

# Great Lakes Software Symposium 2005

Sheraton Chicago Northwest - Arlington Heights, IL

September 16 - 18, 2005

(session listing as of 9/14/2005)

The No Fluff Just Stuff Software Symposium 2005 tour is pleased to announce the Great Lakes Software Symposium coming to Arlington Heights on September 16 - 18, 2005. You will have the opportunity to attend the best technically focused Java / Open Source event anywhere. We make this claim based on the following:

- 1) **Excellent Speakers with unparalleled access**
- 2) **Limited Attendance = 250 Registrants Max**
- 3) **No Vendors, No Sales Pitches, No Marketecture**
- 4) **Unmatched Value - less than 1/3 of the cost of a national conference**
- 5) **Since 2002, we have delivered over fifty(50) conferences throughout North America**

The No Fluff Just Stuff Software Symposium Series caters to individual developers, development teams, project managers, architects and independent consultants. The Great Lakes Software Symposium will offer 5 concurrent sessions over three days with over 55 sessions to choose from. The following topics will be featured:

- 1) ServerSide Java
- 2) .Net
- 3) XML / Web Services
- 4) Client Side Java
- 5) Architecture
- 6) Core Java

## Registration Fees

Attendees	Before 8/29/2005	After 8/29/2005
1-4	\$675	\$775
5-9	\$600	\$675
10-14	\$575	\$650
15-24	\$550	\$625
25+	\$525	\$600

## The Registration Fee includes the following:

- 1) All Access Pass to the three day symposium
- 2) Handouts from all sessions attended w/binder
- 3) CD with all presentational content @ registration
- 4) Custom NFJS Laptop/Backpack
- 5) Opportunity to win an iPod everyday during the symposium raffle

Go to <http://www.nofluffjuststuff.com> for more details.

Questions/Comments: Contact Jay Zimmerman: [zimmerman@nofluffjuststuff.com](mailto:zimmerman@nofluffjuststuff.com) or (303)469-0486.

# Great Lakes Software Symposium 2005

Sheraton Chicago Northwest - Arlington Heights, IL

September 16 - 18, 2005

(session listing as of 9/14/2005)

## **Ajax UI Design Tips and Techniques by Brennan Stehling**

The new Ajax model of web development allows for a new way to allow the user to interact with the system. The new model can have a positive or negative impact of the usability of an application. Many UI experts have addressed concerns and have recommended the proper UI designs.

## **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.

## **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.

## **Herdning 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?

## **Choosing An Agile Path: Agile Case Studies by David Hussman**

With more and more companies choosing to take an agile path, the bounds of agile ways and means are expanding. Agile implementations differ a great deal depending on the company, the project or product, and the players.

## **Managing Agile Projects: Dispelling the Myths by David Hussman**

There is a myth that agile projects do not need project management and that they cannot be estimated and planned. In this tutorial we will dispel those rumors. With a strong focus on small group exercises and experiencing the concepts, you will learn and experience why prescriptive planning fails, how agile planning succeeds, why a self-organizing team will always outperform a team managed through command-and-control, and how to tell when a project is on track as well as how and when to make an adaptive action to get it back on track.

## **Writing and Telling User Stories by David Hussman**

The technique of expressing requirements as user stories is one of the most broadly applicable techniques introduced by Extreme Programming. User stories are an effective approach on all time constrained projects, not just those using XP.

## **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.

## **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.

### **Transitioning to Agile # Keys to Success by David Hussman**

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

### **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.

### **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.

### **Agile Feature Priority Indifference Using /Functions and Facades by Doug Knesek**

Feature priority indifference is a primary agile design goal. It reduces or eliminates the ordering dependency of features by allowing features to be decoupled from each other. Traditional OO design approaches make feature decoupling difficult by encouraging the specification of interfaces that cross feature boundaries. /Functions and Facades/ is an approach to development that enables feature decoupling through hyper-narrow interfaces and inversion of control.

### **Seaside: A Radical Web Framework by Glenn Vanderburg**

We've been writing web applications now for 10 years, and they're still no fun. They're awkward and clumsy to write. Internally, they're overly complicated (which almost invariably means that they're buggy). Meanwhile, they're usually too primitive externally. To put it another way: the web programming model is so cumbersome for programmers that the users pay#through reduced features, clumsy interaction, bugs, and poor performance. There's a better way. I know -- who needs another web framework? But Seaside makes even Rails look primitive.

### **JavaScript Exposed: There's a Real Programming Language In There! by Glenn Vanderburg**

Now that web browsers are settling down and Ajax applications are on the rise, it's time to take JavaScript seriously. That means learning it the right way: looking at the fundamentals of the language and surveying its strengths and weaknesses, instead of just copying other people's poorly written examples. This talk takes that approach to JavaScript. It's a "no web pages" introduction to the JavaScript language.

### **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 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.

### **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.

### **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.

### **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.

### **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.

### **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.

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

Spring offers developers a simpler, more robust method for configuring applications. These benefits extend to security through the ACEGI framework. ACEGI makes the otherwise daunting task of securing your application logical and straightforward. More importantly, through its support for single sign-on provision through Yale's CAS system and its ability to provide instance-level authorization, Spring extends the common security model of most J2EE apps beyond what they are traditionally capable of.

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

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

### **Spring Intro by Justin Gehrtland**

The Spring framework is one of the fastest growing open source frameworks. New job postings are gaining rapidly, and many customers are adopting Spring instead of heavier alternatives. In this session, we'll introduce Spring. You'll see how Spring can give you much of the power of EJB, without the complexity or pain. Spring uses concepts like dependency injection and aspect oriented programming to ease standard enterprise development. Spring developers write plain, ordinary Java objects (POJOs), instead of sophisticated components. In this session, you'll see a basic Spring application. You'll also see some

details about some of the enterprise integration strategies, including: # Spring AOP # Transactions # Persistence # Model/view/controller When the session is over, you won't be an expert, but you should have a much clearer understanding of what Spring does, what it doesn't do, and why it's growing so rapidly.

### **Advanced Hibernate by Justin Gehtland**

Hibernate is easy to get started with, but can sometimes be hard to make efficient or secure. In fact, the default settings for Hibernate create applications that will run slowly, cause unwanted round trips to the database, and may be more restrictive and/or permissive from a security standpoint than you would otherwise want.

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

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

### **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.

### **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.

### **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.

### **Language Oriented Programming Part 2: Practice by Neal Ford**

This session builds on the theory presented in part 1 and shows how to use the new breed of tools, Language Workbenches, to build your own domain languages.

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

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

### **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).

### **Aspect-oriented refactoring: Taking refactoring to a new level by Ramnivas Laddad**

Refactoring allows reorganizing code while preserving the external behavior, while AOP facilitates modularizing crosscutting concerns in a system through use of a new unit of modularity called aspect. Aspect-oriented refactoring synergistically combines these two techniques to refactor crosscutting elements. Individually, refactoring and AOP both share the high-level goal of creating systems that are easier to understand and maintain without requiring huge upfront design effort. A combination of the two # aspect-oriented refactoring # helps in reorganizing code corresponding to crosscutting concerns to further improve modularization that is easy to understand, highly consistent, and simple to change.

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

Aspect Oriented Programming (AOP) enables modularizing implementation of crosscutting concerns that abound in practice: logging, tracing, dynamic profiling, error handling, service-level agreement, policy enforcement, pooling, caching, concurrency control, security, transaction management, business rules, and so forth. Traditional implementation of these concerns requires you to fuse their implementation with the core concern of a module. With AOP, you can implement each of the concerns in a separate module called

aspect. The result of such modular implementation is simplified design, improved understandability, improved quality, reduced time to market, and expedited response to system requirement changes. Come to this session and learn all about how AOP can help you simplify developing complex systems.

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

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

### **Effective AJAX by Rob Sanheim**

You have probably heard all the hype surrounding Ajax by now. But how do you actually go about implementing the next Google Maps or Basecamp? In this presentation we'll learn the several open source frameworks that make developing Ajax easier, faster, and more fun.

### **SWT Fundamentals by Scott Delap**

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

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

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

### **Creating Polished Swing Applications by Scott Delap**

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

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

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

### **5 Minutes Forms with JGoodies Binding and Validation by Scott Delap**

Application developers often spend hours on the simple tasks of laying out a form, wiring components to objects, and validating the data entered. This is time that could be much better spend on the business problems your application is trying to solve. This session will show how to leverage open source libraries to take the work out of the form building process.

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

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

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

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

### **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.

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

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

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

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

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

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

### **Test First Development by Venkat Subramaniam**

Do you know that unit testing is more of an act of design than verification? What are its benefits? How do we write effective tests? How does unit testing relate to evolutionary design? How does it help you with refactoring? When should you write your tests? What are the types of tests you could write? These are some of the questions that you would ask if you are interested in Unit Testing. What is a better way to learn than practicing it? In this session the attendees will participate in designing and developing a small yet full application. Instead of PowerPoint slides, you will learn from example. The code you help develop will be available for free download on the speaker's web site.

### **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.

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

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

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

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