and discuss how it can help modularize and version your enterprise Java applications.

Drooling with Groovy and Rules by Venkat Subramaniam

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

Java 6 Features, what's in it for you? by Venkat Subramaniam

What benefit do new Java 6 features offer you. Are there issues with using these features. The objective of this presentation is not simply to introduce you to the features, but to the effective use of these as well.

get Fit by Venkat Subramaniam

Unit testing tells you, the programmer, that your code (and the change) meets your expectations. How do you know if you are meeting your customers' expectations? Agile development is all about feedback and doing what's relevant to the customers, isn't it? Framework for Integration testing or Fit helps you to automate tests for customer expectations.