

Central Florida Software Symposium

Four Points by Sheraton Studio City Hotel

August 22 - 24, 2008

<http://www.nofluffjuststuff.com/conference/orlando/2008/08/index.html>
(event schedule as of August 21, 2008)

Fri, Aug. 22, 2008				
	Hollywood 1	Hollywood 2	Silver Screen 1	Silver Screen 2
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	Evolutionary SOA Neal Ford	Give it a REST Brian Sletten	Know your Java? Venkat Subramaniam
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	The Productive Programmer: Mechanics Neal Ford	RESTlet for the Weary Brian Sletten	Towards an Evolutionary Design Venkat Subramaniam
4:45 - 5:00 PM	BREAK			
5:00 - 6:30 PM	Grails for Struts Developers: A Groovy Alternative Scott Davis	Test Driven Design Neal Ford	Resource-Oriented Computing w/ NetKernel : Software for the 21st Century Brian Sletten	The Java Memory Model Brian Goetz
6:30 - 7:15 PM	DINNER			
7:15 - 8:00 PM	Keynote: by Neal Ford			

Sat, Aug. 23, 2008				
	Hollywood 1	Hollywood 2	Silver Screen 1	Silver Screen 2
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Code Metrics & Analysis for Agile Projects Neal Ford	Structuring concurrent applications in JDK 5.0 Brian Goetz	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	JVM Language Shootout: Clojure, JRuby, and Scala Stuart Halloway
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	Architecture and Scaling Ken Sipe	Effective Concurrent Java Brian Goetz	Viva La Javolution! Brian Sletten	Refactoring JavaScript Stuart Halloway
12:30 - 1:30 PM	LUNCH			
1:30 - 3:00 PM	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe	Java Performance Myths Brian Goetz	JavaServer Faces: A Whirlwind Tour David Geary	How to Fail with 100% Code Coverage Stuart Halloway
3:00 - 3:15 PM	BREAK			
3:15 - 4:45 PM	Spring 2.5 - Spring without XML Ken Sipe	MOPping Up Groovy Venkat Subramaniam	Facelets Explained David Geary	Software Project Estimation Mark Johnson
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS			

Sun, Aug. 24, 2008				
	Hollywood 1	Hollywood 2	Silver Screen 1	Silver Screen 2
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	SOAs Challenges Ken Sipe	FP on JVM Venkat Subramaniam	New Features in Spring Web Keith Donald	Using Ajax4jsf David Geary
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	Getting Started with BPEL Mark Johnson	Acceptance Testing Application Behavior Venkat Subramaniam	Spring Web Flow 2 Deep Dive Keith Donald	Seam David Geary
12:30 - 1:15 PM	LUNCH			
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION			
2:15 - 3:45 PM	Maintaining Project Integrity with JDepend, Macker, PMD, Maven, and other open source tools David Bock	Caring about your Code Quality Venkat Subramaniam	Decorating Web Pages with Ajax using Spring JavaScript Keith Donald	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary
3:45 - 4:00 PM	BREAK			
4:00 - 5:30 PM	Intermediate Maven David Bock	GIS for Web Developers: Adding Where to Your Application Scott Davis	Promoted to Technical Lead. Now what do I do? Mark Johnson	The Google Web Toolkit, Part Two David Geary

Central Florida Software Symposium

Four Points by Sheraton Studio City Hotel

August 22 - 24, 2008

<http://www.nofluffjuststuff.com/conference/orlando/2008/08/index.html>
(event schedule as of August 21, 2008)

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.

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.

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.

Give it a REST by Brian Sletten

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

RESTlet for the Weary by Brian Sletten

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

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.

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

We write very complicated software, don't we? In our systems, we detect when simple things happen. Customers log in, people buy things, a stock is sold at a particular price, inventory shifts locations... all of these events mean little things, but what about the larger picture? Complex events are particular patterns of simpler events that suggest something deeper is happening. Do you know how you'd discover these bigger

picture occurrences? Come hear how the Esper open source software represents a new class of complex event processing (CEP) frameworks that can be added to even high volume, high transaction systems.

Viva La Javolution! by Brian Sletten

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

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

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

Intermediate Maven by David Bock

Maven is a build tool that does a lot, demos well, and leaves the build maintainers managing what seems like unbridled complexity. It doesn't have to be that way - Maven is driven by some strong 'build process methodology', and that complexity can become manageable by wrapping your head around it. Furthermore, you can migrate to Maven 'piecemeal', by mapping your existing ant build to the Maven Lifecycle and calling your existing Ant tasks - you can decide to sip the Maven kool-aid. Ideally, a build tool should be so simple and approachable that it fades into the project background and allows anyone to maintain it. Unfortunately, Maven's power comes at the expense of this ideal - Maven's philosophy is more like "the build process is so important that the people maintaining it should be steeped in the ways of Maven". This talk will give you the exposure you need without elevating The Maven Way to a religion.

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 Explained by David Geary

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet.

Using Ajax4jsf by David Geary

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

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.

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.

The Google Web Toolkit, Part Two by David Geary

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

New Features in Spring Web by Keith Donald

Spring has a number of interesting modules for web application development, including Spring Web MVC, Spring Web Flow, Spring JavaScript, and Spring Faces. This session will provide an overview of these modules and show how they relate to one another. By the end of this session, you'll understand how Spring simplifies the development and deployment of rich web applications. You'll also gain a glimpse into the roadmap for Spring Web 3.0.

Spring Web Flow 2 Deep Dive by Keith Donald

Web Flow is a Spring Web MVC extension that allows you to define Controllers using a higher-order domain-specific-language. This language is designed to model user interactions that require several requests into the server to complete, or may be invoked from different contexts. This session dives deep into the features of the Web Flow definition language, and illustrates how to use it to create sophisticated controller modules.

Decorating Web Pages with Ajax using Spring JavaScript by Keith Donald

Spring JavaScript is a JavaScript abstraction framework that allows you to progressively enhance a web page with behavior. The framework consists of a public JavaScript API along with an implementation that builds on the Dojo Toolkit. Spring.js aims to simplify the use of Dojo for common enterprise scenarios while retaining its full-power for advanced use cases. Come to this session to learn to use Spring.js and Dojo to create compelling user interfaces.

Architecture and Scaling by Ken Sipe

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

Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence by Ken Sipe

Well the standards created EntityBeans.... yea. and the community created Hibernate. Fortunately the standards body learned some lessons and created JPA. JPA requires a vendor implementation and none make a better choice than Hibernate. Combined with Spring this trio is a powerhouse when it comes to developer productivity on applications requiring persistence.

Spring 2.5 - Spring without XML by Ken Sipe

Spring 2.5 is brand spanking new, with a number of fantastic features. With growth of large and complex Spring applications which struggle with xml manageability and with the added pressure of Guice and SEAM there is a push for less XML, with solution leaning towards annotations. Spring 2.5 adds to the toolset provided in Spring 2.0 to provide a development environment where XML is greatly reduced... or eliminated if you so choose.

SOAs Challenges by Ken Sipe

SOA... Is it hype? What's real... and what's not? What is the right abstraction level?

Software Project Estimation by Mark Johnson

As developers we dread when management requests a project estimate. Typically, you do not have the opportunity to understand all the requirements, the team composition is unknown, and you have been given

until tomorrow end of day to produce an estimate. Several months later everyone is yelling at you about the software estimation errors encountered during the project.

Getting Started with BPEL by Mark Johnson

With all of these web services becoming available there is an increasing need for tools to pull together multiple web services into one composite service. BPEL is an up and coming approach to orchestrating a workflow consisting of Web Service calls.

Promoted to Technical Lead. Now what do I do? by Mark Johnson

You have just received the much desired promotion to Technical Team Lead The team is waiting your direction. You What should you do now?

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.

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!

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: 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?

Code Metrics & Analysis for Agile Projects by Neal Ford

What does code + methodology have to do with one another? Everything! Agile projects focus on delivering working code, and tools exist to allow you to verify some quality metrics for your code. This session is a survey of tools and metrics that allow you to determine the quality of your code and strategies to "wire it" into your agile project.

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 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.

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

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

JVM Language Shootout: Clojure, JRuby, and Scala by Stuart Halloway

In this talk, we will explore and compare three of the most interesting new JVM languages: Clojure, JRuby, and Scala. Each of these languages aims to greatly simplify writing code for the JVM, and all three 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.

Refactoring JavaScript by Stuart Halloway

The rise of Ajax and Rich Web Applications, plus the success of dynamic languages, has caused people to revisit the JavaScript language. Now that we take JavaScript seriously as a language, it is time to get serious about the quality of JavaScript code, through refactoring. In this talk, we will approach refactoring JavaScript in three phases: Test first, then refactor. Bring JavaScript code under test, so that you can refactor with confidence. Refactoring 101. Explore some important refactorings: composed method, extract method, introduce named parameter, and extract object Common problems. Work through three problems endemic to legacy JavaScript code: making JavaScript unobtrusive, refactoring to prototype-based inheritance, and refactoring to functional style.

How to Fail with 100% Code Coverage by Stuart Halloway

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

Know your Java? by Venkat Subramaniam

Java has been around for well over a decade now. It started out with the goal of being simple. Over the years, its picked up quite a bit of features and along comes complexity. In this presentation we will take a look at some tricky features of Java, those that can trip you over, and also look at some ways to improve your Java code.

Towards an Evolutionary Design by Venkat Subramaniam

A good design is critical for success with agile development. That does not mean a big up-front design. The design has to be evolutionary. However, the design you evolve must be extensible and maintainable. After all, you can't be agile if your design sucks.

MOPping Up Groovy by Venkat Subramaniam

Groovy's dynamic capabilities makes it an attractive language for Meta Programming. There are several facilities to intercept method calls to do AOP kind of operations. You can also inject methods dynamically.

You can also easily perform method synthesis as well. In this presentation we will take a look at techniques that make Groovy pretty Hip for MOP.

FP on JVM by Venkat Subramaniam

Functional Programming Languages (FPLs) have been around for a long time. A lot of features that we get excited about in dynamic languages are common place in FPLs. FPLs are gaining importance due to various changes in our industry. What's exciting is that you can use them on the JVM. In this presentation we will dig into the details of what makes FPLs so interesting and look at ways to use them on the JVM?in your Java projects.

Acceptance Testing Application Behavior by Venkat Subramaniam

How do you ensure your applications meet the expectations of your key customers? In this session we will explore using the FIT tool and Behavior Driven Design tools to do exactly this.

Caring about your Code Quality by Venkat Subramaniam

We all have seen our share of bad code. We certainly have come across some good code as well. What are the characteristics of good code? How can we identify those? What practices can promote us to write and maintain more of those good quality code. This presentation will focus on this topic that has a major impact on our ability to be agile and succeed.