

# Central Ohio Software Symposium 2006

Columbus Marriott North - Columbus, OH

May 19 - 21, 2006

(session agenda as of 5/17/2006)

Friday, May 19					
	1	2	3	4	5
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Spring Intro Justin Gehlrand	JavaScript Exposed: There's a Real Programming Language in There! (Part 1) Glenn Vanderburg	The State of AOP Ramnivas Laddad	JavaServer Faces: A Whirlwind Tour David Geary	Practices of an Agile Developer Venkat Subramaniam
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Spring Dependency Injection Justin Gehlrand	JavaScript Exposed: There's a Real Programming Language in There! (Part 2) Glenn Vanderburg	Enterprise AOP with AspectJ Ramnivas Laddad	JSF: State of the Art David Geary	Refactoring your code - a key step in agility Venkat Subramaniam
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Spring Security with ACEGI Justin Gehlrand	Java Collections Power Techniques Glenn Vanderburg	Testing Strategies for Web Applications Ramnivas Laddad	Effective Teams: The dirty little secret Bruce Tate	Open Source Tools for Agile Development Venkat Subramaniam
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	KEYNOTE BY NEAL FORD				

Saturday, May 20					
	1	2	3	4	5
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Introduction to Hibernate Justin Gehlrand	Modern Project Infrastructures Glenn Vanderburg	Shale: Turbo-charge your JSF Apps David Geary	What's New in Spring 2 Bruce Tate	Real-world Agile Development Neal Ford
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Advanced Hibernate Justin Gehlrand	Java Performance Myths Glenn Vanderburg	Ajaxian Faces David Geary	Where Agile meets Argyle: New processes in established companies Bruce Tate	Portal Standards and implementation Venkat Subramaniam
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	Ajax Architecture Justin Gehlrand	The Productive Programmer Neal Ford	Killer Web UIs David Geary	Politics of Persistence Bruce Tate	Performance Monitoring in J2EE Applications Ramnivas Laddad
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Java Platform Security and JAAS Justin Gehlrand	Spring AOP in Depth Ramnivas Laddad	Testing with Selenium Neal Ford	Three Technologies to Watch Bruce Tate	Working with Rules Engines Venkat Subramaniam
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS				

Sunday, May 21					
	1	2	3	4	5
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Dependency Management Techniques Kirk Knoernschild	Understanding the Role of an ESB Mark Richards	Software Tools That Make Life Easier: Part One Jared Richardson	Groovy for Java Programmers Venkat Subramaniam	Web Application Security Vulnerabilities Neal Ford
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	From Code to Architecture Kirk Knoernschild	Java EE Command Pattern Architecture Mark Richards	Software Tools That Make Life Easier: Part Two Jared Richardson	Java 5 Features. What's in it for you? Venkat Subramaniam	Introduction to Tapestry Neal Ford
12:30 - 1:15 PM	LUNCH				
1:15 - 2:00 PM	EXPERT PANEL DISCUSSION				
2:00 - 3:30 PM	Benefits of the Build - A Case Study in Continuous Integration Kirk Knoernschild	EJB 3 Part 1: Core Spec and Spring Comparison Mark Richards	Software Development Techniques Jared Richardson	Programming with Mock objects Venkat Subramaniam	Pragmatic Extreme Programming Part 1: Planning & Design Neal Ford
3:30 - 3:45 PM	BREAK				
3:45 - 5:15 PM	GOF Patterns Applied Kirk Knoernschild	EJB 3 Part 2: Java Persistence API (JPA) Mark Richards	Pragmatic Tracer Bullets Jared Richardson	Prudent OO Design Venkat Subramaniam	Pragmatic Extreme Programming Part 2: Architecture, Coding, and Testing Neal Ford

# Central Ohio Software Symposium 2006

Columbus Marriott North - Columbus, OH

May 19 - 21, 2006

(session agenda as of 5/17/2006)

## **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.

## **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.

## **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.

## **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

## **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.

## **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.

## **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.

## **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.

## **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

## **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.

### **Java Collections Power Techniques by Glenn Vanderburg**

The Java Collections framework is a cornerstone of Java development. It's been a part of J2SE for six years now. Every Java developer knows it#how to create Lists, Maps, and Sets, how to put things into them and take things out, and how to iterate over the contents. But there's a lot more to the collections framework than that -- and very few programmers really know how to exploit the power that's just under the surface.

### **Modern Project Infrastructures by Glenn Vanderburg**

The support infrastructure for your software project is a crucial factor for success. A new generation of tools offers significant benefits over their predecessors. This talk discusses how to choose the right mix of tools for a top-shelf project infrastructure.

### **Java Performance Myths by Glenn Vanderburg**

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.

### **JavaScript Exposed: There's a Real Programming Language in There! (Part 1) by Glenn Vanderburg**

With the sudden importance of Ajax, it's time to take JavaScript seriously. That means learning it the right way: looking at the fundamentals of the language and surveying its strengths and weaknesses, instead of just copying other people's poorly written examples.

### **JavaScript Exposed: There's a Real Programming Language in There! (Part 2) by Glenn Vanderburg**

Building on part 1, this talk dives deep into JavaScript's object model. We'll see how it differs from more mainstream object-oriented languages, and why. We'll explore how to hide some of those differences, as well as the reasons you might not want to. Additionally, we'll cover useful tools for JavaScript testing, debugging, and profiling.

### **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.

### **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 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.

### **Ajax Architecture by Justin Gehtland**

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.

### **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.

### **Java Platform Security and JAAS by Justin Gehrtland**

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. You will learn the core APIs: SecurityManager, AccessController, Permissions and Policy JAAS Subjects, Principals, and LoginModules

### **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 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 Hibernate by Justin Gehrtland**

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.

### **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.

### **Benefits of the Build - A Case Study in Continuous Integration by Kirk Knoernschild**

Agile processes such as XP and RUP advocate continuous integration, where shorter iterations produce an incremental and functional growth of the system. The fundamental component of any Continuous Integration strategy is an automated and repeatable build. In addition to ensuring your application is always in a functional state, a robust build strategy enables a number of other important lifecycle activities.

### **Dependency Management Techniques by Kirk Knoernschild**

Why is software so difficult to change? When you establish your initial vision for the software's design and architecture, you imagine a system that is easy to modify, extend, and maintain. Unfortunately, as time passes, changes trickle in that exercise your design in unexpected ways. Unlike what you had anticipated, each change begins to resemble nothing more than another hack, until finally the system becomes a tangled web of code that few developers care to venture through. Eventually, modifications to the software intended to improve the system have the opposite affect of breaking other parts of the system. The software is beginning to rot.

### **From Code to Architecture by Kirk Knoernschild**

The code we write has a tremendous impact on our software architecture. In fact, the code is our architecture. Not all of it, of course, but a good share of it. The problem is that we often don't fully

comprehend the architectural impact of our code at the time we create it. One poorly designed class or method can severely impact the resiliency, stability, extensibility, and maintainability of your software. There are ways we can create flexible architectures...but we have to emphasize loose coupling among system components.

### **GOF Patterns Applied by Kirk Knoernschild**

Design Patterns are proven and powerful techniques that can help improve the resiliency, maintainability, and extensibility of your applications. However, overusing or misapplying patterns is a common mistake often times resulting in applications that are over-architected, and resemble a tangled web of classes. How can patterns be applied to achieve the goal of better software?

### **EJB 3 Part 2: Java Persistence API (JPA) by Mark Richards**

In addition to providing a simplified API, the new EJB 3 specification (JSR-220) defines a standard ORM Java Persistence API (JPA) that replaces those nasty Entity Beans that were part of the EJB 2.x specification. 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 see how to use the JPA for simple queries, complex queries, and finally stored procedures. This session is part two of a two-part EJB 3 session.

### **Java EE Command Pattern Architecture by Mark Richards**

Tired of dealing with EJBs but cannot use other frameworks like Spring? How would you like to replace all of your remote Stateless Session Beans with POJOs and still access them remotely within Java EE? By using the Java EE Command Pattern we can write EJBs as POJOs and solve many of the issues facing EJB, including testability, configuration complexity, and performance, and still remain within our behemoth Java EE environment. The Java EE Command Pattern is a simple pattern that can significantly reduce the complexity of large-scale Java EE enterprise applications. In this session we will explore the numerous issues facing a typical EJB architecture and learn how the use of the Java EE Command Pattern can solve these issues. We will walk through the different design alternatives and see how the command pattern is implemented. Through interactive coding examples you will learn what components make up the Command Pattern framework and what simple coding changes are required to convert a complex EJB-based application to a remotable POJO-based application.

### **Understanding the Role of an ESB by Mark Richards**

The Enterprise Service Bus is an integral part of any Service-Oriented Architecture. It is the glue that binds the business services to the client applications. There are many ESB third-party products and solutions in the marketplace, but in most cases these products only serve to further confuse us in terms of what an ESB is, particularly when you consider that an ESB is really an architectural component that has many different implementations. In this session we will take a detailed, product-agnostic look at the role of an ESB and the capabilities an ESB must provide. Through this session you will learn what an ESB is, the role of an ESB, what capabilities it provides, and the various ways an ESB can be implemented. We will also take a close look at the Java Business Integration (JBI) specification (JSR-208) and see what impact it will have with the ESB world. With the information from this session you will learn how to determine your own specific requirements for an ESB and then match those requirements to the product space rather than having the tail wag the dog!

### **EJB 3 Part 1: Core Spec and Spring Comparison 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 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. During the session I will demonstrate the new features of EJB 3 through interactive coding examples. We will then look at how the EJB 3 specification differs from the Spring Framework, where each is useful, and speculate as to what will happen in the future with these two frameworks. This session is part one of a two-part EJB 3 session (part two covers the new Java Persistence API).

### **Pragmatic Extreme Programming Part 1: Planning & Design by Neal Ford**

This session begins a detailed discussion about how to actually get XP done in the real world (and what to tell your boss). This session includes artifacts (project tracking sheets, code coverage reports, etc.) from real XP projects.

### **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.

### **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.

### **Introduction to Tapestry by Neal Ford**

This session delves into details about building web applications with Tapestry, covering configuration, templates, and separation of concerns.

### **Pragmatic Extreme Programming Part 2: Architecture, Coding, and Testing by Neal Ford**

Continues the discussion from Part 1, focusing on how to keep the benefits of XP without sacrificing it's effectiveness. This session shows real artifacts of XP in action.

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

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

### **Testing Strategies for Web Applications by Ramnivas Laddad**

Ever wondered if you can automate testing of your web application, but couldn't produce a satisfactory solution? If so, this is the session for you! Attend this session to understand the alternatives you have for unit and functional testing of web applications.

### **Spring AOP in Depth by Ramnivas Laddad**

Support for aspect-oriented programming is an important part of the Spring framework. It is the AOP support that allows keeping implementation of functionality such as transaction management and security out of your POJOs. While many developers only use aspects provided with Spring, once you understand how it all works, you can make a better use of those aspects, extend them, and write brand new aspects.

### **The State of AOP by Ramnivas Laddad**

A lot is happening in the field of Aspect-oriented programming (AOP). AspectJ and AspectWerkz, the two leading AOP implementations, have merged, bringing in their respective strengths. The merged version, AspectJ 5, adds many new features aimed at simplifying writing and deploying aspects. The new features include an annotation-based and XML-based syntax to define aspects, support for new Java 5 concepts, and load-time weaving. The tools support for AOP continues to improve, as well. Further, the most popular IOC framework, Spring, enables integrating aspects written in AspectJ. There is also serious discussion and preliminary work going on to support AOP right into the VM itself. All in all, there is a lot to learn about the changes in the exciting field of AOP. This session is designed to help you get up to date with all these changes.

### **Performance Monitoring in J2EE Applications by Ramnivas Laddad**

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

### **Enterprise AOP with AspectJ by Ramnivas Laddad**

Enterprise application development is a gold mine for applications of AOP. There are many crosscutting concerns found in a typical enterprise application, ranging from well-known security and transaction management to application- and technology-specific concerns. Using AOP leads to implementations that are easy to understand and easy to change.

### **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.

### **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.

### **Java 5 Features, What's in it for you? by Venkat Subramaniam**

A number of new features have been introduced in Java. What benefit do these features offer you. Are there issues with using these features. For instance, when should you use annotation? The objective of this presentation is not simply to introduce you to the features, but to the effective use of these as well.

### **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.

### **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.

### **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.

### **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.

### **Portal Standards and implementation by Venkat Subramaniam**

Portals and Portlets allow you to personalize your web application. However, developing and deploying portlets across different portals can be a challenge. What is WSRP and JSR-168. How are these related and how are these different? Are these competing technologies or do they work with each other?

### **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?