

Central Florida Software Symposium

Sheraton Studio City Hotel

August 24 - 26, 2007

<http://www.nofluffjuststuff.com/sh/2007-08-orlando>
(event schedule as of August 22, 2007)

Fri, Aug. 24, 2007				
	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	JavaServer Faces: A Whirlwind Tour David Geary	10 Ways to Improve Your Code Neal Ford	Groovy: Greasing the Wheels of Java Scott Davis	Java Annotations: From Definition to Consumption Ted Neward
2:45 - 3:15 PM	BREAK			
3:15 - 4:45 PM	Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico David Geary	Implementing SOA Neal Ford	Groovy and Java: The Integration Story Scott Davis	Java6: Exploring Mustang Ted Neward
4:45 - 5:00 PM	BREAK			
5:00 - 6:30 PM	Ajaxian Faces David Geary	Agile Project Management with Mingle Neal Ford	Real World Grails Scott Davis	Enterprise Performance and Scalability Ted Neward
6:30 - 7:15 PM	DINNER			
7:15 - 8:00 PM	Keynote: by Scott Davis			

Sat, Aug. 25, 2007				
	Hollywood 1	Hollywood 2	Silver Screen 1	Silver Screen 2
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Spring 2.0: New and Noteworthy Mark Fisher	Productive Programmer: Acceleration, Focus, and Indirection Neal Ford	RAD JSF with Seam, Facelets, and Ajax4jsf, Part One David Geary	The Busy Java Developer's Guide to Threads and Synchronization Ted Neward
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	Message Driven POJOs Mark Fisher	Productive Programmer: Automation and Canonicity Neal Ford	RAD JSF with Seam, Facelets, and Ajax4jsf, Part Two David Geary	The Busy Java Developer's Guide to java.util.concurrent Ted Neward
12:30 - 1:30 PM	LUNCH			
1:30 - 3:00 PM	The Google Web Toolkit, Part One David Geary	Building ChangingThePresent: Agility in Action Bruce Tate	Spring MVC Essentials Mark Fisher	The Enterprise Service Bus: Do We Really Need It? Mark Richards
3:00 - 3:15 PM	BREAK			
3:15 - 4:45 PM	The Google Web Toolkit, Part Two David Geary	Effective Teams Bruce Tate	Advanced Grails Scott Davis	EJB3 Core Specification (JSR-220) Mark Richards
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS			

Sun, Aug. 26, 2007				
	Hollywood 1	Hollywood 2	Silver Screen 1	Silver Screen 2
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Making Architecture Work Through Agility Mark Richards	Debugging and Testing the Web Tier Neal Ford	Ruby for Java programmers Bruce Tate	The Secrets of GORM Scott Davis
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	Intro to Java Persistence API (JPA) Mark Richards	Advanced Selenium Neal Ford	Rails for Java Programmers Bruce Tate	Introduction to Spring Security Mark Fisher
12:30 - 1:15 PM	LUNCH			
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION			
2:15 - 3:45 PM	Advanced Java Persistence API (JPA) Mark Richards	Building DSLs in Static and Dynamic Languages Neal Ford	The Zen of REST Scott Davis	The Busy Java Developer's Guide to Java Platform Security Ted Neward
3:45 - 4:00 PM	BREAK			
4:00 - 5:30 PM	Making The Right Persistence Framework Choice Mark Richards	Web2.0 and Java Mike Levin	Mocking Web Services Scott Davis	The Busy Developer's Guide to Rules and Rules Engines Using JESS Ted Neward

Central Florida Software Symposium

Sheraton Studio City Hotel

August 24 - 26, 2007

<http://www.nofluffjuststuff.com/sh/2007-08-orlando>
(event schedule as of August 22, 2007)

Building ChangingThePresent: Agility in Action by Bruce Tate

ChangingThePresent is the increasingly popular charity donations portal that lets you give donation gifts instead of another pair of fuzzy slippers. The site is built and maintained under unusual circumstances. The team is distributed, with no more than two developers in any one place. The team uses agile techniques such as automated testing, heavy customer involvement, and a SCRUM-like release plan to deliver the core features.

Effective Teams by Bruce Tate

Most conferences will try to tell you that the secret to good software development lies with a process, or a technology, or an architecture. Here's a dirty little secret. You can build working software with an outdated two tier architecture, a waterfall process and COBOL. How? By building a great team. These techniques were used to build one of the most unique and complex up and coming Ruby on Rails sites.

Ruby for Java programmers by Bruce Tate

With the explosion of Ruby on Rails and the Java community interest in features like closures and continuations, the Ruby programming language is an excellent one for all developers to know. As the JRuby virtual machine picks up steam, Ruby becomes a must language to understand. The best way to learn Ruby is to see it in action.

Rails for Java Programmers by Bruce Tate

The productivity of Ruby on Rails cannot be denied, but the explosion of Ruby on Rails left many developers, with hard commitments to Java deployment platforms, out in the cold. The continued evolution of JRuby can change that. JRuby is a Ruby implementation on the Java virtual machine. And yes, it runs Rails. In this session, you will learn Rails as it was meant to be learned, within the context of building a live site, from scratch.

JavaServer Faces: A Whirlwind Tour by David Geary

JavaServer Faces (JSF) has arrived. The standard Java-based web application framework based on Struts, JSF really took off in 2006. Embraced by developers, vendors, and open-source projects, JSF has hit its stride. If you haven't come up to speed on JSF basics, this is the place to start.

Killer JavaScript Frameworks: Prototype, Scriptaculous, and Rico by David Geary

An introduction to the popular Prototype JavaScript framework, and two frameworks built on top of Prototype: Scriptaculous and Rico.

Ajaxian Faces by David Geary

JavaServer Faces is a perfect platform for implementing Web 2.0 interfaces with Ajax. This session explores how you can use these two potent technologies--JSF and Ajax--together to create applications that look and behave like desktop applications but run in the browser.

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.

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.

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.

Spring MVC Essentials by Mark Fisher

Spring MVC is a powerful and flexible framework for building web applications. Its interface-based API promotes loose coupling, yet it also offers many convenient base classes for common functionality. Whether you use the base classes or roll your own, you will discover numerous strategies and extension points. This flexibility will be greatly appreciated once you are up to speed, but newcomers often wonder where to begin.

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.

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. We will take a look under the hood and find out just what an ESB is really doing, and take a quick look at JBI (JSR-208) and see the impact it has on the ESB world. 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.

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 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.

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.

Web2.0 and Java by Mike Levin

Community, metadata, ranking and websites that grow over time are attributes of Web2.0. What else? Come to Orlando, breathe some fresh Florida air and feel the sunshine!. Then, stroll over to the Central Florida Software Symposium and find out. Mike Levin will tell you all about Web2.0 and show you some snippets in Java.

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.

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.

Agile Project Management with Mingle by Neal Ford

Mingle is an innovative project management tool with "skinnable religion", virtual card walls, highly customizable workflow, and role-based views. This talk describes its setup, use, and some implementation details of how it was created.

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.

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.

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.

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.

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..."

Advanced Grails by Scott Davis

Many demonstrations of new technology focus on the shiny turnkey features -- "Look at what this thing magically does for you out of the box!" While Grails has many gee-whiz scaffolding features, it is a framework first. A framework should "make easy things easy and hard things possible." (Apologies to the Perl community for co-opting their battle cry.) This talk focuses on the hard things that are possible with Grails, but require just a bit of glue code to implement.

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.

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.

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).

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.

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?

The Busy Java Developer's Guide to Threads and Synchronization by Ted Neward

Java's threading capabilities have been a part of the Java platform since its inception, yet for many Java developers, using Threads still remain a dark and mysterious art, and synchronization beyond the use of the "synchronized" keyword is almost unknown.

The Busy Java Developer's Guide to java.util.concurrent by Ted Neward

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

The Busy Java Developer's Guide to 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 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.