

Greater Wisconsin Software Symposium 2006

Marriott Milwaukee West - Waukesha, WI

February 24 - 26, 2006

(session listing as of 2/22/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the Greater Wisconsin Software Symposium coming to Waukesha on February 24 - 26, 2006. 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 Greater Wisconsin 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) Core Java
- 3) Architecture
- 4) XML / Web Services
- 5) Client Side Java

Registration Fees

Attendees	Before 2/6/2006	After 2/6/2006
1-4	\$725	\$825
5-9	\$650	\$725
10-14	\$625	\$700
15-24	\$600	\$675
25+	\$575	\$650

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 or (303)469-0486.

Greater Wisconsin Software Symposium 2006

Marriott Milwaukee West - Waukesha, WI

February 24 - 26, 2006

(session listing as of 2/22/2006)

Unit Testing Java Objects with Groovy by Andrew Glover

What makes Groovy particularly appealing with respect to other scripting platforms is its seamless integration with the Java platform. Because it's based on the Java language (unlike other alternate languages for the JRE, which tend to be based on earlier predecessors), Groovy presents an incredibly short learning curve for the Java developer. And once that learning curve has straightened out, Groovy can offer an unparalleled rapid development platform.

Taking Quality to the Next Level through Code Coverage Analytics by Andrew Glover

Understanding what code coverage represents, how to effectively apply it, and how to avoid its pitfalls will give you an unprecedented understanding of how unit tests may or may not be covering you from sneaky defects.

Unit Testing Best Practices by Andrew Glover

In the years since JUnit's introduction, a number of frameworks have been built to enhance its utility for testing and validating XML, controlling the state of a database, testing legacy code, performance testing, and functional web testing.

Applied Design Patterns by Brian Sletten

Just about every modern software developer has a copy of the Gang of Four "Design Patterns" book sitting on a shelf; many of them have actually read it. The dark secret of the patterns community is that there is often a large gulf between whiteboard simplicity and real-world complexity. Language choice plays a part in the design (and even importance) of patterns. The situation is made even more confusing by the fact that many of the core patterns have now been "voted off the island" for one reason or another. This talk will give a pragmatic overview of the motivations behind design patterns and will focus on applying a handful of the GOF patterns to example scenarios in Java, Ruby and C#. A quick introduction to the role AOP plays in changing the patterns landscape will also be covered.

JmDNS : Easy Service Discovery for the 21st Century by Brian Sletten

Service-oriented architectures (SOAs) are all the rage. But how do you find all of these services once they are deployed? Configuration files are so 90's. Software of the 21st Century should be able to find related services and components without users having to specify particular configurations at start up. The IETF's ZeroConf multicast DNS protocol was designed to solve exactly this problem. JmDNS is Java-based open source implementation of this capability that allows local-link applications to find and use automatically discovered capabilities. Apple's Rendezvous technology is another open-source ZeroConf implementation behind many of the exciting applications it is building for OS X these days. Come learn how you can interact with these or your own service discovery-savvy applications without even having to learn how to spell UDDI. Bring your wireless notebooks to participate in a service-oriented environment (please have a working Java environment as we won't have time to debug installation issues).

Introducing the Semantic Web 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 the new vision of machine-processable documents and metadata to improve search, knowledge discovery and data integration and management. While there are many naysayers chiding such grand visions, there are also pragmatic and useful technologies emerging that can be applied today.

Politics of Persistence by Bruce Tate

This session will help a Java developer choose a persistence framework. After the session, you will # Understand the core strengths and weaknesses of the main persistence frameworks in the Java space # Understand where marketing influences can impact persistence # Know what's going on behind the scenes to impact the persistence pictures # Answer questions about persistence frameworks that might not be mainstream

Introduction to Hibernate by Bruce Tate

O/RM (Object/Relational Mapping) seeks to eliminate repetitive or tedious work enabling the CRUD (create, read, update, delete) that underlies most applications. Hibernate is a popular, open-source O/RM tool that uses reflection (instead of code generation, like EJB, or bytecode injection, like JDO) to manage your

persistence layer.

Three Technologies to Watch by Bruce Tate

The state of the art is progressing rapidly, and dynamic languages are driving the revolution. Find out about these topics that will be central to programming. We'll discuss continuation servers, metaprogramming frameworks and functional languages.

Where Agile meets Argyle: New processes in established companies by Bruce Tate

Agile programming is a collection of core principles and techniques that allow software developers to create lighter, more responsive applications, and to have fun doing it. Many established organizations are either openly or sub-consciously hostile to many of the principles of Agile development.

What's New in Spring 2 by Bruce Tate

In this session, we'll review the new features of Spring 2.0. If you've been using Spring 1.x, you'll want to hear about the improvements.

Testing your Rails Application by Dave Thomas

The Ruby on Rails framework has unit and functional testing baked right in. In this talk we'll see how easy it is to get started with testing in Rails, and we'll explore just how deep the testing support goes.

Herding Racehorses and Racing Sheep by Dave Thomas

Are you frustrated by experts who can't tell you what to do, or by junior team members who refuse to see the big picture? How can you best develop careers: both yours and those of your teammates and managers? How can we learn to apply experience more effectively, and why do the many approaches designed to tame complexity actually end up increasing it?

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.

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...)

Using Ajax with Ruby on Rails by Dave Thomas

Ajax is becoming a requirement for new applications: it creates richer user experiences and more dynamic applications. However, doing Ajax by hand is difficult and error prone. The good news is that if you use Rails, you don't have to do Ajax the hard way.

JSF: State of the Art by David Geary

In 2005, JSF hit its stride, as evidenced from overwhelming support from both vendors and the open-source community. JSF 1.0 had plenty of holes, but open-source projects have arisen to address those needs. This session takes a look at three of those projects:

- Tomahawk (MyFaces component library)
- Facelets
- Seam

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.

JavaServer Faces: A Whirlwind Tour by David Geary

JavaServer Faces (JSF) has arrived. The standard Java-based web application framework based on Struts, JSF really took off in 2005. Embraced by developers, vendors, and open-source projects, JSF has started to hit its stride. If you haven't come up to speed on JSF basics, this is the place to start.

Creating, Telling, and Tracking User Stories by David Hussman

The questions around user stories are many, and the list is only growing larger as their popularity of increases. Many organizations are on their path to adopting stories as requirements vehicles, possibly struggling with story writing as well as finding a way to fit them into their organization. The participants of this session will become agile customers and product owners, creating stories for project, organizing them into themes, and using them during mock planning activities. We will also discuss how to connect to product owners outside the project community and briefly review several tools for tracking and managing user stories.

#Show Me the Numbers# - Agile Planning Tools and Techniques by David Hussman

As agile grows, so too do the questions for how to track and communicate progress within the project community as well as to upper management and others interested in progress. This session will focus on tools and techniques for tracking an agile project plan from creation to project completion. We will create a simple plan in a planning tool, and run a mock project, showing how the plan addresses: communicating progress, addressing missed estimates, scope modifications, and more.

Automating Business Value with FIT and FitNesse by David Hussman

Agile communities consider stories #done# when the acceptance tests (also called story tests) are shown to the customer. Originally, this was a manual process, but in recent years, several frameworks have been created to automate this process, providing acceptance testing all the benefits of automated unit testing. One of the most popular of these is called FIT, created by Ward Cunningham. The presentation will briefly discuss stories, the origin and authoring of story tests, and a demonstration of how FIT and FitNesse (FIT living within a Wiki) can be used to automate acceptance tests.

The Agile Enterprise by David Hussman

As with many methodologies, moving agile into larger organizations poses larger challenges. There are many factors outside the developer world that can crash all the benefits of agile without regard to its success. This session will address how agile interacts with enterprise concepts like SOA, cross team collaboration / program management, and tracking at the enterprise level # and more. Various tools and techniques will be discussed, and at least part of the session will include Q/A for the presenter to field specific questions about your organization.

Shale: Turbo-charge your JSF Apps by David Geary

Struts is the most popular Java-based Web application framework today, but that's rapidly changing. There's a newcomer on the block, a leaner, meaner, better-designed framework loosely based on Struts that's poised to dethrone Struts as the reigning king of Java-based web application frameworks. That framework, of course, is JavaServer Faces. Craig McClanahan, the father of Struts and the co-spec lead for JSF 1.0, has proposed reinventing Struts for Struts 2.0 as a set of services for JSF applications. That new framework, which has no direct ties to Struts as we know it, is called Shale.

Losing Battles and Winning Wars: Adopting Agile by David Hussman

Adopting agile is different for each company, but most companies will go through some amount of change during the adoption of agile. This session will discuss some of the most common difficulties for adopting agile and provide various plans of attack. The session will start with a listing of issues for the session participants, and some portion of the session will be dedicated to an open forum where the presenter will address the issues collected.

Dependency Management Techniques by Kirk Knoernschild

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

From Code to Architecture by Kirk Knoernschild

The code we write has a tremendous impact on our software architecture. In fact, the code is our architecture. Not all of it, of course, but a good share of it. The problem is that we often don't fully comprehend the architectural impact of our code at the time we create it. One poorly designed class or method can severely impact the resiliency, stability, extensibility, and maintainability of your software. There are ways we can create flexible architectures...but we have to emphasize loose coupling among system

components.

Enterprise AOP with AspectJ by Ramnivas Laddad

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

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.

Myths and Realities of AOP by Ramnivas Laddad

Aspect-oriented programming (AOP) promises to modularize crosscutting concerns. Like all new technologies, AOP has its share of over zealotry and unjustified criticism, neither of which is useful to developers deciding if they should use AOP in their applications. Attend this talk to understand the real deal behind AOP and change your perspective of AOP forever.

Better Software Through Refactoring by Rob Sanheim

Refactoring is the process of improving the design of code without changing the behavior. The practice of refactoring is increasingly becoming a required skill for any professional developer, particularly as agile methodologies and incremental development spread. This presentation will bring you up to speed in refactoring for Java but will be applicable for any OO language.

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.

Introducing the Eclipse Rich Client Platform by Scott Delap

Rich client application development using Java can be intimidating given the vast flexibility in application design and structure. It also can be frustrating to create the large number of support services (persistence, menus, event and job frameworks) that a large scale rich client application needs. The Eclipse Rich Client Platform is one project attempting to solve these issues by providing a core infrastructure that not only provides the day to day services a rich client application developer needs, but also providing a suggested path to guide you down the road of designing your application. This presentation introduces both the Eclipse RCP and the tools provided by the Eclipse IDE that assist developers in writing RCP apps.

Real World Web Services by Scott Davis

Jonathon Schwarz (President and COO of Sun Microsystems) calls it #The Participation Age#. Tim O'Reilly (Founder and CEO of O'Reilly Media) calls it #Web 2.0#. Regardless of the name, the web is moving away from a static web page publishing model towards a Service-Oriented Architecture (SOA) model where the data is at least as important as the presentation, if not more so.

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...

Java GUI Deployment 101 by Scott Delap

This session will introduce attendees to the various options and problems that exist with deployment of a Java UI application written in Swing or SWT. Topics covered will include obfuscation of classes in preparation of deployment, open source installers, and Java Webstart.

Ajax, Flash, and Java - Choosing The Right Rich Client Technology for Your Next Project by Scott Delap

Today's users are beginning to demand richer and richer application experiences. Plain html pages simply don't cut it anymore. Applications like Google Maps (Ajax) and Yahoo Maps (Flash) show how the UI experience can be pushed to the next level. As an IT manager, how do you decide which route to take however? Should you use Ajax because it is the new "it" technology. Is Flash a viable option with its 95%+ browser availability? Perhaps Java deployed through web start is really the best choice in contrast to what the buzz would lead you to believe. This presentation takes a look at these three core rich client technologies from both deployment/user experience and ease of development perspectives.

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.

Java Platform Security and JAAS by Stuart Halloway

The Java platform is built from the ground up with security in mind. This talk will introduce the security features of the J2SE, building quickly from the basic classes to realistic examples.

Cryptography for Programmers by Stuart Halloway

For centuries people have used crypto to build (and break) secure systems. Computers have only raised the pitch of conflict, providing enormous cryptographic power at commodity prices. Most programmers do not write their own crypto libraries, instead relying on the services of an operating system or virtual machine. But even with all this support, building secure systems is a daunting task.

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.

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.

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.

Advanced Hibernate by Stuart Halloway

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.

Programming Java Concurrency by Stuart Halloway

Java has always provided a model for concurrency and threads. With Java 1.5, this model received a major facelift. Learn how to use the new concurrency utilities to build responsive, scalable, and correct concurrent applications.

J2EE Security @ Work: J2EE Meets JAAS by Tom Marrs

Have you wasted time writing lots of security-based code and ever wondered if there's a better way to add security to your application? Are you confused by declarative security? Have you read about JAAS (Java

Authentication and Authorization Service) but wondered where it fits? Have you ever said, "Can I just see a working example"? If so, then this talk is for you. Have you ever said, "Can I just see a working example"? If so, then this talk is for you.

J2EE Web Services @ Work by Tom Marrs

Have you tried to deploy J2EE Web Services and thrown up your hands in frustration at the lack of tool support? Have you been confused by the tangled web of new deployment descriptors? Do you want to know how to develop and deploy J2EE-compliant Web Services so that it works every time? If so, then this talk is for you. at the lack of tool support? If so, then this talk is for you.

Java/J2EE Architecture @ Work: EJB 3 vs Spring and Hibernate by Tom Marrs

You've used EJB in the past and been disappointed - it was too heavy and difficult to use. Like Bruce Tate, maybe you've gone from "Bitter" to "Better, Faster, Lighter". With EJB 3 shipping in early 2006, maybe it's time to take another look. We'll compare EJB 3 with alternative frameworks - Spring and Hibernate - to see if EJB 3 has closed the gap.

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.

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

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

Open Source Tools for Agile Development by Venkat Subramaniam

As a Java developer, you have taken the time to learn the basics of the language and relevant parts of its rich API. However, you need more than that to develop serious industrial strength applications. In this presentation, the speaker will introduce you to a number of open source tools which you can use to improve your application quality and your development process.

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.

Groovy for Java Programmers by Venkat Subramaniam

Object-oriented scripting languages, or agile dynamic languages, as some like to call those, are gaining programmers' attention. Groovy bring this excitement to the Java platform with its ability to generate byte code. You can use Groovy instead of Java for some parts of your application. By learning it, you can switch between the languages where you consider fit.

Refactoring your code - a key step in agility by Venkat Subramaniam

Refactoring is one of the core practices in Agile Software Development. Refactoring is based on some core principles that apply to more than writing good code. But, what's refactoring? Why should you do it? How do you go about doing that? What tools are available to successfully refactor your App?