

# Greater Atlanta Software Symposium

Atlanta Marriott Perimeter Center

October 05 - 07, 2007

<http://www.nofluffjuststuff.com/sh/2007-10-atlanta>

(event schedule as of October 4, 2007)

Fri, Oct. 05, 2007					
	Salon A/B	Salon C/D	Salon E	Monroe/Jackson	Jefferson/Washington
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Implementing SOA Neal Ford	Java Annotations: From Definition to Consumption Ted Neward	Shippers Unite! Jared Richardson	Architecting JavaServer Faces Applications Kito Mann	Groovy: Greasing the Wheels of Java Scott Davis
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	10 Ways to Improve Your Code Neal Ford	Enterprise Performance and Scalability Ted Neward	Agile Software Testing Strategies Jared Richardson	AJAX and JSF: Natural Synergy Kito Mann	Groovy and Java: The Integration Story Scott Davis
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Power Regular Expressions in Java Neal Ford	Java6: Exploring Mustang Ted Neward	Build Teams, Not Products Jared Richardson	Introduction to JBoss Seam Kito Mann	The Secrets of GORM Scott Davis
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Scott Davis				

Sat, Oct. 06, 2007					
	Salon A/B	Salon C/D	Salon E	Monroe/Jackson	Jefferson/Washington
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	JavaScript for Ajax Programmers Stuart Halloway	Domain Driven Design Venkat Subramaniam	Building DSLs in Static and Dynamic Languages Neal Ford	The Zen of REST Scott Davis	Distributed Teams: Remote Agility Jared Richardson
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Prototype: Ajax and JavaScript++ Stuart Halloway	Spring 2.0: New and Noteworthy Mark Fisher	OSGi: A Well Kept Secret Venkat Subramaniam	Mocking Web Services Scott Davis	Software Development Techniques Jared Richardson
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Getting Started with Grails Jason Rudolph	Structuring concurrent applications in JDK 5.0 Brian Goetz	get Fit Venkat Subramaniam	Building Enterprise Applications with JavaServer Faces and Spring Kito Mann	This Week In Refactoring Stuart Halloway
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Going Further with Grails Jason Rudolph	Effective Concurrent Java Brian Goetz	FP for Java Programmers Venkat Subramaniam	The Role of Spring in an ESB Mark Fisher	Screwing Up Agile Stuart Halloway
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS				

Sun, Oct. 07, 2007					
	Salon A/B	Salon C/D	Salon E	Monroe/Jackson	Jefferson/Washington
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Beginning Object-Relational Mapping with Hibernate Brian Sam-Bodden	The Busy Java Developer's Guide to Debugging and Monitoring Ted Neward	Introduction to Spring Security Mark Fisher	Advanced Domain Models in Grails: Enterprise Integration Made Easy Jason Rudolph	Productive Programmer: Acceleration, Focus, and Indirection Neal Ford
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	10 ways to use Hibernate effectively Brian Sam-Bodden	The Java Memory Model Brian Goetz	Message Driven POJOs Mark Fisher	Why Test Driven Development? Muness Alrubaie	Productive Programmer: Automation and Canonicity Neal Ford
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden	Beyond ACID: transactions management, in theory and practice Brian Goetz	Introduction to JRuby Neal Ford	Creating Manageable Systems with JMX Vladimir Vivien	JBoss ESB Deep Dive Burr Sutter
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	Java Performance Myths Brian Goetz	Rails for JRuby Neal Ford	Unit Testing - Java, Groovy and JRuby Muness Alrubaie	Open Source SOA Burr Sutter

# Greater Atlanta Software Symposium

Atlanta Marriott Perimeter Center

October 05 - 07, 2007

<http://www.nofluffjuststuff.com/sh/2007-10-atlanta>  
(event schedule as of October 4, 2007)

## **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.

## **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.

## **Beyond ACID: transactions management, in theory and practice by Brian Goetz**

Transactions are the software building blocks of enterprise applications, but not all transactional systems are created equally. This talk covers the basics of what transactions are, why they are essential to building reliable enterprise software, the fundamental properties of transactions, and how transactions are supported and implemented in popular frameworks such as Java EE and Spring.

## **Java Performance Myths by Brian Goetz**

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.

## **Beginning Object-Relational Mapping with Hibernate by Brian Sam-Bodden**

Hibernate is an open source Object-Relational Mapping Framework that mostly automates the tedious and time-consuming task of persisting Java objects to a relational database. Hibernate is quickly becoming the preferred way for enterprise developers to overcome the object-relational impedance mismatch and a good alternative to the coarse-grained Entity EJBs, low-level raw JDBC, and by-committee specifications like JDO. Learn what your choices in the ORM arena, what to look for in an ORM tool, and how to get started with Hibernate for your next J2SE or J2EE project.

## **10 ways to use Hibernate effectively by Brian Sam-Bodden**

Learn 10 tried and true ways to improve the way you use Hibernate today. In this session you would learn about a collection of 10 tips, tricks, practices and tools that will make you more effective at designing, implementing, testing and tuning your application's Hibernate-powered object-relational layer.

## **Professional Java UI development with the Eclipse RPC by Brian Sam-Bodden**

Learn how to build featured rich applications using the Eclipse Rich Client Platform. The Eclipse platform is an open tools platform, on top of this platform you can build your own applications (which do not need to be IDE like or IDE related). Yet you can enjoy the benefits of working with a mature and featured rich platform that can greatly reduce the amount of time required to create a professional-looking and robust Java UI application.

## **Beginning Drools - Rule Engines in Java by Brian Sam-Bodden**

Software development is expensive, when business rules are hard-coded in your application's source code, changes and additions to those rules translate to wasted time and money. Good object-oriented, component-based approaches can alleviate the burden of keeping up with changes in the business world but they still require that expert knowledge of the changes be passed from the decision makers to the

business analysts and finally to programmers that need to implement these changes. Business Rule Engines and Business Rule Languages are based on the basic premise of separation of concerns by empowering business domain experts to express the rules of business in a way that it is directly usable by applications.

### **JBoss ESB Deep Dive by Burr Sutter**

This session will be a deep dive into the capabilities of the open source JBoss Enterprise Service Bus 4.2 GA. An ESB is primarily categorized by its capabilities in the areas of protocol mediation/abstraction, transformation, orchestration, routing, endpoint registry, etc. Numerous live demos of ESB functionality.

### **Open Source SOA by Burr Sutter**

At some point, code will be written, software tools will be acquired and systems will be built. Unfortunately the Java development world is a confused mess as it relates to a method of building a Service Oriented Architecture (SOA)-based anything. Our objective is to answer the following questions: Should I use Web Services everywhere? Is an Enterprise Services Bus (ESB) useful and required? Should I be programming in the XML-based syntax of BPEL instead of Java? Do I need JBI and/or SCA? What Open Source implementations are available to solve SOA related challenges?

### **Shippers Unite! by Jared Richardson**

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

### **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.

### **Build Teams, Not Products by Jared Richardson**

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

### **Distributed Teams: Remote Agility by Jared Richardson**

How do you keep a team scattered across time zones in sync?

### **Software Development Techniques by Jared Richardson**

Throughout our software careers we learn habits from our coworkers, from books we've read, and occasionally, from conferences we attend. Much of our competence comes from the tips and tricks we pick up as we go.

### **Getting Started with Grails by Jason Rudolph**

Grails is an open-source web application framework that's all about getting things done. Grails combines best-of-breed Java technologies (including Hibernate and Spring), convention over configuration, and the powerful and dynamic Groovy language. Together with these elements and Groovy's ability to seamlessly integrate with your existing Java code, Grails finally legitimizes rapid web application development for the Java platform.

### **Going Further with Grails by Jason Rudolph**

Grails provides a tremendous jump-start to any web application. With easy scaffolding and zero-configuration ORM, you're up and running in no time. But what's needed to effectively move from a vanilla Grails project to a fully-customized application fit for public consumption?

### **Advanced Domain Models in Grails: Enterprise Integration Made Easy by Jason Rudolph**

Have you seen someone develop a Rails or Grails application in a matter of minutes, only to later discover that their domain model and database schema followed conventions that are different from your existing systems? Or perhaps you're interested in using Grails, but you don't want to duplicate your existing Java domain classes in Groovy. In this session, we'll see how Grails makes it easy to hook into your pre-defined schemas or existing entity classes, while still getting all the rapid application development (RAD) goodness that Grails has to offer.

### **Architecting JavaServer Faces Applications by Kito Mann**

Over the past few years, a lot of time has been spent explaining what JSF is, and how different pieces of it work. However, little attention has been given to the process of architecting applications. This makes JSF architecture seem like a black art, since there are so many possible approaches to the application's architecture.

### **AJAX and JSF: Natural Synergy by Kito Mann**

With the emergence of AJAX as a preferred way of building web user interfaces, JavaServer Faces (JSF) has proved itself to be a natural fit for integrating AJAX with Java sever-side logic.

### **Introduction to JBoss Seam by Kito Mann**

JBoss Seam is a popular open-source application framework for Java Platform, Enterprise Edition (Java EE) 5.0. For web application developers, a significant benefit of Seam is that it greatly enhances JavaServer Faces technology. This session explains key Seam features such as tight integration with EJB3, Hibernate and JPA integration, conversations, RESTful web pages, and so on.

### **Building Enterprise Applications with JavaServer Faces and Spring by Kito Mann**

For developers who are currently using Spring and JavaServer Faces together, this session explains how to handle common application development concerns such as conversational scope, transaction management, and application partitioning.

### **Spring 2.0: New and Noteworthy by Mark Fisher**

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. In this session we'll provide a practical tour of what's new in Spring 2.0. Spring 1.x users who are looking to upgrade to Spring 2.0 will love this session. If you're not using Spring already, this talk will give a great overview of the things you're missing out by not using Spring 2.0.

### **The Role of Spring in an ESB by Mark Fisher**

An Enterprise Service Bus (ESB) brings flow-related functionality such as message routing and transformation to a Service-Oriented Architecture. An ESB also provides a layer of abstraction with endpoints for various protocols and transports. These features promote decoupling of integration logic from business functions, flexibility in the transport layer, and pluggability of POJO services.

### **Introduction to Spring Security by Mark Fisher**

Spring Security (a.k.a. Acegi) enables self-contained, consistent solutions for securing your applications. The interceptor-based approach is non-invasive even when extended to accommodate domain-specific requirements. The two main security processes (authentication and authorization) are decoupled in order to provide flexibility across a wide variety of providers and strategies.

### **Message Driven POJOs by Mark Fisher**

One of the most exciting new features of Spring 2.0 is its support for Message-Driven POJOs. It is now possible to receive JMS messages asynchronously and delegate the handling of those messages to simple objects. If your POJO has a return value, it will automatically be sent to a reply destination. Spring's messaging containers support configurable pooling of concurrent consumers and offer full integration with Spring's transaction management.

### **Why Test Driven Development? by Munees Alrubaie**

Test Driven Development is a central tenet of Agile software development methodologies and is a powerful design technique for any software developer. The reasons for this are many: it encourages more modular, simpler code. It helps you avoid YAGNI. It can be used to document a software system, by providing a specification and examples of its use. A test suite, is also a necessary safety net for any serious refactoring work.

### **Unit Testing - Java, Groovy and JRuby by Munees Alrubaie**

TDD has been around for a long time. And the tools available to write and use your unit tests continue to improve. Some of the tools we use on a regular basis include JUnit, the most popular testing frameworks, Cobertura, a test coverage analysis tool and jMock, a Mock Objects library that allows you to test the

interactions between your objects. Less common, but quickly becoming popular is the practice of writing tests using dynamic languages such as Groovy and Ruby.

### **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.

### **Building DSLs in Static and Dynamic Languages by Neal Ford**

This session discusses building Domain Specific Languages and DSL-style code in Java, Groovy, and Ruby. It discusses the different types of DSLs, details on how to implement them in Java, Groovy, and Ruby, and example problem domains where DSLs make sense.

### **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.

### **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.

### **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.

### **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.

### **The Secrets of GORM by Scott Davis**

GORM (the Grails Object/Relational Mapper) is one of the many high points of the Grails web framework. GORM is a thin Groovy wrapper over Hibernate, but that doesn't begin to capture excitement of what GORM

brings to the party. Imagine being able to call `book.save()` and `book.delete()` on your `Book` class; calling `Book.get(1)` to retrieve your book from the database by primary key; using `Book.list()` to pull an `ArrayList` of `Book` objects into your application. Now imagine getting all of that functionality (and more) for free with each new class you define. No interfaces to implement. No abstract classes to extend. Persistence that is transparent, automatic, and simple to use: GORM.

### **KEYNOTE: No, I Won't Tell You Which Web Framework to Use: or The Truth (with Jokes) by Scott Davis**

"Which framework should I use?" is the question most often heard on the No Fluff, Just Stuff tour. It's well worth asking. Unfortunately, there is no simple answer. After years on the tour, most speakers have crafted a response that would make any Washington politician proud -- long on style, but essentially, "Well, it depends..."

### **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.

### **JavaScript for Ajax Programmers by Stuart Halloway**

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

### **Prototype: Ajax and JavaScript++ by Stuart Halloway**

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

### **This Week In Refactoring by Stuart Halloway**

Contributing to open source is great for your career. In a few short hours, you can learn, teach, promote your skills, and improve the quality of the community. In this talk, we will show you how, by doing it.

### **Screwing Up Agile by Stuart Halloway**

Agile software techniques like Scrum and XP are increasingly popular; there are tons of resources on the web to help you do agile right. Tragically, there are very few resources to help you screw up an agile project. This talk will show you how to ruin agile projects.

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

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

### **Enterprise Performance and Scalability by Ted Neward**

Wondering why your enterprise Java app just? sucks? Trying to figure out why you can't get more than 10 concurrent users online at the same time? Looking for ways to try and spot the slowdowns and ways to fix them?

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

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

### **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.

### **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.

### **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.

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

Most interest around Functional Programming (FP) has been academic until recently. Recent commercial languages are beginning to exploit FP features. Knowing more about FP will not only help us make better use of these features, but to exploit those. In this session we will take a close look at FP.

### **Creating Manageable Systems with JMX by Vladimir Vivien**

Starting with the Java 5 programming language, it has gotten easier to incorporate monitoring and manageability into any application running on a standard VM. Developers now have access to a wealth of runtime VM profile information exposed through Java Management Extensions (JMX), including memory consumption, garbage collection, and thread activities (with even more capabilities in the Java 6 programming language). Using JMX, developers are also able to expose runtime control and management information for their own applications.