

# Lone Star Software Symposium 2005

Crowne Plaza Park Central - Dallas, TX

November 04 - 06, 2005

(session agenda as of 11/1/2005)

Friday, November 4					
	1	2	3	4	5
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	An Introduction to JavaServer Faces David Geary	Programming Java Concurrency Stuart Halloway	Cascading Style Sheets: a Programmer's Perspective Eitan Suez	Runtime Code Generation for Java and Beyond Glenn Vanderburg	OpenSource Ecosystems Dave Thomas
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Felix: A bag of Tricks for Java Server Faces David Geary	Unit Testing Java with Jython and JRuby Stuart Halloway	XML Data Binding with JiBX Eitan Suez	Under the Hood of Java Memory Management Glenn Vanderburg	Herding Racehorses and Racing Sheep Dave Thomas
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Shale: The Next Struts? David Geary	Cryptography for Programmers Stuart Halloway	The State Machine Compiler Eitan Suez	Ruby for Java Programmers Dave Thomas	Programming with Mock objects Venkat Subramaniam
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote by Dave Thomas, entitled "Art in Programming"				

Saturday, November 5					
	1	2	3	4	5
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Killer Web UIs David Geary	SOA and ESB: Next Wave of Enterprise Development or Return of the Son of CORBA? Neal Ford	Introduction to Java Reflection Stuart Halloway	Introduction to Ajax Ben Galbraith	Naked Objects Applied Eitan Suez
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Good, Bad and Ugly of Java Generics Venkat Subramaniam	Web Application Security Vulnerabilities Neal Ford	Java Platform Security and JAAS Stuart Halloway	Ajxian JavaScript Frameworks Ben Galbraith	Transitioning to Agile: A Dozen Keys to Success Mike Cohn
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	BIRDS OF A FEATHER SESSIONS				
2:15 - 3:45 PM	Java Metadata Jason Hunter	Programming with Aspects Venkat Subramaniam	Ruby on Rails Dave Thomas	Making the Most of XML Ben Galbraith	Overview of Agile Estimating and Planning Mike Cohn
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	New Features in Java 5 Jason Hunter	Java Collections Power Techniques Glenn Vanderburg	Clean Up Your Code: 10 Java Coding Tricks, Techniques, and Philosophies Neal Ford	Being Productive with Java in the Enterprise Ben Galbraith	Project Economics: Selecting and Prioritizing High Value Projects Mike Cohn

Sunday, November 6					
	1	2	3	4	5
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	Advanced Enterprise Debugging Techniques Neal Ford	Groovy for Java Programmers Venkat Subramaniam	Introduction to Spring Bruce Tate	Introduction to Web services, 2005 edition Ted Neward	Salvaging Struggling Projects: Digging for Gold Instead of Digging a Deeper Hole Mike Cohn
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	An Introduction to XQuery Jason Hunter	Power Regular Expressions in Java Neal Ford	Introduction to Hibernate Bruce Tate	The Fallacies of Enterprise Systems Ted Neward	Agile Methodologies Venkat Subramaniam
12:30 - 1:15 PM	LUNCH				
1:15 - 2:00 PM	EXPERT PANEL				
2:00 - 3:30 PM	Extreme Web Caching Jason Hunter	Thinking Inside the Box: Building Spring-Enabled Portlet Applications Craig Walls	Language Oriented Programming Part 1: Theory Neal Ford	Effective Enterprise Java: Security Ted Neward	Where Agile meets Argyle: New processes in established companies Bruce Tate
3:30 - 3:45 PM	BREAK				
3:45 - 5:15 PM	Open Source from the Inside Jason Hunter	At Your Service: Service-Oriented Spring Craig Walls	Stretching Java Bruce Tate	Effective Enterprise Architecture Ted Neward	Pragmatic Extreme Programming Neal Ford

# Lone Star Software Symposium 2005

Crowne Plaza Park Central - Dallas, TX

November 04 - 06, 2005

(session agenda as of 11/1/2005)

## **Ajaxian JavaScript Frameworks by Ben Galbraith**

In the "Introduction to Ajax" session, we discuss what Ajax is, how it works, and how others are using it. This session goes deeper into Ajax by reviewing the existing JavaScript frameworks that aim to make it easier.

## **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?).

## **Being Productive with Java in the Enterprise by Ben Galbraith**

It sounded like such a good idea back in the mid-nineties: based the Java platform on a standards-based, open community, and let anyone participate. There is no question that Sun's strategy for Java's stewardship via the JCP and sponsored open-source has yielded some enormous benefits. However, these have not been enjoyed without tremendous cost.

## **Introduction to Ajax by Ben Galbraith**

Ajax -- called DHTML just a few months ago -- has revolutionized (or "radically iterated", if you like) web application development in the short few months since the term was coined. What is it all about? Why are we excited about a set of capabilities that have been sitting in our browser for years? What can you do with it? And, how can you do it?

## **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.

## **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.

## **Introduction to Hibernate by Bruce Tate**

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.

## **At Your Service: Service-Oriented Spring by Craig Walls**

Where Spring promotes loose-coupling between your application objects, service-oriented architecture (SOA) encourages loose-coupling between applications that interact with each other.

## **Thinking Inside the Box: Building Spring-Enabled Portlet Applications by Craig Walls**

Windows changed everything. Back in the days of MS-DOS, you could only run one application at a time. Switching between writing a letter and balancing your checkbook involved closing a word processor and opening a spreadsheet. But now you can be running dozens of applications simultaneously, each inside its own window. And now switching from one application to another may be as simple as a shift of your eye or a click of the mouse button.

## **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?

### **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.

### **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...)

### **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.

### **An Introduction to JavaServer Faces by David Geary**

There are a lot of Java-based web application frameworks, but how many of them are: 1. Based on the most popular open-source framework (Struts), and 2. The standard that must be supported by every J2EE 1.5 container? JavaServer Faces (JSF) debuted in the Spring of 2004. Throughout the rest of 2004, JSF gained momentum with a handful of books and a growing user community that includes the popular MyFaces open-source JSF implementation which has recently moved from SourceForge to Apache. Perhaps the most telling sign of the times is Craig McClanahan's proposal for Struts 2.0, code named Shale, which reinvents Struts as a set of services for JSF applications.

### **Felix: A bag of Tricks for Java Server Faces by David Geary**

Okay, so you know a little about JSF. You understand managed beans, action outcomes and how to attach standard JSF validators to components in a JSP page. But there is a great deal of functionality that the average web application supports that JSF doesn't provide out of the box. For example, wouldn't you like to have JSF automatically place asterisks in front of labels for required fields? You are going to implement client-side validation, which JSF does not support out of the box, aren't you? Of course, you're going to test your application, right? And don't forget to trap unauthorized use of the back button.

### **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: The Next Struts? 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.

### **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.

### **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.

### **The State Machine Compiler by Eitan Suez**

Classes will often bear various states. Examples include a user who may be "logged in" or "logged out," a bill that is "open" or "paid," or potentially a more complex situation where an object obeys a set of complex rules that determines which of a number of possible states that object is in. The Gang of Four gave us the State Pattern, a fairly straight-forward mechanism for developers to model and implement the behaviour of stateful objects. The State Pattern is only the beginning of the story. Robert Martin developed the State Machine Compiler and has taken the job of developing and maintaining stateful systems to a new level. Today, SMC is a well-maintained open source project hosted on sourceforge.net. Come learn about SMC, a fundamental tool for implementing stateful classes and systems that every software developer should have in his toolchest.

### **Naked Objects Applied by Eitan Suez**

Join Eitan in this hands-on session on Naked Objects. This session uses the "learning by doing" approach to learning an API or framework. Naked Objects is a powerful tool that can give you a significant advantage in the development of business systems. It gives you the ability to prototype a software application so quickly that it can be performed during information gathering phases of a project. It gives you the power to codevelop the core business model of your application with a non-developer business expert at your side. No prerequisite knowledge of Naked Objects is required.

### **Under the Hood of Java Memory Management by Glenn Vanderburg**

Most of the time, Java's automatic memory management works really well#it's one of the things that makes programming in Java a pleasant and productive experience, and it's nice that we don't have to worry about managing memory manually. However, although it's usually nice to ignore memory management, occasionally we have to pay close attention. Sometimes we need to take control of certain aspects of memory management. Sometimes Java programs do exhibit memory leaks, or unacceptably long garbage collection pauses, or very poor overall performance. But because Java's memory management is supposed to be "fully automatic," it can be difficult to find out what's really going on inside the VM.

### **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.

### **Runtime Code Generation for Java and Beyond by Glenn Vanderburg**

Every now and then, it's really helpful to be able to generate a new Java class at runtime. Some problems just can't be solved any other way. It's one of those troublesome tasks: it's fairly tricky to do, and you only need to do it occasionally#but when you need it, you really need it (and usually you need it yesterday). So you have to start essentially from scratch, learning about how to do it on the fly, under pressure.

### **Java Metadata by Jason Hunter**

Java's new Metadata facility introduced in J2SE 5.0 defines a way to attach decorations to classes, fields, methods, and even packages that can be extracted by the compiler or runtime tools to provide advanced functionality. Think of metadata as an extended @deprecated flag, or think of XDoclet++. In this tutorial session you'll learn how Metadata fits in the Java platform (and how it compares to the C# platform). We'll cover how to use the metadata attributes provided in the core J2SE libraries and how to write your own. We'll also show a bit of what's coming in JSR-181, tasked to define standard metadata attributes for web services.

### **Extreme Web Caching by Jason Hunter**

Web Caching is very important for high traffic, high performance web site but few people know all the professional-level strategies. In this talk I'll share some of the tricks of the trade, including advanced tips from Yahoo's Mike Radwin. We'll start with the basics: using client-side caches, conditional get, and proxies. Then we'll talk about more advanced features: how best to handle personalized content, setting up an image caching server, using a cookie-free domain for static content, and using randomization in URLs for accurate hit metering or sensitive content.

### **New Features in Java 5 by Jason Hunter**

The new Java 5 release introduces a number of significant Java language enhancements: generics, typesafe enums, autoboxing, an enhanced "for" loop, a static import facility, and a general-purpose metadata facility. This talk gives an overview of the changes and helps you understand what all the funny new syntax means.

### **An Introduction to XQuery by Jason Hunter**

XQuery is a new language from the W3C that lets you query and manipulate XML -- or anything that can be represented as XML, such as relational databases. As a Java developer -- especially a server-side Java developer -- XQuery is key to searching and manipulating large XML repositories or performing any XML-centric task. This talk introduces XQuery. I'll explain the XQuery language; I'll show how to call XQuery from Java; and as the creator of JDOM, I'll also explain when to use XQuery instead of JDOM, and when to use both.

### **Open Source from the Inside by Jason Hunter**

Open source isn't about a license, it's about human interaction and individual motivation. I've seen open source from all sides. I've been an individual contributor and a project leader. I've worked on commercial and open source efforts, and have both helped commercial projects go open and designed ways for open projects to absorb commercial codebases. I've been on the front lines in the Apache/Sun negotiations on open source Java that ended on the JavaOne keynote stage with Scott McNealy. In this talk, I'd like to share my favorite stories in and around open source and the lessons they teach us.

### **Overview of Agile Estimating and Planning by Mike Cohn**

Estimating and planning are key skills. A good plan helps both the organization and the developers working on the project. In this session you'll learn how an easy and effective approach to estimating and planning that can help you create more realistic plans.

### **Salvaging Struggling Projects: Digging for Gold Instead of Digging a Deeper Hole by Mike Cohn**

Projects struggle for many reasons#overly aggressive deadlines, unproven technologies, scope creep, team dynamics, communication problems, and inter-team coordination are just some of the reasons. If not given attention, these problems can ultimately cause a project to fail entirely. However, if you act early and in the right way, most struggling projects can be turned around.

### **Project Economics: Selecting and Prioritizing High Value Projects by Mike Cohn**

Almost all of us have worked on too many projects that have failed because of economic reasons rather than technical reasons. Just as the technical team is required to estimate the effort that will go into a project, a marketing or product management team should estimate the benefits of doing the project. Benefits can come in the form of additional sales, increased customer retention, increased operating efficiencies, and so on.

### **Transitioning to Agile: A Dozen Keys to Success by Mike Cohn**

Transitioning to an agile process from a traditional process is fraught with potential dangers. Attend this class and learn the dozen things you absolutely must do in order to succeed.

### **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.

### **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.

### **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.

### **Pragmatic Extreme Programming by Neal Ford**

This session talks about how to actually get XP done in the real world (and what to tell your boss). This session includes artifacts (like project tracking sheets) from real XP projects.

### **Advanced Enterprise Debugging Techniques by Neal Ford**

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

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

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

### **Language Oriented Programming Part 1: Theory 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.

### **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.

### **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.

### **Introduction to Java Reflection by Stuart Halloway**

Reflection is writing code that manipulates itself. Well-written reflective code automates a broad class of repetitive, error-prone programming tasks. Poorly-written reflective code obfuscates programs and destroys the benefits of the type system. We'll focus on the former.

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

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

### **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.

### **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.

### **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.

### **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?

### **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.

### **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.

### **Programming with Aspects by Venkat Subramaniam**

OOP is currently the most popular and practical software development approach. One of the reasons for its popularity is the ability to separate concerns, focusing on behaviors as they relate to business or technical issues. But this very same capability reaches its limits in OOP when it comes to global and crosscutting concerns. Aspect Oriented Programming is receiving attention for its ability to address these concerns. How is it similar and different from OOP? What are the traits of AOP and what are the limitations of utilizing it in projects? In this interactive presentation, the speaker will introduce AOP, discuss its capabilities and benefits, and share his cautious optimism on how to put it to use in your projects.

### **Agile Methodologies by Venkat Subramaniam**

Agile development is picking up steam. You have heard about eXtreme Programming(XP). What other Agile methodologies are you familiar with and what do they bring of interest or significant to the table of Agility? More important, why should you learn about these different methodologies instead of simply focusing on one? There is no one shoe that fits all. Any methodology that requires you to follow it in totality and not let you adapt is rather dogmatic, not pragmatic. To be effective we have to take the best of different approaches and apply to our projects base on our specific needs.

### **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.

### **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.