

Desert Southwest Software Symposium 2006

Sheraton Phoenix Sky Harbor Hotel - Phoenix, AZ

July 28 - 30, 2006

(session listing as of 7/27/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the Desert Southwest Software Symposium coming to Phoenix on July 28 - 30, 2006. You will have the opportunity to attend the best technically focused Java / Open Source event anywhere. We make this claim based on the following:

- 1) **Excellent Speakers with unparalleled access**
- 2) **Limited Attendance = 250 Registrants Max**
- 3) **No Vendors, No Sales Pitches, No Marketecture**
- 4) **Unmatched Value - less than 1/3 of the cost of a national conference**
- 5) **Since 2002, we have delivered over 75 conferences throughout North America**

The No Fluff Just Stuff Software Symposium Series caters to individual developers, development teams, project managers, architects and independent consultants. The Desert Southwest Software Symposium will offer 5 concurrent sessions over three days with over 55 sessions to choose from. The following topics will be featured:

- 1) XML / Web Services
- 2) ServerSide Java
- 3) .Net
- 4) Core Java
- 5) Architecture
- 6) Client Side Java

Registration Fees

| Attendees | Before 7/3/2006 | After 7/3/2006 |
|-----------|-----------------|----------------|
| 1-4 | \$725 | \$825 |
| 5-9 | \$650 | \$725 |
| 10-14 | \$625 | \$700 |
| 15-24 | \$600 | \$675 |
| 25+ | \$575 | \$650 |

The Registration Fee includes the following:

- 1) All Access Pass to the three day symposium
- 2) Handouts from all sessions attended w/binder
- 3) CD with all presentational content @ registration
- 4) Great NFJS Swag
- 5) Opportunity to win an iPod everyday during the symposium raffle

Go to <http://www.nofluffjuststuff.com> for more details.

Questions/Comments: Contact Jay Zimmerman: jzimmerman@nofluffjuststuff.com or (303)469-0486.

Desert Southwest Software Symposium 2006

Sheraton Phoenix Sky Harbor Hotel - Phoenix, AZ

July 28 - 30, 2006

(session listing as of 7/27/2006)

Improving Java code quality with code auditing tools by Brian Goetz

Does your program have bugs, despite unit tests, integration tests, and code reviews? You bet. Fortunately, there are some new code auditing tools that can help spot some bugs missed by other approaches.

Introduction to Java threads by Brian Goetz

The Java language included support for threads and concurrency from day 1, but writing correct multithreaded programs is not easy. This session will cover the how and why of using threads in Java.

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.

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.

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.

Applied AOP by Brian Sletten

Most people new to Aspect-Oriented Programming (AOP) are fed up with separation of concerns zealots explaining how great their techniques are at dealing with... logging. Ok, you get it. Logging is a cross-cutting concern that can be appropriately modularized. What else does AOP have to offer? A lot, it turns out. This talk will give an introduction to the motivations of AOP as well as a series of concrete examples drawn from enterprise and client side Java. Come learn how AspectJ-flavored AOP can begin to benefit you immediately either in development or production environments. Learn how to enforce architectural policies, find Swing threading issues, reduce the invasiveness of the Observer design pattern or even improve the reusability of your domain models.

Introducing the Semantic Web by Brian Sletten

Just as the world is feeling comfortable with the Web, Tim Berners-Lee et al inform us that what we have seen so far is just the beginning. His original plans at CERN were larger and grander. The Semantic Web is the new vision of machine-processable documents and metadata to improve search, knowledge discovery and data integration and management. While there are many naysayers chiding such grand visions, there are also pragmatic and useful technologies emerging that can be applied today.

NetKernel : XML Processing for the 21st Century by Brian Sletten

A wise man once said, "XML is like lye. It is very useful, but humans shouldn't touch it." If you've had to incorporate XML into your project by hand, you have probably been burned by getting too close. NetKernel turns this wisdom on its head and encourages you to use XML like the liquid data stream you want it to be. Imagine the simplicity of REST married to the power of Unix pipes. Come see how this open source / commercial product built on a compelling modern architecture can be used to create, manipulate and transform XML.

Effective Teams: The dirty little secret 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.

Java/Ruby Integration with JRuby and ReST by Bruce Tate

You can have rapid web development with Rails without losing access to your critical Java code. With the explosion of the Ruby programming language, more developers will need a strategy for letting Java and

Ruby interoperate. This session explores two strategies: JRuby and Rails-based web services.

Three Technologies to Watch by Bruce Tate

The state of the art is progressing rapidly, and dynamic languages are driving the revolution. Find out about these topics that will be central to programming. We'll discuss continuation servers, metaprogramming frameworks and functional languages.

What's New in Spring 2 by Bruce Tate

In this session, we'll review the new features of Spring 2.0. If you've been using Spring 1.x, you'll want to hear about the improvements.

Where Agile meets Argyle: New processes in established companies by Bruce Tate

Agile programming is a collection of core principles and techniques that allow software developers to create lighter, more responsive applications, and to have fun doing it. Many established organizations are either openly or sub-consciously hostile to many of the principles of Agile development.

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.

JSF: State of the Art by David Geary

In 2005, JSF hit its stride, as evidenced from overwhelming support from both vendors and the open-source community. JSF 1.0 had plenty of holes, but open-source projects have arisen to address those needs. This session takes a look at three of those projects:

- Tomahawk (MyFaces component library)
- Facelets
- Seam

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 2005. Embraced by developers, vendors, and open-source projects, JSF has started to hit its stride. If you haven't come up to speed on JSF basics, this is the place to start.

Killer Web UIs by David Geary

User interfaces are usually the most turbulent aspect of an application during development. Constant tinkering with the UI means constant changes to your code, so as a UI developer, you want to minimize the scope and effects of those code changes. Open-source Java provides two powerful software packages that help you manage UI complexity: Tiles and Sitemesh. Tiles composes webpages from discrete regions of your user interface known as tiles. A tile contains a JSP page for layout and one or more JSP pages for content. Sitemesh decorates webpages with decorators that can be associated with URL patterns. Once you set up your decorators, you can decorate pages that match a decorator's URL pattern.

Shale: Turbo-charge your JSF Apps by David Geary

Struts is the most popular Java-based Web application framework today, but that's rapidly changing. There's a newcomer on the block, a leaner, meaner, better-designed framework loosely based on Struts that's poised to dethrone Struts as the reigning king of Java-based web application frameworks. That framework, of course, is JavaServer Faces. Craig McClanahan, the father of Struts and the co-spec lead for JSF 1.0, has proposed reinventing Struts for Struts 2.0 as a set of services for JSF applications. That new framework, which has no direct ties to Struts as we know it, is called Shale.

All Roads Lead to.. AOP? by Eitan Suez

An exercise in refactoring, playing with Java 5 annotations, varargs, JUnit, and more (see detail description for more).

Cascading Style Sheets: a Programmer's Perspective by Eitan Suez

Today, the Cascading Style Sheets (CSS) specification is well supported by the major browsers (Mozilla, Safari, IE). CSS has become a practical tool for web content publishers that has helped turn heavy, buggy, and hard-to-maintain web sites into lean, clean, and stylish ones. CSS is sometimes stereotyped as a technology geared for graphic designers and artists. I beg to differ: I see CSS as a refactoring tool for content publishers and one that encourages content to become more strongly semantic. Come see a developer's perspective on CSS and how it can be applied to refactor your web content.

Extreme Agility with jMatter by Eitan Suez

The jMatter framework is a modern implementation of the Naked Objects Architectural Pattern using Swing, Hibernate, and deployed with Java WebStart. This open-source framework produces 2-tier workgroup apps (Swing front-ends that talk to rdbms back-ends) intended to be used in a LAN or VPN environment. Developers using a Naked Objects style framework focus on building a behaviourally complete domain model and leave everything else (UI, persistence, etc) to the framework. By focusing on the domain model only, jMatter claims to offer 10x productivity for building Swing workgroup apps. Come discover jMatter in a hands-on presentation where we'll be developing a live application and hold discussions about this new empowering style of producing business applications.

Hibernate by Example by Eitan Suez

This talk covers the core of the Hibernate Object/Relational Mapping framework by example; that is: in a hands-on manner.

XML Data Binding with JiBX by Eitan Suez

JiBX is an open source XML data binding API for Java. JiBX is younger than most other APIs in this space (Castor XML, BEA XMLBeans, JAXB). JiBX's philosophy on data binding is that: [a] databinding should be fast, and [b] databinding frameworks should allow for the divergence and evolution of your codebase from its xml representation. JiBX excels on both counts and consequently is a practical tool for the purpose of data binding. In this session, Eitan will be covering all aspects of Dennis Sosnoski's JiBX framework.

Pragmatic Unit Testing with TestNG and EasyMock by Howard Lewis Ship

You've heard about unit testing but were daunted when it came time to put the pedal to the metal. That's because JUnit is just one tool and there's others you need to learn about, including the wonderful and wierd EasyMock and the easy and powerful TestNG.

Tapestry In Action (Part One) by Howard Lewis Ship

An introduction to the Jakarta Tapestry web application framework, which will explain the concepts and features of the framework while live coding simple applications. Tapestry forms, request cycle, component object mode. The use of several important components (including the powerful Table data grid) will be featured.

Tapestry In Action (Part Two) by Howard Lewis Ship

In Tapestry, components are not an add-on; in fact, anything but! Tapestry components are integral to the entire framework # if something dynamic is going on in a page, there's a component involved.

Pragmatic Tracer Bullets by Jared Richardson

Are your product designs hit or miss? Do you have trouble building a loosely coupled system? Is your code incestuous? Refactoring not an option with your code base? Tracer Bullets help keep your project out of the fire. Tracer Bullet Development: * helps you create great software * lends itself to an iterative cycle * can be used for demos early and often * is easily refactored * allows your teams to work in parallel * makes a very testable system

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.

Software Tools That Make Life Easier: Part One by Jared Richardson

a.. Do you spend more time fighting your tools than writing code? b.. Do you avoid merging your code with your teammates because of #Integration Hell#? c.. Do the same bugs keep sneaking back into your product? d.. Do your builds depend on the roll of the dice? A good set of infrastructure tools can go a long way toward smoothing out these and other problems. Come see how to make your toolset work seamlessly in the background so you can Just Work. We'll cover source code management (SCM), build scripts, automated test harnesses, automatic builds, feature tracking and issue tracking.

Software Tools That Make Life Easier: Part Two by Jared Richardson

This talk is a continuation of Part One of the Tools talk. During Part Two we'll cover Continuous Integration, automated testing, bug tracking, and feature tracking.

Advanced Hibernate by Justin Gethland

Hibernate is easy to get started with, but can sometimes be hard to make efficient or secure. In fact, the

default settings for Hibernate create applications that will run slowly, cause unwanted round trips to the database, and may be more restrictive and/or permissive from a security standpoint than you would otherwise want.

Ajax Architecture by Justin Gehrtland

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

JavaScript for Ajax Programmers by Justin Gehrtland

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.

Spring AOP by Justin Gehrtland

Learn to use Spring AOP, aspect injection, and AspectJ integration. Spring provides powerful support for Aspect-Oriented Programming (AOP), via Spring Advisors Dependency Injection for Aspects Integration with AspectJ

Spring Dependency Injection by Justin Gehrtland

Dependency Injection (DI) is the cornerstone of Spring. The core concept is quite simple, but (surprise!) actual practice can become complex. To take full advantage of Spring DI, you need to understand not only the basics on configuration, but also the container lifecycle model and the various hooks provided by the framework.

Spring Intro by Justin Gehrtland

The Spring framework is one of the fastest growing open source frameworks. New job postings are gaining rapidly, and many customers are adopting Spring instead of heavier alternatives. In this session, we'll introduce Spring. You'll see how Spring can give you much of the power of EJB, without the complexity or pain. Spring uses concepts like dependency injection and aspect oriented programming to ease standard enterprise development. Spring developers write plain, ordinary Java objects (POJOs), instead of sophisticated components. In this session, you'll see a basic Spring application. You'll also see some details about some of the enterprise integration strategies, including: # Spring AOP # Transactions # Persistence # Model/view/controller When the session is over, you won't be an expert, but you should have a much clearer understanding of what Spring does, what it doesn't do, and why it's growing so rapidly.

Spring Security with ACEGI by Justin Gehrtland

Spring offers developers a simpler, more robust method for configuring applications. These benefits extend to security through the ACEGI framework. ACEGI makes the otherwise daunting task of securing your application logical and straightforward. More importantly, through its support for single sign-on provision through Yale's CAS system and its ability to provide instance-level authorization, Spring extends the common security model of most J2EE apps beyond what they are traditionally capable of.

Advanced Enterprise Debugging Techniques by Neal Ford

This session discusses techniques and tools for debugging enterprise applications (without using `System.out.println()`!)

Clean Up Your Code: 10 Java Coding Tricks, Techniques, and Philosophies by Neal Ford

This session delivers 10 techniques for improving your code, whether you are freshly graduated or a grizzled veteran.

Real-world Agile Development by Neal Ford

Lots of developers want to use Agile development technique but don't know where to start. This session discusses how to get started with Agility, the key benefits you can expect, and the pitfalls to avoid.

SOA: Next Wave of Enterprise Development or Return of the Son of CORBA? by Neal Ford

Is Service Oriented Architecture the next wave of distributed computing or just the same old crap in a shiny new package? This session provides an overview of what most people agree is the definition of SOA. I talk about SOA, ESB, CORBA, your MOM, and a bunch of other acronyms.

Testing with Selenium by Neal Ford

This session describes the use and workings of Selenium, the open source web user interface testing tool.

The Productive Programmer by Neal Ford

This session shows you how to become a more productive programmer every day by using tools that you didn't know you already had.

Extend the Customization Possibilities of Your Java App with Scripts by Ted Neward

Ever wished you could just put parts of your program in end-users' hands and let them build the infinite little changes they want? Ever thought about how you might make your application more robust by writing less code, not more? Embed a scripting engine into your application--complete with the safeguards necessary to ensure that users can't do anything they shouldn't be able to--and release yourself from the Principle of Perpetual Enslavement.

Java5: The Language, The Libraries, The VM by Ted Neward

Java5 introduced a whole slew of new features, including annotations (JSR 175), new language features (the enhanced for loop, generics, static imports, and more), new library support (java.lang.instrument, among others), and some interesting enhancements to the virtual machine itself.

Effective Enterprise Java: State Management by Ted Neward

Managing state--both transient state (like your shopping cart) and your durable state (like your order placements, your inventory management forms, and so on)--is tricky in an enterprise application. In this talk, we'll examine some of the trickiness, both high-level and low-.

Pragmatic XML Services by Ted Neward

There's a lot of talk about web services, and most of it falls into one of two categories: lots of low-level talk about vendor-specific tools and extensions, or lots of high-level talk that never shows you a line of code. XML services aren't that hard, and in this talk, we'll see how, why and when to do one.

Get Groovier with Grails by Venkat Subramaniam

Inspired by the Ruby on Rails project, Grails brings the ease of web development and "convention over configuration" to the Java platform. We will learn how to create web applications using Grails, how to integrate it with Hibernate, and how to Ajax it, all using the built in features of Grails. This section assumes that you are familiar with Groovy or you have attended the #Groovy for Java Programmers# session. The session will be example driven with live coding where we will build a web application from scratch.

Groovy for Java Programmers by Venkat Subramaniam

Object-oriented scripting languages, or agile dynamic languages, as some like to call those, are gaining programmers' attention. Groovy bring this excitement to the Java platform with its ability to generate byte code. You can use Groovy instead of Java for some parts of your application. By learning it, you can switch between the languages where you consider fit.

Open Source Tools for Agile Development by Venkat Subramaniam

As a Java developer, you have taken the time to learn the basics of the language and relevant parts of its rich API. However, you need more than that to develop serious industrial strength applications. In this presentation, the speaker will introduce you to a number of open source tools which you can use to improve your application quality and your development process.

Practices of an Agile Developer by Venkat Subramaniam

You have worked on software projects with varying degree of success. What were the reasons for the success of your last project? What were the reasons for those that failed? A number of issues contribute to project success - some non-technical in nature. In this presentation the speaker will share with you practices in a number of areas including coding, developer attitude, debugging, and feedback. The discussions are based on the book with the same title as the talk.

Programming with Mock objects by Venkat Subramaniam

You are convinced that Test Driven Development is good for you and your project. You realize the benefits it has to offer. What's holding you back? All the code and components that your code so heavily depends on is most likely making you wonder if TDD is really for you. We will start out by looking at dependency and dependency inversion. Then we will discuss how mock objects can help separate our code from its dependencies.

Refactoring your code - a key step in agility by Venkat Subramaniam

Refactoring is one of the core practices in Agile Software Development. Refactoring is based on some core principles that apply to more than writing good code. But, what's refactoring? Why should you do it? How do you go about doing that? What tools are available to successfully refactor your App?

Test First Development by Venkat Subramaniam

Do you know that unit testing is more of an act of design than verification? What are its benefits? How do we write effective tests? How does unit testing relate to evolutionary design? How does it help you with refactoring? When should you write your tests? What are the types of tests you could write? These are some of the questions that you would ask if you are interested in Unit Testing. What is a better way to learn than practicing it? In this session the attendees will participate in designing and developing a small yet full application. Instead of PowerPoint slides, you will learn from example. The code you help develop will be available for free download on the speaker's web site.

Working with Rules Engines 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.