

# Southern Ohio Software Symposium 2005

Cincinnati Marriott Northeast - Mason, OH

August 26 - 28, 2005

(session agenda as of 8/23/2005)

Friday, August 26					
	1	2	3	4	5
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Intro to JavaServer Faces Kito Mann	Writing Secure Web Services (with Java and Axis) Justin Gehrtland	Introduction to Spring Bruce Tate	OpenSource Ecosystems Dave Thomas	Making the Most of XML Ben Galbraith
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Migrating from Struts to JSF Kito Mann	Introduction to Hibernate Justin Gehrtland	Where Agile meets Argyle: New processes in established companies Bruce Tate	Ruby for Java Programmers Dave Thomas	Creating Polished Swing Applications Ben Galbraith
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Struts Shale: Struts 2.0? Kito Mann	Advanced Hibernate Justin Gehrtland	Politics of Persistence Bruce Tate	Ruby on Rails Dave Thomas	Advanced Swing: Architecture and Frameworks Ben Galbraith
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: Dave Thomas (Art in Programming)				

Saturday, August 27					
	1	2	3	4	5
8:15 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Performance monitoring in J2EE applications Ramnivas Laddad	Spring Security with ACEGI Justin Gehrtland	Introduction to Portlets Kito Mann	Beyond Java Bruce Tate	SWT Fundamentals Ben Galbraith
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	SOA and ESB: Next Wave of Enterprise Development or Return of the Son of CORBA? Neal Ford	Easy Enterprise Applications with JBoss, Hibernate, AspectJ, and XDoclet Rod Cope	Herdng Racehorses and Racing Sheep Dave Thomas	Stretching Java Bruce Tate	Advanced SWT and JFace Ben Galbraith
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL featuring Dave Thomas, Ben Galbraith, Bruce Tate, Stuart Halloway, Justin Gehrtland, Venkat Subramaniam and Ramnivas Laddad				
2:15 - 3:45 PM	Introduction to Aspect-oriented programming with AspectJ Ramnivas Laddad	Groovy = Java + Ruby + Python for the JVM Rod Cope	The Fallacies of Enterprise Systems Ted Neward	Spring MVC Justin Gehrtland	Creating Killer Graphics and Professional PDFs with XML Ben Galbraith
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Introduction to Aspect-oriented programming with AspectJ Ramnivas Laddad	Advanced Groovy Rod Cope	Working with Java Metadata Ted Neward	Advanced Enterprise Debugging Techniques Neal Ford	AJAX: Creating Next-Generation, Highly Dynamic, Off-line Capable Web Applications with HTML and JavaScript Ben Galbraith

Sunday, August 28					
	1	2	3	4	5
8:15 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Good, Bad and Ugly of Java Generics Venkat Subramaniam	Unit Testing Java with Jython Stuart Halloway	Clean Up Your Code: 10 Java Coding Tricks, Techniques, and Philosophies Neal Ford	Pair Programming for the Single Programmer Scott Davis	Design Pattern Modularization with AOP Ramnivas Laddad
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Test First Development Venkat Subramaniam	Programming Java Concurrency Stuart Halloway	Power Regular Expressions in Java Neal Ford	Guerrilla Web Techniques Scott Davis	Introduction to Web services, 2005 edition Ted Neward
12:30 - 1:15 PM	LUNCH				
1:15 - 2:00 PM	BIRDS OF A FEATHER SESSIONS				
2:00 - 3:30 PM	Prudent OO Design Venkat Subramaniam	Cryptography for Programmers Stuart Halloway	Web Application Security Vulnerabilities Neal Ford	Testing the Web Tier Scott Davis	Effective Enterprise Java: Security Ted Neward
3:30 - 3:45 PM	BREAK				
3:45 - 5:15 PM	Agile Software Development Venkat Subramaniam	Java Platform Security and JAAS Stuart Halloway	Language-oriented Programming and Language Workbenches: Building Domain Languages atop Java Neal Ford	Real World Web Mapping Scott Davis	Effective Enterprise Architecture Ted Neward

# Southern Ohio Software Symposium 2005

Cincinnati Marriott Northeast - Mason, OH

August 26 - 28, 2005

(session agenda as of 8/23/2005)

## **Creating Killer Graphics and Professional PDFs with XML by Ben Galbraith**

You can do some pretty cool things with XML these days (despite what some curmudgeons in the technology world may claim). In the past few years, XML has solidified its place as the lingua franca of data sharing and data manipulation. But XML as a data transfer language is only marginally interesting. Things get really exciting when XML is dynamically transformed into other formats. In this session, I focus on two XML formats which can be readily transformed into high-quality presentation-centric output formats. XSL-FO is a typesetting format for XML that can be readily converted into PDF (or Postscript and some other formats). SVG is a vector graphics language in XML -- a sort of open-source version of the popular Macromedia Flash format. SVG files can be converted into beautiful, completely scalable -- and interactive -- images.

## **SWT Fundamentals by Ben Galbraith**

The Eclipse project's SWT GUI toolkit provides one of the only viable alternatives to Swing for creating so-called rich client applications in Java. Whereas Swing paints its own widgets and has distinguished itself with a complex (and often obtuse) API, SWT relies on the host operating system for widget rendering and sports a simple, clean API. If your goal is to create a Java application that "looks" like a normal Windows application (or OS X, or Linux), SWT will revolutionize your world. In this session, I introduce SWT from the ground up. I start at a high-level, but quickly move into the details of SWT's API. By the presentation's end, attendees will have a solid understanding of SWT.

## **Creating Polished Swing Applications by Ben Galbraith**

Too often, Swing applications are slow, ugly, and hard-to-maintain. It turns out that it doesn't have to be this way. Swing can be used to create highly-responsive, beautiful applications that are very maintainable. If this isn't consistent with your own experience, don't feel bad; it's not very obvious how to make Swing sing.

## **AJAX: Creating Next-Generation, Highly Dynamic, Off-line Capable Web Applications with HTML and JavaScript by Ben Galbraith**

As recent high-profile web apps such as Google's GMail have shown, modern browsers are capable of natively rendering web apps with highly dynamic and compelling UIs - fetching server data without page refreshes, animating and manipulating page contents on-the-fly, even offline use. The line between web and "desktop" apps is blurring.

## **Advanced SWT and JFace by Ben Galbraith**

This session picks up where SWT Fundamentals leaves off. Among the advanced topics I discuss are creating custom SWT widgets and exploring tight native integration. I combine another compelling topic with the advanced SWT material: JFace. SWT is a more akin to AWT than Swing; it's concerned more with wrapping native functionality than providing any high-level abstractions. JFace is an API on top of SWT that provides such abstractions. The combination of SWT and JFace is comparable to Swing. My coverage of JFace includes an introduction to several of its frameworks, such as the Viewer and Window frameworks, along with many examples. Learning JFace will enable you to write complex SWT applications much faster.

## **Making the Most of XML by Ben Galbraith**

For many of us, XML has become a ubiquitous presence in application development, whether parsing, validating, or manipulating it. For many of us, all that XML is coupled with pain, in the form of tedious APIs (like, say, the W3C DOM API) and confusing technologies (oh, I don't know, W3C XML Schema?).

## **Advanced Swing: Architecture and Frameworks by Ben Galbraith**

Are you spending more time plumbing your Swing applications than solving business problems? Has your Swing application grown out of control? This session is for you.

## **Beyond Java by Bruce Tate**

All programming languages have a limited life span, and Java is no different. This is a philosophical session rather than a programming session. Sooner or later, Java will lose its leadership position. This session will explore Java's strengths and weaknesses. We'll try to understand whether conditions are ripe for alternatives to emerge, and what those alternatives may be.

## **Stretching Java by Bruce Tate**

In Stretching Java, we'll look at some of Java's limitations, see how other programming languages solve those problems, and look at how Java developers can implement those ideas in Java using open source frameworks, design strategies, and tools.

### **Introduction to Spring by Bruce Tate**

This session, for the Spring beginner, helps you: # Understand dependency injection and inversion of control # Know the meaning of lightweight containers and Spring # Understand the basic pieces of Spring # See core Spring modules in action, including Persistence, AOP, transactions. Attendees need not know anything about Spring. This session does talk about integration with core J2EE frameworks like JDBC and transactions.

### **Politics of Persistence by Bruce Tate**

This session will help a Java developer choose a persistence framework. After the session, you will # Understand the core strengths and weaknesses of the main persistence frameworks in the Java space # Understand where marketing influences can impact persistence # Know what's going on behind the scenes to impact the persistence pictures # Answer questions about persistence frameworks that might not be mainstream

### **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.

### **Ruby for Java Programmers by Dave Thomas**

Ruby recently enjoyed its tenth birthday. Instead of cake and candles, the community celebrated by releasing a wave of new libraries and frameworks that make Ruby programming even easier. This talk features some of the best of these, as we explore Ruby.

### **OpenSource Ecosystems by Dave Thomas**

Open Source communities produce high quality software with little management and (typically) no pay. Most people looking at open source focus on using this software in their projects.

### **Herding Racehorses and Racing Sheep by Dave Thomas**

Are you frustrated by experts who can't tell you what to do, or by junior team members who refuse to see the big picture? How can you best develop careers: both yours and those of your teammates and managers? How can we learn to apply experience more effectively, and why do the many approaches designed to tame complexity actually end up increasing it?

### **Ruby on Rails by Dave Thomas**

The Ruby on Rails framework has exploded onto the scene over the last few months. Propelled by some genuine benefits, and fueled by a whole lot of controversy, Rails seems here to stay. So, is it a Java killer? (No.) Is it a great way to develop certain classes of web application? (Yes.) Does it really deliver the 10-fold increase in developer productivity that some have claimed? (It depends...)

### **Introduction to Hibernate by Justin Gehtland**

O/RM (Object/Relational Mapping) seeks to eliminate repetitive or tedious work enabling the CRUD (create, read, update, delete) that underlies most applications. Hibernate is a popular, open-source O/RM tool that uses reflection (instead of code generation, like EJB, or bytecode injection, like JDO) to manage your persistence layer. This session will introduce you to Hibernate. After an overview of common usage scenarios, including web and enterprise applications, we'll examine the basics of getting Hibernate running. We'll cover the mapping file format and syntax, including common relational mapping structures. Then, we'll examine the Hibernate API for interacting with the framework. Finally, we'll cover the common architectural decisions you'll have to make as you include this (or any other) O/RM framework.

### **Spring Security with ACEGI by Justin Gehtland**

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.

### **Spring MVC by Justin Gehtland**

The Spring team, as in all things they do, have learned the valuable lessons of the past when introducing a Spring solution. Spring MVC is everything Struts should be, and more besides.

### **Advanced Hibernate by Justin Gehland**

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.

### **Writing Secure Web Services (with Java and Axis) by Justin Gehland**

Web Services are message-oriented. This means that any application intention (the need for security, for transactionality, for reliability, etc.) must be included in the message and not just assumed as external context. The WS-Security specifications are very advanced and currently being used in the wild to create robust, secure web services.

### **Struts Shale: Struts 2.0? by Kito Mann**

With the growing popularity of new Java web frameworks, such as JavaServer Faces, Tapestry, and WebWork, Struts 1.x has lost its competitive edge in the web framework landscape. Recently, Craig McClanahan, the founder of Struts, initiated Struts Shale, a proposed next-generation framework built on top of JavaServer Faces.

### **Introduction to Portlets by Kito Mann**

In late 2003, the Java Community Process released the Portlet API, designed to ease the progress of writing portlets for different portal environments. Using the Portlet API, developers can build reusable application components that work with portal servers from IBM, BEA, Oracle, Vignette, Apache, and other companies and open source organizations.

### **Migrating from Struts to JSF by Kito Mann**

As JavaServer Faces (JSF), the new standard Java web application framework, grows in popularity, development teams are beginning to evaluate different strategies for migrating from Struts to JSF.

### **Intro to JavaServer Faces by Kito Mann**

JavaServer Faces (JSF) is a standard web user interface framework, developed under the Java Community Process (JSR 127), and released in March, 2004. JSF specifies a web user interface component model, complete with server-side event handling, validation, internationalization, page navigation, and declarative mapping between user interface components and Java objects.

### **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.

### **Language-oriented Programming and Language Workbenches: Building Domain Languages atop Java by Neal Ford**

This session shows how to use Java as the building block for domain-specific languages. It discusses the next revolution in programming: language-oriented programming and the nascent tools that support it.

### **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.

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

Are Service Oriented Architecture and Enterprise Service Buses 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 and some of the characteristics of ESBs. I talk about EAI, your MOM, SOA, ESB, and all the other acronyms I can come up with.

### **Web Application Security Vulnerabilities by Neal Ford**

This session highlights common mistakes made by web programmers, stating the problems and avoidance techniques.

### **Design Pattern Modularization with AOP by Ramnivas Laddad**

Design patterns # object oriented, concurrency control, and J2EE # all have certain crosscutting elements present. The obvious result of conventional implementation is unclear implementation that is tedious to implement and tough to change. Aspect-oriented programming (AOP) offers a way to simplify implementation of these design patterns. Further, AOP offers new design patterns of its own that allow for new ways of implementing functionalities.

### **Performance monitoring in J2EE applications by Ramnivas Laddad**

J2EE has become the main new platform for enterprise application deployment. Good performance is an important requirement from the business viewpoint. Supporting this requirement needs application profiling during the development phases and performance monitoring after deploying the application. Come to this session to understand challenges and choice in monitoring J2EE applications.

### **Introduction to Aspect-oriented programming with AspectJ by Ramnivas Laddad**

Aspect Oriented Programming (AOP) enables modularizing implementation of crosscutting concerns that abound in practice: logging, tracing, dynamic profiling, error handling, service-level agreement, policy enforcement, pooling, caching, concurrency control, security, transaction management, business rules, and so forth. Traditional implementation of these concerns requires you to fuse their implementation with the core concern of a module. With AOP, you can implement each of the concerns in a separate module called aspect. The result of such modular implementation is simplified design, improved understandability, improved quality, reduced time to market, and expedited response to system requirement changes. Come to this session and learn all about how AOP can help you simplify developing complex systems.

### **Advanced Groovy by Rod Cope**

It's easy to start using Groovy, but there are lots of goodies that aren't so obvious in the beginning. In this session, we'll cover things like currying, single object iteration, dynamic language extensions, enhancing the JDK, default parameters, advanced closures, active proxies, and more.

### **Easy Enterprise Applications with JBoss, Hibernate, AspectJ, and XDoclet by Rod Cope**

This session demonstrates how to make J2EE(TM) development faster and easier by integrating powerful Open Source tools to produce a rapid application development and deployment infrastructure. See how to harness the power of AOP (Aspect-Oriented Programming) in a practical EJB application to reduce clutter, improve readability, and remove drudgery.

### **Groovy = Java + Ruby + Python for the JVM by Rod Cope**

Groovy is a new dynamic, object-oriented scripting language for the Java Virtual Machine. It has the expressive power of Ruby, the simplicity of Python, and can use all existing Java code. See how to build a Swing GUI, execute Ant scripts in-line, access a database, read and write XML, and more in a few lines of code.

### **Pair Programming for the Single Programmer by Scott Davis**

The full title of this talk is, "The Sound of One Hand Clapping, or How to Pair Program with a Single Programmer -- Scaling XP to Small Projects." Everyone talks about using J2EE for massive projects, but what about the lone wolf developer? Can they still apply the lessons learned from agile development methodologies to their everyday work?

### **Testing the Web Tier by Scott Davis**

Hopefully your test plan involves more than, "Well, it compiled..." JUnit is fast becoming a required part of the modern Java developer's toolkit. Unit testing your Java classes is a great start, but your test plan shouldn't stop there. This talk will introduce several additional testing tools for the web developer -- HttpUnit, Canoo WebTest, and JMeter. These tools allow you to test a live website with no changes to the production code. Even better, you can test sites that have been implemented in technologies other than Java.

### **Guerrilla Web Techniques by Scott Davis**

Frameworks? We don't need no stinkin' web frameworks. OK, so maybe that's overstating the case. Web frameworks do plenty of good things, but sometimes they can also be golden handcuffs. Too many web developers fall into the trap of thinking, "If it can't be done by my web framework, then it simply can't be done."

### **Real World Web Mapping by Scott Davis**

In this presentation, we'll explore the top four mapping sites and show you how to take advantage of their free services. MapQuest, Yahoo Maps, Google Maps, and MSN Virtual Earth all bring slightly different capabilities to the table. These sites allow you to create your own interactive maps with minimum effort and no previous mapping experience. They take care of hosting the mapping data and making it easy to manipulate -- all you have to do is bring a little bit of know-how to the party.

### **Java Platform Security and JAAS by Stuart Halloway**

The Java platform is built from the ground up with security in mind. This talk will introduce the security features of the J2SE, building quickly from the basic classes to realistic examples.

### **Programming Java Concurrency by Stuart Halloway**

Java has always provided a model for concurrency and threads. With Java 1.5, this model received a major facelift. Learn how to use the new concurrency utilities to build responsive, scalable, and correct concurrent applications.

### **Unit Testing Java with Jython by Stuart Halloway**

JUnit is great. Jython is even better. Unit testing libraries look the same everywhere, so why not use the one that lets you get your job done faster?

### **Cryptography for Programmers by Stuart Halloway**

For centuries people have used crypto to build (and break) secure systems. Computers have only raised the pitch of conflict, providing enormous cryptographic power at commodity prices. Most programmers do not write their own crypto libraries, instead relying on the services of an operating system or virtual machine. But even with all this support, building secure systems is a daunting task.

### **Working with Java Metadata by Ted Neward**

As part of JDK 1.5, Java has introduced a facility for developers to create and use custom metadata annotations, as developed by the JSR 175 committee. This represents a radical new shift for the Java programming language, quite possibly larger and farther-reaching than generics or any other language feature.

### **Effective Enterprise Architecture by Ted Neward**

Bring all of your enterprise Java questions to this open forum discussion hosted by the author of #Effective Enterprise Java#, Ted Neward.

### **Effective Enterprise Java: Security by Ted Neward**

Security's become a hot topic among enterprise developers in recent years, but to many developers, security is still the white elephant in the middle of the room. Discussions about security usually begin with, "Uh, we'll worry about that later", or, "Start with two really large prime numbers.....". Security isn't as hard as developers make it out to be, but it is something that developers need to face and recognize.

### **Introduction to Web services, 2005 edition by Ted Neward**

WSDL, and Schema and SOAP, oh my! It's 2005, and the Web services landscape looks even more confusing than it did two years ago, despite all sorts of promises to the contrary. What's it all mean, and how the heck did we get here when the original goal was to try and keep it all simple?

### **The Fallacies of Enterprise Systems by Ted Neward**

There's a set of fallacies that every enterprise developer has fallen for at some point in their enterprise development lives, and unless they've come to realize it early enough, all cause big trouble and painful learning experiences in the long run.

### **Good, Bad and Ugly of Java Generics by Venkat Subramaniam**

Java introduced Generics in the 1.5 version (Java 5). What are the capabilities of Generics? How do you use it? Are there some gotchas in using it? In this example driven presentation, we will start at the basics of generics and look at its capabilities. We will then look at some of the under the hood details on generics implementation. We will then delve into the details of some of the changes to Java libraries to accommodate generics. Finally we will take a look at some restrictions and pitfalls that we need to be familiar with when it comes to practical and prudent use of generics.

### **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.

### **Prudent OO Design by Venkat Subramaniam**

Is your code object-oriented? Developing with objects involves more than using languages like Java, C#, C++ or Smalltalk for that matter. From time to time, the OO paradigm stumps even expert developers. Agile programming becomes a mere act of hack if we code without knowing the OO principles. What are these principles # the ones that influence your design? In this presentation the speaker will present some of the challenges that are fundamental in nature. Then he will present OO Design principles and good practices for prudent development of OO code.

### **Agile Software Development by Venkat Subramaniam**

You have probably worked on a few projects that have succeeded and then a few that have failed. What were the factors that influenced the success or failure of those projects? You want to develop a system that is robust, maintainable, within budget, of high quality and with fewer defects. How can you realize those goals? What steps, process, tools you can use or follow to achieve this. In this session, the speaker will present a number of approaches that lead to successful development. He will also present his personal experience with those in implementing software projects. Attendees are encourage and expected to present their views on what has or has not worked for them.