

Great Lakes Software Symposium

Westin Chicago Northwest

November 21 - 23, 2008

<http://www.nofluffjuststuff.com/conference/chicago/2008/11/index.html>

(event schedule as of November 22, 2008)

Fri, Nov. 21, 2008						
	Ballroom 3 & 4	Ballroom 1 & 2	Chambers	Marlborough	Radcliffe	Leighton
12:00 - 1:00 PM	REGISTRATION					
1:00 - 1:15 PM	WELCOME					
1:15 - 2:45 PM	Java Persistence: Approaching the Silver Bullet Mark Richards	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	Test Driven Design Neal Ford	Designing for Ajax, part 1 Nathaniel Schutta	10 Tips for Getting Your Project Back on Track Jared Richardson	Java 7 Preview Alex Miller
2:45 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Enterprise Messaging Using JMS (Part 1) Mark Richards	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind Scott Davis	Techniques 2008 Jared Richardson	Designing for Ajax, part 2 Nathaniel Schutta	The Productive Programmer: Mechanics Neal Ford	Design Patterns Reconsidered Alex Miller
4:45 - 5:00 PM	BREAK					
5:00 - 6:30 PM	Enterprise Messaging With JMS (Part 2) Mark Richards	Rapid Web Development with Grails and Ajax Scott Davis	Evolutionary SOA Neal Ford	Improving Code Quality Nathaniel Schutta	Credit Card Software Development: Recognizing and Repaying Technical Debt Jared Richardson	Java Concurrency Idioms Alex Miller
6:30 - 7:15 PM	DINNER					
7:15 - 8:00 PM	Keynote: by Neal Ford					

Sat, Nov. 22, 2008						
	Ballroom 3 & 4	Ballroom 1 & 2	Chambers	Marlborough	Radcliffe	Leighton
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe	Transaction Design Patterns Mark Richards	YSlow: Building Your Website for Speed Scott Davis	Distributed Teams: Remote Agility Jared Richardson	Project Smells Nathaniel Schutta	Java Collections API Alex Miller
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	Spring 2.5 - Spring without XML Ken Sipe	Refactoring into Testability Scott Davis	"Design Patterns" in Dynamic Languages Neal Ford	Build Teams, Not Products Jared Richardson	How to Fail with 100% Code Coverage Stuart Halloway	Exploring Terracotta Alex Miller
12:30 - 1:30 PM	LUNCH					
1:30 - 3:00 PM	Spring and JMS: Message-Driven POJOs Mark Richards	A Thorough Introduction To Groovy Jeff Brown	JavaServer Faces: A Whirlwind Tour David Geary	Architecture and Agility Are Not Mutually Exclusive David Hussman	JavaScript: the Good, the Bad, and the Ugly Nathaniel Schutta	Regular Expressions in Java Neal Ford
3:00 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Architecture and Scaling Ken Sipe	Agile Test Driven Development With Groovy Jeff Brown	Facelets David Geary	Agile Management & Managing Agility David Hussman	Refactoring JavaScript Stuart Halloway	Web 2.0 Punchlist: Making Your Web Applications Suck Less Neal Ford
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION					

Sun, Nov. 23, 2008						
	Ballroom 3 & 4	Ballroom 1 & 2	Chambers	Marlborough	Radcliffe	Leighton
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	7 Habits of Highly Effective Developers Ken Sipe	Groovin' builds Gant get any easier Andrew Glover	Introduction to Hibernate Scott Leberknight	Give it a REST Brian Sletten	Using Ajax4jsf David Geary	Coaching and Leading Agile Projects David Hussman
10:30 - 11:00 AM	MORNING BREAK					
11:00 - 12:30 PM	Java Memory, Performance and the Garbage Collector Ken Sipe	Tactical Continuous Integration with Hudson Andrew Glover	Real World Hibernate Tips Scott Leberknight	RESTlet for the Weary Brian Sletten	Intro to Seam David Geary	Automating Customer Acceptance David Hussman
12:30 - 1:15 PM	LUNCH					
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION					
2:15 - 3:45 PM	Iteration 0 Ken Sipe	Java.next #1: Common Ground Stuart Halloway	Google Your Domain Objects With Hibernate Search Scott Leberknight	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	Easy BDD with Groovy Andrew Glover	Introduction to NetKernel : Software for the 21st Century Brian Sletten
3:45 - 4:00 PM	BREAK					
4:00 - 5:30 PM	Hacking - The Dark Arts Ken Sipe	Java.next #3: Dispatch Stuart Halloway	Groovier Spring (More Flexible Applications With Spring and Groovy) Scott Leberknight	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	Less is Always More Andrew Glover	The Semantic Web : The Future, Now Brian Sletten

Great Lakes Software Symposium

Westin Chicago Northwest

November 21 - 23, 2008

<http://www.nofluffjuststuff.com/conference/chicago/2008/11/index.html>
(event schedule as of November 22, 2008)

Java 7 Preview by Alex Miller

Possible library and language changes for Java 7 have been in discussion since early 2007 and many things have been proposed. This talk will focus on those items that are most likely to be included and some of the more controversial language changes that might be included in a future release. For a comprehensive list of almost everything being considered, see <http://tech.puredanger.com/java7>.

Design Patterns Reconsidered by Alex Miller

The Design Patterns book launched a revolution in object-oriented design and provided a vocabulary for OO developers to communicate their ideas. However, in some cases, patterns used blindly can lead to awkward, confusing, or hard to maintain code. It is time for some common patterns used in Java to be reconsidered so that we can derive the benefits from patterns while minimizing their concerns. This talk will re-evaluate key patterns like Singleton, Template Method, Visitor, and Proxy. These patterns have downsides and in some cases, do more harm than good. Examples of each pattern will be given in Java and examined for clarity, testability, and flexibility. Important problems will be discussed and examples of alternate solutions will be given.

Java Concurrency Idioms by Alex Miller

This presentation will look at the many new additions in Java 5 and 6 for concurrent programming such as Atomics, Locks, synchronizers, and concurrent collections. In particular, we will be looking at common concurrency idioms around locking and access to shared state, thread coordination, thread pooling, and work execution. Each of these topics will be presented with code examples demonstrating common idioms and the usage of these new concurrency primitives.

Java Collections API by Alex Miller

Did you know that Java 5 and 6 added 8 new interfaces and 16 new collection implementations to the JDK, more than doubling the size of the collection API? Collections 201 gives you an update on all of the interfaces, implementations, and utilities and gives you guidance on picking the perfect collection. In particular, Java 5 introduced a new major collection type Queue and a whole new `java.util.concurrent` package with data structures optimized for concurrent use.

Exploring Terracotta by Alex Miller

Terracotta is an open-source Java clustering technology. It creates a virtual, durable Java heap that is shared across a cluster of Java Virtual Machines. This is done by dynamically instrumenting bytecode at load time to intercept calls to read and write fields, and also to enter and exit monitor locks. Information about these calls is then transmitted to the Terracotta Server (which can also be clustered) and out to other nodes in the cluster as needed. The advantage of this approach is that many Java programs can be clustered without code changes by providing just external Terracotta configuration. Many performance optimizations are performed to minimize communication and locking costs. Terracotta is commonly used for session sharing in web applications, distributed caching, and distributed workflow processing. This presentation will give an overview of the Terracotta technology, how it's implemented, and common use cases that can benefit from the technology. We will look at some code and cluster some Java applications during the presentation.

Groovin' builds Gant get any easier by Andrew Glover

There's no question that Ant is the de facto standard for building Java applications; however, even its creator has acknowledged an inherent limitation with Ant's expressiveness due to its reliance on XML. Recently, the popularity of Ruby and the Rails framework has brought to focus Ruby's de facto build platform: Rake. Rake's expressiveness comes from its reliance on Ruby itself to define a DSL for software assembly. While Rake's ultimate focus is Ruby, there are a number of interesting projects that utilize expressive DSLs for building Java including Gant, which uses Groovy as a DSL format and builds upon Ant's existing cornucopia of tasks.

Tactical Continuous Integration with Hudson by Andrew Glover

This session will walk attendees through a series of iterations on a fictional Java project where an automated build system is created that facilitates compilation, testing, inspection, and deployment. This build system is then plugged into the Hudson CI server and as features are coded using Agile techniques like developer testing, attendees will ultimately see firsthand how a Continuous Integration process reduces risk and improves software quality.

Easy BDD with Groovy by Andrew Glover

Behavior-driven development, or BDD, has attracted a lot of attention via RSpec in the Ruby community, but BDD's roots stem from JBehave, a Java based framework modeled off of the xUnit paradigm. But JBehave isn't the only framework available for Java developers-- with the advent of Groovy, new options are available for embracing BDD in the spirit of RSpec's innovative behavior based DSL.

Less is Always More by Andrew Glover

The spectacular success and market penetration of the iPod, Agile software development principles, Twitter, and even Google are just a few examples of minimalism in action.

Give it a REST by Brian Sletten

As developers, we sometimes get to make choices about the technologies we use, sometimes not. We base these decisions on personal experiences, recommendations from others and a general sense of where the industry is going. Web Services have been all the rage for several years now. We have been told time and again that we should be building systems around them; as an industry, we've never been more confused. Perhaps it is time to Give it a REST.

RESTlet for the Weary by Brian Sletten

If you have started to take a look at REST as way of exposing web services or managing information spaces, you may be frustrated by the support offered by legacy containers. There is no direct support for REST concepts in the J2EE specs (yet). XML-based configurations are so 1990's. Come learn about Restlets, a little API that has caught the attention of many in the RESTafarian community. **Prerequisite:** *Give it a REST (unless you are very comfortable with REST)*

Introduction to NetKernel : Software for the 21st Century by Brian Sletten

Imagine the simplicity of REST married to the power of Unix pipes with the benefits of a loosely-coupled, logically-layered architecture. If that is hard to imagine, it may be because the architectures available to you today are convoluted accretions of mismatched technologies, languages, abstractions and data models. NetKernel is a disruptive technology that changes the game. It has been quietly gaining mind share in the past several years; people who are exposed to it don't want to go back to the tired and blue conventions of J2EE and .NET. Not only does it make building the kinds of systems you are building today easier, it does it more efficiently, with less code and a far more scalable runway to allow you to take advantage of the emerging multi-core, multi-CPU hardware that is coming our way. Come see how this open source / commercial product can change the way you think about building software.

The Semantic Web : The Future, Now by Brian Sletten

Just as the world is feeling comfortable with the Web, Tim Berners-Lee et al inform us that what we have seen so far is just the beginning. His original plans at CERN were larger and grander. The Semantic Web is a vision of machine-processable documents and metadata to improve search, knowledge discovery and data integration and management. The only problem is that there is no such thing. There is no Semantic Web, just the Web we have that is increasingly semantics-enabled. Forget the hype. Come learn how the technologies of this vision are being used today on the Web and in the Enterprise by more people than you might think.

JavaServer Faces: A Whirlwind Tour by David Geary

In April 2005, annual growth rates for jobs in JavaServer Faces, Struts, and Ruby on Rails were all at about 0%. Today, Struts' growth rate still hovers around 0%, but JSF and Rails have taken off. At the end of 2007, both JSF and Rails were growing at a rate of between 400-500% annually (according to indeed.com). JSF has passed the adoption tipping point, and is now the Java-based framework of choice, as is evidenced by its ecosystem. From vendors such as MyEclipse and RedHat to open source projects such as Seam, Facelets, and Ajax4JSF, JSF is where the action is. Come see why JSF is so popular. In this code- and demo-intensive session, I'll show you the fundamentals of JSF. **Prerequisite:** *Some knowledge*

of Java-based web applications, such as Struts, is a plus, but is not required. If you have a significant experience with JSF, you probably already know most of what's covered in this session.

Facelets by David Geary

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet. It's popularity is well deserved, and in fact, much of what is in Facelets today will make its way into the JSF 2.0 spec due out in 2008. So not only can you come to this session and see some really cool demos that you can put to use in the real world, but you'll also be learning JSF 2.0 before it's even been defined! How's that for a ROI? **Prerequisite:** *Some knowledge of JSF is essential. If you're familiar with a templating framework, such as Velocity or Tiles, that's a plus, but not required.*

Using Ajax4jsf by David Geary

Ajax4jsf makes it very easy to add Ajax to your JSF applications. Come to this presentation to see how.

Intro to Seam by David Geary

Have you ever stopped to think that you need to learn two frameworks to develop a non-trivial, database-backed, web application? Struts and iBatis; JSF and Hibernate; Tapestry and EJB3.0.

Filthy Rich Clients with the Google Web Toolkit, Part I by David Geary

The Google Web Toolkit (GWT) is truly a revolutionary framework that lets you develop Ajaxified web applications without knowing anything about Ajax or JavaScript. But the GWT goes way beyond basic Ajax by letting you implement desktop-like applications that run in the ubiquitous browser.

Filthy Rich Clients with the Google Web Toolkit, Part II by David Geary

In the second part of this talk, you will learn how to extend the GWT by implementing custom widgets, including a scrolling viewport and a drag and drop framework. After discussing custom widgets, you will see how to integrate database access into your GWT applications, and how to deploy your GWT applications to external servers.

Architecture and Agility Are Not Mutually Exclusive by David Hussman

Being agile does not mean living life one iteration at a time. Agile projects without a long view can run into the common design problems of the past. Planning iteration by iteration is often foolish and feeds the myth that agile projects do not think beyond a few weeks. Successful agile projects plan within iterations and across iterations. The later planning is called release planning and it is the forum where agility first engages architecture and other cross cutting concerns.

Agile Management & Managing Agility by David Hussman

Management and agility are not mutually exclusive. Many managers are already working in an agile manner as a means to improve, produce, or simply survive. Other managers hear about projects using agile methods and struggle to find a place in the project community. This session provides a new way to think about managing projects. Some managers will find that their existing practices and skills are supported and enhanced by the forums and metrics provided within an agile project while others will be challenged by some of the principles and practices.

Coaching and Leading Agile Projects by David Hussman

Successfully coaching agile communities involves using a wide variety of skills. Coaches help guide coding and design, collaboration and communication, the writing and telling of user stories and much more. The coach needs to continuously show and teach the varied interactions that connect and support the entire project community. This session will explore and teach coaching skills. The session will reference a wide variety of agile coaching as well as drawing from cross disciplinary techniques like those used by music producers to help foster creativity while helping to ensure products are delivered and challenges confronted.

Automating Customer Acceptance by David Hussman

Why should the value of test driven development (TDD) stay stuck in the realm of coding? The ideas behind TDD are now being successfully applied to the automation of business value. While this has been going on for some time within the agile community, it is not starting to spread to main stream development. There are more tools coming available everyday which allow developers, testers, and customers (or product owners) to work together to automate acceptance tests. This process helps clarify the needs of the end user

before development begins and removes more of the wasteful work based on incorrect assumptions from vague requirements.

10 Tips for Getting Your Project Back on Track by Jared Richardson

Software projects fail over and over for many of the same reasons. We'll look at some of the more avoidable problems and some solid ways to fix them, or avoid them in the first place.

Techniques 2008 by Jared Richardson

There are a number of great techniques you can use across technologies and projects. Come hear some of my favorites and contribute a few of your own. We'll discuss topics from DRY to creating a zone defense for your product.

Credit Card Software Development: Recognizing and Repaying Technical Debt by Jared Richardson

Technical debt has long been recognized in technical circles for years, but convincing your manager to budget time to repay "technical debt" has always been problematic. Let's couch the term technical debt concept in language more familiar to our managers: credit card debt.

Distributed Teams: Remote Agility by Jared Richardson

How do you keep a team scattered across time zones in sync?

Build Teams, Not Products by Jared Richardson

A great team builds great software, but how do you build a great team?

A Thorough Introduction To Groovy by Jeff Brown

Groovy is an agile dynamic language for the Java platform. The language and its libraries bring many things to the table to ease the process of building applications for the Java platform. This session provides a detailed run through Groovy with lots of code samples to drive home the power of the language.

Agile Test Driven Development With Groovy by Jeff Brown

Dynamic languages bring a lot of interesting elements to the table for teams interested in doing Test Driven Development (TDD). Groovy lends itself very well to TDD and this session demonstrates many features of the language and its libraries that help teams build more testable systems and build better tests.

Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence by Ken Sipe

Well the standards created EntityBeans.... yea. and the community created Hibernate. Fortunately the standards body learned some lessons and created JPA. JPA requires a vendor implementation and none make a better choice than Hibernate. Combined with Spring this trio is a powerhouse when it comes to developer productivity on applications requiring persistence.

Spring 2.5 - Spring without XML by Ken Sipe

Spring 2.5 is brand spanking new, with a number of fantastic features. With growth of large and complex Spring applications which struggle with xml manageability and with the added pressure of Guice and SEAM there is a push for less XML, with solution leaning towards annotations. Spring 2.5 adds to the toolset provided in Spring 2.0 to provide a development environment where XML is greatly reduced... or eliminated if you so choose.

Architecture and Scaling by Ken Sipe

Scale... what is scale... how do you applications which are scalable. How do you know if the application scales?

7 Habits of Highly Effective Developers by Ken Sipe

Thoughts lead to words, words lead to action, actions lead to habits. In this session we'll sharpen the development saw in the process of understanding what makes a hyper-productive programmer. The focus will consist of developer habits and development processes.

Java Memory, Performance and the Garbage Collector by Ken Sipe

You are using Java, whew!!! No need to worry about memory, the garbage collector will handle that. Those who have had a memory issue in Java are not so naive any more. Often memory utilization and heap sizes

are an after thought and are not recognized until the application is in production, often caused by application uptime, production request volume or production sets of data. When the OutOfMemory Error occurs, often the science of development seems to brake down and knobs are turned. First the (-mx) maximum heap space gets adjusted... More is better right. The next OutOfMemory, heads start scratching, code reviews start in earnest, and Google gets several new hits. Did you know that it is possible to get an OutOfMemory error without running out of heap space?

Iteration 0 by Ken Sipe

The success of an Agile / SCRUM project is a successful start. The first interaction is often referred to as iteration 0. Other iterations have a set of stories with clear acceptance certain which establishes the velocity of the team and its effort. What then is accomplished in iteration 0? How do we get an Agile process started.

Hacking - The Dark Arts by Ken Sipe

A live Hacking demonstration exposing the tools and techniques used by Hackers.

Java Persistence: Approaching the Silver Bullet by Mark Richards

Java Persistence has come along way since the days of straight JDBC coding and custom framework development. We have at our disposal several outstanding open source frameworks such as Hibernate, Toplink, iBatis, and OpenJPA (just to name a few), and we now have a promising and emerging standards-based solution called Java Persistence API (JPA). However, all to often we find in the Java persistence space that it is a world of one-size-does-not-fit-all. We continually struggle with traditional ORM solutions like Hibernate when it comes to reporting queries, complex queries, complex relationships, and stored procedures, and we also struggle with managing the enormous amount of SQL required for solutions such as iBATIS or JDBC-based frameworks. In this coding-intensive session we will take a detailed look at identifying and overcoming the challenges we face when using frameworks such as Hibernate, iBATIS, and JPA, and how to combine the various persistence frameworks to create an effective Java persistence solution that approaches (but of course does not reach) the silver bullet.

Enterprise Messaging Using JMS (Part 1) by Mark Richards

The chances are good that at some point in your career you will need to use messaging to pass information between applications, subsystems, or external systems, particularly with service-oriented architecture on the rise. The Java Messaging Service (JMS) allows Java applications to implement messaging using a standard API, thereby removing the dependency on any particular messaging provider. In Part 1 of this session we will take a look at some of the basics of messaging, including sending and receiving messages, message types, and request/reply messaging. I will begin the session by going over the basics of messaging and the JMS API. Then, through interactive coding using OpenJMS I will demonstrate how to connect to JMS providers, send messages, receive messages, and use message properties. Please note that this is a two part session.

Enterprise Messaging With JMS (Part 2) by Mark Richards

In Part 1 of the JMS session I covered messaging models, messaging basics, the JMS API, and point-to-point messaging. In this interactive code-intensive session I will cover some additional JMS topics such as browsing queues, load balancing, publishing and subscribing to messages within the pub/sub model, durable and non-durable subscribers, message selectors, and message filtering. I will also discuss and demonstrate message prioritization, persistent and non-persistent messages, and finally message expiration (expiry). Note that this is Part 2 of a two-part JMS session. **Prerequisite:** *Enterprise Messaging With JMS (Part 1) or some knowledge of JMS*

Transaction Design Patterns by Mark Richards

Most web-based applications rely solely on the database to manage transactions, thereby freeing the developer from having to worry about transaction management. While this works in some circumstances, there are times when the use of transactions is vital to the integrity and operations of an application and its corresponding data. In this session I will demonstrate through real-world coding examples why transactions are such a critical part of the application development process. I will review the basics of both programmatic and declarative transactions, then introduce three transaction design patterns and explain when they should be applied, how to use them, and what problems they solve. By the end of this session you will see that by using transaction design patterns you can build an effective transaction management strategy for your application with very little effort.

Spring and JMS: Message-Driven POJOs by Mark Richards

The Java Message Service (JMS) provides an standard messaging API that allows you to send and receive messages using a variety of messaging providers (including Java EE application servers). The Spring Framework takes this abstraction one step further by providing an robust JMS messaging framework that greatly simplifies message processing. In this session we will see how to use the JMS Messaging Framework provided in Spring 2.5. I will start by describing Spring's overall messaging architecture and how to configure the various beans needed for messaging. Then, through interactive coding I will discuss and demonstrate Spring's JMS Template. which is used for sending messages and receiving messages synchronously. I will then discuss and demonstrate Message Driven POJOs, which are Spring's answer for asynchronous message listeners. After attending this session you will have all the necessary knowledge and code examples to use JMS in your Spring applications. **Prerequisite:** *Knowledge of JMS and Spring*

Designing for Ajax, part 1 by Nathaniel Schutta

So you've convinced the boss that your new web application just has to have Ajax...but now what? With dozens of libraries making even the most blinkish of interactions trivial, how do you decided where to sprinkle the magic Ajax dust? This talk will give a plain old boring "web 1.0" an Ajax facelift with a focus on improving the user experience providing you with a game plan for introducing Ajax to your world.

Designing for Ajax, part 2 by Nathaniel Schutta

We'll pick up where Part 1 left off working in even more advanced approaches such as offline support with Google Gears.

Improving Code Quality by Nathaniel Schutta

It seems that software follows the second law of thermodynamics - in other words, code tends towards disorder. Of course it doesn't have to be that way, and we have a number of tools and techniques that we can apply to keep our code in tip top shape. This talk will discuss ten things you can do to fight back!

Project Smells by Nathaniel Schutta

We all know that code can have a certain...odor but frankly so can projects. Everyone has their favorite horror story or tale of a death march. In this talk, we'll discuss common project smells and what you as a developer can do to maintain your sanity - and your hair line!

JavaScript: the Good, the Bad, and the Ugly by Nathaniel Schutta

Thanks to Ajax, JavaScript is cool again and developers are taking a second look at this much maligned language. This session will give you an overview of this misunderstood language as well as opening your eyes to some of the excellent tools available to ease the pain of developing in this dynamic language.

Test Driven Design by Neal Ford

Most developers think that "TDD" stands for Test-driven Development. But it really should stand for "Test-driven Design". Rigorously using TDD makes your code much better in multiple ways.

The Productive Programmer: Mechanics by Neal Ford

Developers from the 1980s would be shocked at how inefficiently developers use their computers because of the advent of graphical operating systems. This talk describes how to reclaim productivity afforded by intelligent use of command lines and other ways of accelerating your interaction with the computer and bending computers to do your bidding. Stop working so hard for your computer!

Evolutionary SOA by Neal Ford

This session demonstrates that "Agility" and "SOA" complement each other quite well. Just because SOA is buzz-word compliant doesn't mean that you should throw good practices out the window. This session demonstrates how you can apply the principles of agility to building highly complex distributed enterprises.

Keynote: Ancient Philosophers & Blowhard Jamborees by Neal Ford

It turns out that ancient philosophers knew a lot about software -- did you know that Plato defined object-oriented programming? This keynote applies old lessons to new problems and old problems to new lessons. It describes why SOA is so hard, and why people in your company make bone-headed decisions. What other keynote includes Rube Goldberg, Aristotle, Dave Thomas, and Chindia?

"Design Patterns" in Dynamic Languages by Neal Ford

The Gang of Four book should have been entitled "Palliatives for Statically Typed Languages", because the recipes it provides are cumbersome solutions to the problems it poses. Using powerful languages makes the solutions in the GoF book look hopelessly complicated. This session shows how to solve the same problems concisely, elegantly, and with far fewer lines of code using the facilities of dynamic languages.

Regular Expressions in Java by Neal Ford

Regular expressions should be an integral part of every developer's toolbox, but most don't realize what an important topic 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.

Web 2.0 Punchlist: Making Your Web Applications Suck Less by Neal Ford

Provides a punchlist to ensure your shiny new web application is up to spec.

Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy by Scott Davis

There are wild-eyed radicals out there telling you that Java is dead, statically-typed languages are passe, and your skills are hopelessly out-of-date. Those extremists are the same ones who don't bat an eye at throwing out years of experience to learn a new language from scratch, pushing aside a familiar IDE for a new one, and deploying to a whole new set of production servers with little regard to legacy integration. While this "burn the boats" approach to software development might sound exciting to some folks, it's giving your manager the cold shakes right now. What if I told you that there was a way that you could integrate seamlessly with your legacy Java code, continue to use your trusty IDE and stable production servers, and yet take advantage of many of the exciting new dynamic language features that those fanatics keep prattling on about? You'd probably say, "Groovy!" I would, too...

Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind by Scott Davis

This talk focuses on the ways that Groovy can turn a traditional Java developer's world-view upside down. We'll start by talking about how you can thumb your nose at The Man by leaving out many of the main syntactic hallmarks of Java: semicolons, parentheses, return statements, type declarations (aka Duck-typing), and the ever-present try/catch block. Then we'll look at features like operator overloading and method pointers that Groovy welcomes back into the language with open arms.

Rapid Web Development with Grails and Ajax by Scott Davis

Grails is a Java- and Groovy-based web framework that is built for speed. First-time developers are amazed at how quickly you can get a page-centric MVC web site up and running thanks to the scaffolding and convention over configuration that Grails provides. Advanced web developers are often pleasantly surprised at how easy it is to break out of that coarse-grained navigation model using the native Ajax support baked into the framework.

YSlow: Building Your Website for Speed by Scott Davis

How optimized is your website? YSlow, a FireFox/FireBug plugin, doesn't pull any punches. It gives any website an A, B, C, D, or F rating based on 14 individual analysis points. You'll be amazed (or depressed) at what YSlow thinks of your site. In this talk, we'll walk through these points step by step, learning what Yahoo! (the creator of this utility) does to keep its web properties running as quickly as possible.

Refactoring into Testability by Scott Davis

"This code is simply too hard to unit test." That is a common refrain when dealing with software that hasn't been expressly written to be testable. In this section we look at "untestable code" and explore various ways to make it more testable. What you'll come to realize is that "untestable code" is really another way of saying "poorly architected code." We'll demonstrate simple, common-sense strategies that solve both problems.

Introduction to Hibernate by Scott Leberknight

This session introduces the Hibernate Object/Relational Mapping (ORM) framework, showing the basics of persisting Java objects to relational databases. No prior knowledge of Hibernate or ORM is assumed.

Real World Hibernate Tips by Scott Leberknight

Hibernate is a very powerful object/relational mapping framework. With the vast amount of power also comes the responsibility to choose which features of Hibernate to use and how to use them, as well as things to avoid. We'll look at some real world Hibernate tips and tricks in this session.

Google Your Domain Objects With Hibernate Search by Scott Leberknight

Hibernate is one of the pre-eminent object/relational mapping technologies, but the Hibernate Search project adds full-text search capabilities to an already extremely capable tool to allow you to Google your domain objects.

Groovier Spring (More Flexible Applications With Spring and Groovy) by Scott Leberknight

Spring provides a solid foundation for web and enterprise applications. Its support for dynamic languages like Groovy adds interesting capabilities that can make your application architecture more flexible and dynamic.

How to Fail with 100% Code Coverage by Stuart Halloway

Over the last few years, we have taken dozens of projects to 100% coverage, and there are still plenty of things that can go wrong. We will look at examples the various problems, and show how to prevent them from infecting your project.

Refactoring JavaScript by Stuart Halloway

The rise of Ajax and Rich Web Applications, plus the success of dynamic languages, has caused people to revisit the JavaScript language. Now that we take JavaScript seriously as a language, it is time to get serious about the quality of JavaScript code, through refactoring. In this talk, we will approach refactoring JavaScript in three phases: Test first, then refactor. Bring JavaScript code under test, so that you can refactor with confidence. Refactoring 101. Explore some important refactorings: composed method, extract method, introduce named parameter, and extract object Common problems. Work through three problems endemic to legacy JavaScript code: making JavaScript unobtrusive, refactoring to prototype-based inheritance, and refactoring to functional style.

Java.next #1: Common Ground by Stuart Halloway

In this talk, we will explore and compare four of the most interesting new JVM languages: Clojure, Groovy, JRuby, and Scala. Each of these languages aims to greatly simplify writing code for the JVM, and all of them succeed in this mission. However, these languages have very different design goals. We will explore these differences, and help you decide when and where these languages might fit into your development toolkit. For more information see <http://blog.thinkrelevance.com/2008/8/4/java-next-common-ground>.

Java.next #3: Dispatch by Stuart Halloway

Dispatch takes many forms. Single dispatch, switch statements, pattern matching, and multiple dispatch all meet similar needs: Selecting runtime behavior in response to varying runtime conditions. Flexible dispatch is a key element of Java.next. All of the Java.next languages support dispatch strategies that are far more flexible than Java's single dispatch. In this talk (Part 3 of the Java.next series), I will explore how the Java.next languages (Clojure, Groovy, JRuby, and Scala) support dispatch.