

# Northern Virginia Software Symposium 2006

Sheraton Reston - Reston, VA

November 03 - 05, 2006

(session listing as of 11/5/2006)

| Friday, November 3, 2006 |   |  |   |  |  |  |
|--------------------------|---|--|---|--|--|--|
|                          | Ballroom A-C  | Meeting Room 9-10  | Meeting Room 8  | Meeting Room 7   | Meeting Room 6   | Meeting Room 5   |
| 11:30 - 1:00 PM          | REGISTRATION  |  |   |  |  |  |
| 1:00 - 1:15 PM           | WELCOME   |  |   |  |  |  |
| 1:15 - 2:45 PM           | Ajax Architecture<br>Stuart Halloway                  | Java Performance Myths<br>Brian Goetz                            | Java NIO<br>Brian Pontarelli                                    | Agile Requirements with<br>User Stories<br>Pete Behrens    | Applied REST<br>Brian Sletten  | Open Source Tools for<br>Agile Development<br>Venkat Subramaniam |
| 2:45 - 3:15 PM           | BREAK   |  |   |  |  |  |
| 3:15 - 4:45 PM           | JavaScript for Ajax<br>Programmers<br>Stuart Halloway | Introduction to Java<br>threads<br>Brian Goetz                   | Jini - Not just for your<br>toaster anymore<br>Brian Pontarelli | Agile Estimating, Planning<br>and Tracking<br>Pete Behrens | NetKernel : XML<br>Processing for the 21st<br>Century<br>Brian Sletten | Spring into Unit Testing<br>Venkat Subramaniam                   |
| 4:45 - 5:00 PM           | break   |  |   |  |  |  |
| 5:00 - 6:30 PM           | Spring Dependency<br>Injection<br>Stuart Halloway     | Structuring concurrent<br>applications in JDK 5.0<br>Brian Goetz | ACEGI Authentication - The<br>AJAX way<br>Brian Pontarelli      | Agile Tooling: Team to<br>Enterprise<br>Pete Behrens       | De-mystifying JSF<br>Ed Burns  | Working with Rules<br>Engines<br>Venkat Subramaniam              |
| 6:30 - 7:15 PM           | DINNER  |  |   |  |  |  |
| 7:15 - 8:00 PM           | Keynote: Keynote: How Hard Can It Be? - Andy Hunt     |  |   |  |  |  |

| Saturday, November 4, 2006 |   |  |   |  |  |  |
|----------------------------|---|--|---|--|--|--|
|                            | Ballroom A-C  | Meeting Room 9-10  | Meeting Room 8  | Meeting Room 7   | Meeting Room 6   | Meeting Room 5   |
| 8:00 - 9:00 AM             | BREAKFAST   |  |   |  |  |  |
| 9:00 - 10:30 AM            | Improving Java code<br>quality with code auditing<br>tools<br>Brian Goetz | Agile Enterprise<br>Architecture: The role of the<br>architect<br>Pete Behrens | Enterprise Grade Ajax with<br>JSF<br>Ed Burns   | Message Driven POJOs<br>Mark Fisher                      | JRuby<br>Stuart Halloway                               | The State of AOP<br>Ramnivas Laddad                        |
| 10:30 - 11:00 AM           | BREAK   |  |   |  |  |  |
| 11:00 - 12:30 PM           | Refactoring Your Wetware<br>Andy Hunt                                     | Effective Hibernate<br>Scott Leberknight                                       | The Java Memory Model<br>Brian Goetz  | Programming with Mock<br>objects<br>Venkat Subramaniam   | Prototype: Ajax and<br>JavaScript++<br>Stuart Halloway | Spring AOP in Depth<br>Ramnivas Laddad                     |
| 12:30 - 1:30 PM            | LUNCH   |  |   |  |  |  |
| 1:30 - 3:00 PM             | Pragmatic Learning<br>Andy Hunt   | The Busy Java Developer's<br>Guide to ClassLoaders<br>Ted Neward               | Clean Up Your Code: 10<br>Java Coding Tricks,<br>Techniques, and<br>Philosophies<br>Neal Ford | New Features in Java 5<br>Jason Hunter                   | Spring Security with ACEGI<br>Stuart Halloway          | Holistic Testing<br>Scott Davis                            |
| 3:00 - 3:15 PM             | BREAK   |  |   |  |  |  |
| 3:15 - 4:45 PM             | Java Metadata<br>Jason Hunter   | Essential db4o<br>Ted Neward   | Testing with Selenium<br>Neal Ford  | Practices of an Agile<br>Developer<br>Venkat Subramaniam | What's New in Rails 1.1?<br>Stuart Halloway            | Domain Driven Design with<br>AOP and DI<br>Ramnivas Laddad |
| 4:45 - 5:30 PM             | BIRDS OF A FEATHER  |  |   |  |  |  |

| Sunday, November 5, 2006 |  |   |  |  |   |  |
|--------------------------|--|---|--|--|---|--|
|                          | Ballroom A-C   | Meeting Room 9-10   | Meeting Room 8   | Meeting Room 7   | Meeting Room 6  | Meeting Room 5   |
| 8:00 - 9:00 AM           | BREAKFAST  |   |  |  |   |  |
| 9:00 - 10:30 AM          | Techniques in Architecture<br>Agility<br>Mark Richards | Spring/Hibernate<br>Integration Basics<br>Scott Leberknight   | Groovy: Greasing the<br>Wheels of Java<br>Scott Davis  | Performance Monitoring in<br>J2EE Applications<br>Ramnivas Laddad      | Refactoring, Hands On!<br>Steve Metsker                       | Pragmatic XML Services<br>Ted Neward   |
| 10:30 - 11:00 AM         | BREAK  |   |  |  |   |  |
| 11:00 - 12:30 PM         | Extreme Web Caching<br>Jason Hunter                    | Introduction to TestNG, the<br>next generation testing<br>framework for developers<br>Andrew Glover | Get Groovier with Grails<br>Venkat Subramaniam         | Effective Java Persistence<br>Using Spring and iBATIS<br>Mark Richards | Easing into Agile<br>Scott Davis                              | Java Annotations: From<br>Definition to Consumption<br>Ted Neward                        |
| 12:30 - 1:15 PM          | LUNCH  |   |  |  |   |  |
| 1:15 - 2:00 PM           | EXPERT PANEL DISCUSSION                                |   |  |  |   |  |
| 2:00 - 3:30 PM           | Greasemonkey: Ajax For<br>One<br>Jason Hunter          | Pragmatic Extreme<br>Programming Part 1:<br>Planning & Design<br>Neal Ford                          | Real World Web Services<br>Scott Davis                 | EJB 3 Part 1: Core Spec<br>and Spring Comparison<br>Mark Richards      | Developing Portlets with<br>Spring Portlet MVC<br>Mark Fisher | Extend the Customization<br>Possibilities of Your Java<br>App with Scripts<br>Ted Neward |
| 3:30 - 3:45 PM           | BREAK  |   |  |  |   |  |
| 3:45 - 5:15 PM           | Forgotten Algorithms<br>Jason Hunter                   | Pragmatic Extreme<br>Programming Part 2:<br>Architecture, Coding, and<br>Testing<br>Neal Ford       | Rolling Your Own Google<br>Maps, part I<br>Scott Davis | EJB 3 Part 2: Java<br>Persistence API (JPA)<br>Mark Richards           | Capistrano: Application<br>Deployment and More<br>David Bock  | Java6: Exploring Mustang<br>Ted Neward   |

# Northern Virginia Software Symposium 2006

Sheraton Reston - Reston, VA

November 03 - 05, 2006

(session listing as of 11/5/2006)

## **Introduction to TestNG, the next generation testing framework for developers by Andrew Glover**

No one will argue that JUnit has positively affected the quality of thousands of Java applications around the world. JUnit's simplicity and ease of use ushered in a whole new era of code quality; however, as many developers have found, its simplicity has also limited its use. TestNG was designed from the ground up to overcome some of JUnit's limitations; moreover, TestNG's features make it a great tool to complement your JUnit tests.

## **Keynote: How Hard Can It Be? by Andy Hunt**

Does creating software have to be so hard?

## **Pragmatic Learning by Andy Hunt**

How you learn new technology and acquire new skills is key to your personal success. But how do you learn how to learn? What tricks tips can you use to learn more faster, and retain more of what you learn?

## **Refactoring Your Wetware by Andy Hunt**

Software development happens in your head; not in an editor, IDE, or design tool. We're well educated on how to work with software and hardware, but what about wetware -- our own brains?

## **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.

## **ACEGI Authentication - The AJAX way by Brian Pontarelli**

Learn how to use ACEGI in an AJAX friendly way so that when you user's sessions expire, they can quickly log back into the application and continue where they left off. This will cover in detail the changes that were made to the ACEGI framework to support AJAX authentication, why ACEGI needed to be changed and how to implement this solution in a new or existing application that uses the ACEGI framework for authentication.

## **Java NIO by Brian Pontarelli**

The Java NIO packages that were added in JDK 1.4 and these packages allow Java applications to perform true non-blocking IO operations. This presentation will cover the basics of the standard IO packages, which date back to the beginning of Java, and some of the shortcomings they have. This will be followed by coverage of the newer NIO packages and how they address these issues.

## **Jini - Not just for your toaster anymore by Brian Pontarelli**

This presentation covers all the basics of the Jini platform, which has recently been transitioned from Sun to

Apache. This presentation will show how to construct a service based application using Jini as well as how the Jini network is structured and deployed. In addition, a demonstration of the cool Jini features such as dynamic discovery, recovery and provisioning will be given.

### **Applied REST by Brian Sletten**

REST sounds like such a simple thing. But, what is it really? How do you convince your boss to let you try it when she has been sold on the equation SOAP = SOA + P(rofit)? How do you go about building, deploying, publishing and orchestrating web services without the (Un)Holy Trinity of SOAP, WSDL and UDDI?

### **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.

### **Capistrano: Application Deployment and More by David Bock**

Capistrano (formerly Switchtower) is a tool originally written to help automate application deployment for Ruby on Rails. It does this well, but it has grown up into a tool capable of much, much more. It can be used for deploying Java applications, updating server configurations across an enterprise, administering networks, backing up files, and all sorts of other activities. Any activity you might do from the command line, you can now do simultaneously across large numbers of machines, with all machines succeeding (or rolling back in case of failure) together.

### **De-mystifying JSF by Ed Burns**

In this 90 minute session, Ed Burns will clear up the fog that sometimes surrounds people's understanding of this Web Application Development Framework. Ed is well suited to the task, having helped shape the design of JSF from its inception up to the present day. Upon leaving this session, the participant will know what JSF is good for, why it is good for these things, and how to be productive using it.

### **Enterprise Grade Ajax with JSF by Ed Burns**

This presentation will demonstrate how JSF and AJAX are perfect together. We begin with an insight into the design heritage of JSF, and how that heritage puts JSF in good stead to be productive in the AJAX world.

### **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.

### **Forgotten Algorithms by Jason Hunter**

There are many interesting and useful algorithms that people just don't remember or never learned. The Boyer-Moore string search algorithm is one prime example. The randomized skip list is another. Both solve common problems with wonderful flair and finesse -- and performance-wise they blow the pants off brute force solutions. This session covers these two algorithms plus several others. It's like your college algorithms course but with a practical bent and absolutely zero proofs. Extra bonus: The Google PageRank algorithm.

### **Greasemonkey: Ajax For One by Jason Hunter**

Greasemonkey is a Firefox extension with a hook to let you run your own JavaScript after each page loads. People have written hundreds of Greasemonkey scripts, each using Ajax techniques, to do everything from making a static page dynamic to changing a page's style to integrating comparison pricing on Amazon pages and giving Google search results easy keyboard shortcuts. Greasemonkey lets you take back control of the web browsing experience -- and teaches you Ajax while you're at it. In this talk we'll look at about a dozen useful Greasemonkey scripts. Each script provides a benefit in its own right, but like good engineers we'll dissect them to see what makes them tick.

### **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.

### **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.

### **Developing Portlets with Spring Portlet MVC by Mark Fisher**

Spring's Portlet MVC framework is one of the major new additions in Spring 2.0, bringing the proven benefits of the servlet-based Spring MVC framework to JSR-168 Portlet development.

### **Message Driven POJOs by Mark Fisher**

One of the most exciting new features of Spring 2.0 is its support for Message-Driven POJOs. It is now possible to receive JMS messages asynchronously and delegate the handling of those messages to simple objects. If your POJO has a return value, it will automatically be sent to a reply destination. Spring's messaging containers support configurable pooling of concurrent consumers and offer full integration with Spring's transaction management.

### **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).

### **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.

### **Effective Java Persistence Using Spring and iBATIS by Mark Richards**

Hibernate has evolved as the de facto standard for persistence in most Spring-based applications. However, many people are turning to iBATIS as an open source persistence alternative for Spring. iBATIS is a powerful open source persistence framework that is rapidly gaining in popularity, particularly within the Spring community. In this session you will learn why iBATIS is becoming so popular, how iBATIS differs from Hibernate and JPA (JSR-220), and how to use iBATIS as the persistence framework for spring-based applications. Through interactive coding examples I will demonstrate how to configure iBATIS within Spring and show how to define mapping files for simple SQL statements, complex SQL statements, and Stored Procedures. I will also discuss the various caching strategies available within iBATIS and the techniques, design strategies, and best practices for using iBATIS in small and large-scale spring-based applications.

### **Techniques in Architecture Agility by Mark Richards**

As companies continue to change the way they do business, so must the IT systems that support the business. Changes due to regulatory requirements, competitive advantage, mergers, acquisitions, and industry trends require flexible IT systems to meet the demands of the business. Software Architects must therefore make their architectures more agile to meet the flexible demands of today's business. In this session we will explore some of the challenges facing Software Architecture and discuss several concrete techniques for applying agility to both the architecture process and the technical architecture itself. Through real-world examples provided in this session you will learn how to apply various agile techniques to improve your architectures and overcome some of the challenges facing software architecture in today's

ever-changing market.

### **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.

### **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.

### **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.

### **Testing with Selenium by Neal Ford**

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

### **Agile Enterprise Architecture: The role of the architect by Pete Behrens**

Enterprise Architectures (EA) and Service Oriented Architectures (SOA) are popping up everywhere to better align IT to the business. In this session, we will step behind the technology to help you understand how your organization architecture (yes, how your organization is constructed) is the primary driver in developing an effective Enterprise Architecture, why an agile perspective is required, and an exploration of your role as enterprise architect to achieve the desired results.

### **Agile Estimating, Planning and Tracking by Pete Behrens**

Every project requires estimation, planning and progress tracking. While agile processes are often described as and implemented lacking in these project management techniques, a well-executed agile process actually provides more accurate results with less time investment than traditional methods. This session will demonstrate these agile project management techniques to manage 6-12 month projects.

### **Agile Requirements with User Stories by Pete Behrens**

User Stories, a key practice from Extreme Programming, provide a right-sized solution to more efficiently identify, track and implement product requirements. Learn how identify, write and decompose "good" user stories that drive agile behavior and business value.

### **Agile Tooling: Team to Enterprise by Pete Behrens**

"YAGNI (You Ain't Gonna Need It)" and "Doing the simplest thing possible" are mantras of agile development. A white board, sticky notes, and flip chart paper are by far the best tools for individual teams. However, when coordinating work across 10 - 50 teams across 12 time zones, more tooling is required. Learn how agile enterprises are leveraging tooling to manage their portfolios, projects and products.

### **Domain Driven Design with AOP and DI by Ramnivas Laddad**

Domain Driven Design (DDD) suggests dealing with complex software system using a domain model and preserving the model in implementation. Since domain model entities have rich behavior, so should their software implementation artifacts. A direct mapping between domain model and software artifacts create simple-to-understand, inexpensive-to-implement, and easy-to-evolve systems. While the idea behind DDD isn't new and the value is easily understood, many implementations do not adhere to its principles. This disconnection may be due to many obstacles in implementing it. Combining Dependency Injection (DI) with a full-fledged aspect-oriented programming (AOP) system such as AspectJ help overcome many obstacles.

### **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.

### **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.

### **Easing into Agile by Scott Davis**

How do you get started with an Agile development methodology? Everyone has been talking about eXtreme Programming for years, but how do you get it introduced to your team? Many times, you're not simply transitioning from from one methodology to another -- you're introducing a methodology for the first time. Adding structure to a previously unstructured endeavor. Adding a touch of discipline where programmers once roamed free.

### **Groovy: Greasing the Wheels of Java by Scott Davis**

This is the year of the dynamic scripting language. Ruby (and Rails) has won the hearts and minds of many independent software developers. JavaScript is experiencing a renaissance thanks to the wild success of AJAX and websites like Google Maps. And Groovy (JSR-241) brings the same level of excitement and "scripting goodness" to the Java platform.

### **Holistic Testing by Scott Davis**

Mark Twain once said, "Everyone talks about the weather, but nobody does anything about it." Do you feel the same way about Unit Testing? Are you actively testing your code, or are you just thinking about testing your code... some day... once you get some more free time...

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

In this talk, we'll survey the web services exposed by leading websites (Google, Yahoo, Amazon, eBay) and discuss how they are driving the AJAX revolution. You'll see examples of RESTful, SOAP, and JSON web services, as well as the strengths and weaknesses of each.

### **Rolling Your Own Google Maps, part I by Scott Davis**

The release of Google Maps was a "Wizard of Oz / Technicolor" moment for web developers everywhere. It didn't just change the way we look at mapping sites; it forever changed the way we look at all web sites. It put AJAX on the map, both figuratively and literally.

### **Effective Hibernate by Scott Leberknight**

Hibernate seems simple on the surface yet when you go beyond very simple use cases it can become much more complex. Intended for beginner to intermediate-level Hibernate developers, come see how to put Hibernate to effective use on your projects.

### **Spring/Hibernate Integration Basics by Scott Leberknight**

Hibernate is a very popular Java transparent persistence framework, but you often need to create additional infrastructure to manage sessions, transactions, and lazy-loading in a clean and elegant manner. See how Spring can help.

### **Refactoring, Hands On! by Steve Metsker**

Learn the basics of refactoring #- improving the internal health of your code #- and start honing your skills immediately in this hands-on session.

### **Ajax Architecture by Stuart Halloway**

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.

### **JRuby by Stuart Halloway**

JRuby is not one, but two great technologies: the Ruby language, and the Java Virtual Machine and libraries.

In this talk you will learn the basics of programming JRuby, and how to integrate JRuby code into existing Java projects.

### **JavaScript for Ajax Programmers by Stuart Halloway**

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.

### **Prototype: Ajax and JavaScript++ by Stuart Halloway**

Learn to simplify Ajax development with Prototype through a series of real-world examples. Along the way, learn to code in Prototype's modern JavaScript style, taking advantage of Prototype's extensions to JavaScript's object model

### **Spring Dependency Injection by Stuart Halloway**

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 Stuart Halloway**

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.

### **What's New in Rails 1.1? by Stuart Halloway**

Rails 1.1 is a major new release, with features in every area. We'll cover the most powerful new features, showing the strengths of Rails and of Ruby itself. You'll see not only how to use the new features, but implementation details that you can use to add your own new features.

### **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.

### **Essential db4o by Ted Neward**

Tired of writing object-table mappings? For years, Java developers have wrestled with the problems of storing objects into relational format and retrieving them back again; for all that Hibernate and JDO and other O/R tools make it easier (though not easy) to do, isn't there another way? In this presentation, we'll explore an alternative approach, real object persistence, using the db4o toolkit ([www.db4objects.com](http://www.db4objects.com)).

### **Java Annotations: From Definition to Consumption by Ted Neward**

Want to get the soup-to-nuts story on Java annotations? In this presentation, we'll first talk about what annotations provide to the Java language. After setting ourselves a conceptual basis to operate from, we'll look at the language definition for Java annotations, from how to use them to how to define them. Finally, we'll take a look at the other side of annotations, consuming them at source-level (using "apt", the annotation processing tool), class-level (using a bytecode toolkit such as BCEL), and at runtime (using enhancements to the Reflection API made in Java5).

### **Java6: Exploring Mustang by Ted Neward**

Mustang, the forthcoming Java6 release, is just around the corner, and even if you're not looking to adopt the new platform right away, it's important to know what's there so you can start to plan for it. In this presentation, we'll go over the major new features of the Java6 platform, including the new integrated XML services capabilities (JAX-WS and JAXB), dynamic/scripting language support (javax.script), new JVM "attach" capabilities, new annotations supported by the javac compiler, and more.

### **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.

### **The Busy Java Developer's Guide to ClassLoaders by Ted Neward**

If you've ever gotten a `ClassCastException` and just knew the runtime was wrong about it, or found yourself copying `.jar` files all over your production server just to get your code to run, then you probably find the Java `ClassLoader` mechanism to be deep, dark, mysterious, and incomprehensible. Take a deep breath, and relax--`ClassLoaders` aren't as bad as they seem at first, once you understand a few basic rules regarding their operation, and have a bit more tools in your belt to diagnose `ClassLoader` problems. And once you've got that, and hear about `ClassLoaders`' ability to run multiple versions of the same code at the same time, and to provide isolation barriers inside your application, or even compile code on the fly from source form, you might just find that you like `ClassLoaders` after all... maybe.

### **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.

### **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.

### **Spring into Unit Testing by Venkat Subramaniam**

Test Driving Development is a valuable technique that has several benefits. However, unit testing is hard when your code has dependencies-this often becomes a quick excuse to give up on TDD. This example driven Zero Powerpoint (ZePo) presentation will help you spring into unit test your Spring applications. We will look at techniques to realize good unit testing, and focus on effective use of mock objects and frameworks to help you toss mock objects for your spring application.

### **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.