

Greater Wisconsin Software Symposium

Sheraton Milwaukee Brookfield

Feb. 27 - Mar. 1, 2009

<http://www.nofluffjuststuff.com/conference/milwaukee/2009/02/index.html>

The No Fluff Just Stuff Java Symposium Series is proud to announce the return of the Greater Wisconsin Software Symposium on Feb. 27 - Mar. 1, 2009. GWSS2009 will be held at the Sheraton Milwaukee Brookfield.

Since 2002, the No Fluff Just Stuff Java Symposium has been regarded as the premier Java/Agility event series anywhere serving over 21,000 attendees with some 130 events. The popularity of the NFJS symposium series can be traced to the following:

- 1). Exceptional Speakers
- 2). Limited Attendance - capped at 250 people
- 3). No Vendors, No Sales Pitches, No Marketecture
- 4). Excellent networking opportunity with speakers and fellow attendees because of small size.
- 5). The Best Value in the Java conferencing space period.

You will have the opportunity to attend multiple sessions on variety of interesting topics covering core Java, Enterprise Java, Web 2.0, Dynamic Languages, Architecture, Security, Testing, CI and Agility.

The Registration Fee Includes:

3 Day All Access Pass to GWSS2009

All Meals/Snacks - duration of the symposium

Complimentary copy of "Career 2.0" by Jared Richardson

1-Year IntelliJ license compliments of JetBrains

Session Materials

Custom NFJS Binder

Great Giveaways @ NFJS including \$200 and \$400 Apple Gift cards

Early Bird Registration: \$825/person good thru 2/9/09 after \$925

Excellent Group Discounts Available - bring your entire development team to the show - no travel required!! Rate good thru 2/9/09

Registration Fees

Attendees	Before Feb. 9, 2009	After Feb. 9, 2009
5-9	\$725	\$825
10-14	\$700	\$800
15-24	\$675	\$775
25+	\$650	\$750

Go to <http://www.nofluffjuststuff.com/conference/milwaukee/2009/02/index.html> and register today!

Greater Wisconsin Software Symposium

-Session Schedule-

(event schedule as of February 24, 2009)

Friday, Feb. 27

12:00 - 1:00 PM : REGISTRATION

1:00 - 1:15 PM : WELCOME

1:15 - 2:45 PM - Sessions

Session #1 : Common AntiPatterns and How To Avoid Them by Mark Richards

In the book "97 Things Every Software Architect Should Know" (O'Reilly, 2009) I wrote about the importance of design patterns as a useful means of communication between architects and developers. Equally important to patterns is an understanding of AntiPatterns - things that we repeatably do that produce negative results. AntiPatterns are used by developers, architects, and managers every day and are one of the main factors that prevent progress and success. In this session we will look at some of the more common and significant development and architecture antipatterns. Through coding and design examples, you will see how these antipatterns emerge, how to recognize when the antipattern is being used, and most importantly, how to avoid them. By attending this session, you will be part of a movement to reduce the AntiPattern catalog from hundreds of entries to only a few. **Prerequisite:** None

Session #2 : JSF 2.0: An Introduction by David Geary

This session introduces JSF 2.0 fundamentals, with emphasis on new features in JSF 2.0. **Prerequisite:** Familiarity with JSF, or other component-based frameworks

Session #3 : The Amazing Groovy Weight-loss Plan by Scott Davis

"The central enemy of reliability is complexity." (Dr. Daniel Geer) Java is a powerful programming language. A smart developer can do nearly anything with Java. So the next question is, "How quickly can it be done? How many lines of code does it take to do common tasks?" Groovy greases the wheels of Java by decreasing the complexity of the language while preserving the raw power. At first glance, you might think that this talk is simply about how Groovy drastically reduces the lines of code you need to write. What this talk is really about is bringing simplicity, clarity, readability, and yes, beauty to your source code.

Session #4 : Going Beyond "Hello Agile": Your Questions and Mine by David Hussman

If you are truly working to get value from agile methods, and you have been doing it for some time, you probably have some questions which go beyond "my first agile project." If you are looking for a place to get answers or hear where and how others are struggling, come to this session ready to ask your tough questions. I have coached many communities (of all shapes and sizes) adopt, adapt, and evolve agility beyond the first project or the first few months, and I am sure there will be no shortage of examples and experiences for you questions. Also, I am sure you will learn from others in the audience as well.

2:45 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

Session #5 : On Being a Software Architect by Mark Richards

One way to stop a conversation dead while at a party or gathering is to mention you are a software architect. Why? Because it takes about an hour (complete with Powerpoint slides) to explain what you do for a living. By then the person you are talking to is so bored they would rather sit in a corner licking nine-volt batteries. The problem is that no one inside or outside our industry really knows what a software architect is or what they do. In this highly interactive (and slightly humorous) session we will take a deep dive into the role a software architect plays in the IT industry. We will explore the characteristics an architect needs to have, and the elements that make a good architect and a bad architect. Through amusing antidotes and real-world examples, we will see how to become an effective software architect and help shape the industry in terms of the role and title of software architect. **Prerequisite:** None

Session #6 : JSF 2.0: Ajax and Custom Components by David Geary

This session covers advanced aspects of JSF 2.0. **Prerequisite:** Familiarity with JSF, or other component-based frameworks. Familiarity with Ajax. This session builds on demos shown in the JSF 2.0 Introduction talk, so it is helpful, although not required, if you attend the intro talk before coming to this session.

Session #7 : Groovy XML Ninja Skills by Scott Davis

"XML is like violence: if it doesn't solve your problem, you aren't using enough of it." (Anonymous) XML is everywhere. Whether you are dealing with local configuration files (web.xml, struts-config.xml) or remote web services (SOAP, REST, RSS, Atom), the modern software developer needs to be able to request, slice, and dice XML with ease. That requires a set of razor-sharp tools that reduce the inherent complexity of the problem, not multiply it. Once you see XML tremble in fear at the awesome power of Groovy, you'll wonder what you ever did without it.

Session #8 : Agile Buy-In Starts with Smart Sales by David Hussman

People ask me all the time, "How do you get people to buy in the use of agile methods?" While this is easier than it once was, there are still many challenges, and agile snake oil sales are on the rise. If you are looking to sell agility or deeper your agile investment (or your sales force), this session will provide you with tools that will help you frame your sell points, select your sales tools and communicate the value (of a practice) over the mechanics.

Greater Wisconsin Software Symposium

-Session Schedule-

(event schedule as of February 24, 2009)

4:45 - 5:00 PM : BREAK

5:00 - 6:30 PM - Sessions

Session #9 : Introduction to JMS by Mark Richards

There's no doubt about it - messaging is quickly becoming a standard part of most application architectures, particularly as more and more companies struggle to find ways to integrate heterogeneous environments due to mergers, acquisitions, or to streamline existing application portfolios. The Java Message Service (JMS) API allows Java applications to implement messaging using a standard API, therefore removing the dependency of any particular messaging provider. In this introductory session we will take a look at the basics of messaging and the JMS API. I will start by discussing the different messaging models, the structure of a basic JMS message, and the JMS API interfaces and how they interrelate. Then through interactive coding I will show the basics of sending and receiving messages using the point-to-point messaging model and how to do request/reply processing. NOTE: this session is meant to be an introduction to messaging and JMS - no prior JMS or messaging experience is needed for this session. **Prerequisite:** None

Session #10 : Flex for Java Developers by David Geary

An introduction to Flex for Java developers. **Prerequisite:** Familiarity with Flex and at least one other web application framework

Session #11 : DSLs in Groovy: Say What You Mean, Mean What You Say by Scott Davis

"Simplicity is the ultimate sophistication." (Leonardo da Vinci) The history of computer programming has been bridging the gap between what the user says ("We need to add sales tax to each item in the order") and what the programming language requires you to say ("for Iterator i = orderList.iterator();"). Building Domain Specific Languages (DSLs) allow you to express the solution in the language of the domain user instead of the language of the programmer.

Session #12 : 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.

6:30 - 7:15 PM : DINNER

Keynote: Keynote: On the Lam from the Furniture Police - Neal Ford

Saturday, Feb. 28

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

Session #13 : Emergent Architecture & Design by Neal Ford

Most of the software world has realized that BDOF (Big Design Up Front) doesn't work well in software. But lots of developers struggle with this notion when it applies to architecture and design. Surely you can't just start coding, right? You need some level of understanding before you can start work. This session describes the current thinking about emergent architecture & design. This philosophy allows you to specify only the critical items up front and allow the important architecture and design criteria emerge once you understand the problem better. To make this happen, you must have several infrastructure in place. This session describes the necessary infrastructure to make this possible and then how to tease out architecture and design as you evolve software.

Session #14 : GWT: An Introduction by David Geary

An introduction to Google Web Toolkit. **Prerequisite:** Familiarity with a component-based framework, preferably a desktop application framework

Session #15 : Dim Sum Grails: A Sampler of Practical Non Database-Driven Grails Applications by Scott Davis

"The proof of the pudding is in the eating. By a small sample we may judge of the whole piece." (Miguel de Cervantes Saavedra) Most Grails tutorials demonstrate how easy it is to build simple CRUD (Create/Retrieve/Update/Delete) applications. While skinning a database with a web front-end is undeniably one beneficial aspect of Grails, it isn't the only thing Grails is good for. As you'll see here, Grails can be used to build a wide variety of web applications. You won't see a single HTML table with "edit" and "delete" links, I promise.

Session #16 : What Is Lean And Why Do You Care? by David Hussman

Whether it was intentional or not, the agile community has been borrowing successful ideas from the lean manufacturing for years. Lean practices, like finding and removing wasteful work, can be applied without needing special permission or certification. Ideas like kanban (visual planning aids) and kaizen (continuous learning) are simple, helpful tools that are easily applied and produce great results.

10:30 - 11:00 AM : BREAK

Greater Wisconsin Software Symposium

-Session Schedule-

(event schedule as of February 24, 2009)

11:00 - 12:30 PM - Sessions

Session #17 : Transaction Pitfalls and Strategies by Mark Richards

In previous years I have given sessions related to my book "Java Transaction Design Strategies", where I have reviewed the basics of programmatic and declarative transactions and outlined the basic patterns described in the book. In this new session for 2009 I will focus on some of the pitfalls encountered while dealing with transactions and then how to develop an effective transaction strategy. I will start this session by describing and illustrating some of the common pitfalls I continue to see in both Spring and EJB. I will then describe four common transaction strategies you can use and implement, including a transaction strategy for high-speed transactions, a transaction strategy for client orchestration, a transaction strategy for use with API's, and finally a strategy for highly concurrent environments. Note: This session assumes you know a little bit about transactions and have been using them in either Spring or EJB. It is not intended to be an introductory session on how transactions work. You can obtain a free PDF download of my transaction book at <http://www.infoq.com/minibooks/JTDS> to quickly come up to speed with transactions. **Prerequisite:** Java, Spring or EJB; some knowledge of transactions and JTA.

Session #18 : GWT: Advanced Topics by David Geary

Learn to do really cool stuff with GWT. **Prerequisite:** Knowledge of topics covered in the GWT introduction session

Session #19 : Lizard Brain Web Design by Scott Davis

"There's an old story about the person who wished his computer were as easy to use as his telephone. That wish has come true, since I no longer know how to use my telephone." (Bjarne Stroustrup) The "lizard brain" is the oldest part of the human brain -- the part responsible for autonomic functions like breathing, heart rate, and navigating websites. OK, maybe not that last part, but your website should be easy to use. Stupid easy. Lizard brain easy. Any time your user spends figuring out how to do something -- even for a split second -- is wasted time due to poor design. Inspired by Steve Krug's book "Don't Make Me Think", this talk answers the question, "Why is that website so hard to use?"

Session #20 : 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!

12:30 - 1:30 PM : LUNCH

1:30 - 3:00 PM - Sessions

Session #21 : Advanced Topics in JMS by Mark Richards

This session covers some of the more advanced features of JMS messaging, and is intended for those who are familiar with JMS and messaging in general. Some of the topics I will be covering in this session include message grouping and batching, transacted sessions, client-based acknowledgement, message correlation alternative, and some various messaging design considerations and things to watch out for from a design and coding perspective. I will be doing live coding demonstrations to illustrate the techniques described in this session. Although this session is entirely JMS provider agnostic, I will be using ActiveMQ, a popular open source JMS provider, during the live coding demonstrations. **Prerequisite:** Some knowledge of messaging and JMS would be helpful

Session #22 : What's New in Spring 3 by Ken Sipe

The Spring Framework has led the industry in innovation for years. Starting with dependency injection and promoting testing through removal of framework dependencies. Spring 3.0 continues that innovation in a way that takes full advantage of the Java 5 platform. There are a number of significant changes to the framework. So whether you are new to the framework or an experienced Spring developer, this is a great session to come up to speed on the latest from SpringSource. **Prerequisite:** Java 5

Session #23 : Web 2.0 Checklist: Deconstructing Modern Websites by Scott Davis

"The challenge of modernity is to live without illusions and without becoming disillusioned." (Antonio Gramsci) There are plenty of sarcastic "Web 2.0" checklists out there -- be perpetually in BETA, when in doubt add rounded corners, etc. While we can all laugh at the superficial aspects of the Web 2.0 revolution, there are plenty of serious aspects to it as well. Is your website mash-up friendly or hostile? Do you tell your visitors when things change (via RSS or Atom syndication), or do you expect them to check in daily for updates? Is your website a silo or a part of a larger ecosystem?

Session #24 : Real-world Refactoring by Neal Ford

Refactoring is a fine academic exercise in the perfect world, but we don't really live there. Even with the best intentions, projects build up technical debt and cruddy bad things. This session covers refactoring in the real world, at both the atomic level (how to refactor towards composed method and the single level of abstraction principle) to larger project strategies for multi-day refactoring efforts. This talk provides practical strategies for real projects to effectively refactor your code.

3:00 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

Greater Wisconsin Software Symposium

-Session Schedule-

(event schedule as of February 24, 2009)

Session #25 : Effective Java by Venkat Subramaniam

Java is a well established language, that has been around for more than a decade. Yet, programming on it has its challenges. There are concepts and features that are tricky. When you run into those, the compiler is not there to help you.

Session #26 : 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?

Session #27 : Structuring concurrent applications in JDK 5.0 by Brian Goetz

JDK 5.0 is a huge step forward in developing concurrent Java classes and applications, providing a rich set of high-level concurrency building blocks.

Session #28 : Hands-on Agile Development by Neal Ford

This hands-on session applies agile development practices in solving a real problem in a short time frame. Come visit a day in the life of an agile project.

4:45 - 5:30 PM : BIRDS OF A FEATHER SESSION

Sunday, Mar. 1

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

Session #29 : Building External DSLs by Venkat Subramaniam

Domain Specific Languages (DSLs) are languages targeted at a particular problem and domain. They have context and are fluent. They help users of applications at various levels to easily communicate with your application. Developing DSLs, however, are not easy. You could easily get dragged into using parsers and tools with steep learning curve.

Session #30 : 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.

Session #31 : Effective Concurrent Java by Brian Goetz

The Java programming language has turned a generation of applications programmers into concurrent programmers through its direct support of multithreading. However, the Java concurrency primitives are just that: primitive. From them you can build many concurrency utilities, but doing so takes great care as concurrent programming poses many traps for the unwary.

Session #32 : Agile, Relevance Style by Stuart Halloway

The Agile Manifesto, like any good scripture, admits of many interpretations. There is no one "right path." What works for us may not work for you. At Relevance we have tried many paths, and learned many lessons. Join us to see dozens of ideas that have worked for us, plus some that haven't.

10:30 - 11:00 AM : MORNING BREAK

11:00 - 12:30 PM - Sessions

Session #33 : Cleaning up Code Smell by Venkat Subramaniam

Projects often start out simple, but soon become complex and turn into a lose cannon. Organizations are struggling to maintain and evolve software. Poor code quality is a significant part of that problem. Improving the quality of code is critical to success of enterprise projects.

Session #34 : Security Boundaries by Ken Sipe

Security is a large concern in today's world of software development. Security is a multi-dimensional problem requiring skills at a number of different levels. This session is a security overview of a typical Java web development stack.

Session #35 : The Java Memory Model by Brian Goetz

What's the worst thing that can happen when you fail to synchronize in a concurrent Java program? Its probably worse than you think -- modern shared-memory processors can do some pretty weird things when left to their own devices.

Session #36 : Taking Agile From Tactics to Strategy by Stuart Halloway

Teams adopting agile should begin at a tactical level, but they shouldn't end there. The Agile Manifesto operates at many different levels. Learn to apply the principles of agile at a strategic level. Otherwise you can have a great agile ground game and still lose.

Greater Wisconsin Software Symposium

-Session Schedule-

(event schedule as of February 24, 2009)

12:30 - 1:15 PM : LUNCH

1:15 - 2:15 PM : EXPERT PANEL DISCUSSION

2:15 - 3:45 PM - Sessions

Session #37 : Know your Groovy? by Venkat Subramaniam

Groovy brings the dynamic productivity to the Java platform. One of the strengths of Groovy is the seamless integration with Java—it preserves the Java semantics. However, Groovy does have some differences that can surprise you if you're not expecting.

Session #38 : 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>.

Session #39 : Are All Web Applications Broken? by Brian Goetz

Many developers believe that web frameworks "take care of" the details of concurrency, but this is only because most web applications make limited use of state. Stateful web applications also need to be careful about hazards like races. This talk will use the Java Memory Model to analyze common patterns of state management in web applications. **Prerequisite:** *The Java Memory Model*

Session #40 : 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.

3:45 - 4:00 PM : BREAK

4:00 - 5:30 PM - Sessions

Session #41 : BDD in Java and Groovy by Venkat Subramaniam

In this presentation we will take a look at what BDD is and look at tools to create them in Java and Groovy.

Session #42 : Programming Clojure by Stuart Halloway

Find out why Clojure is Java.next: * Clojure provides clean, fast access to all Java libraries. * Clojure provides all the low-ceremony goodness you know and love from dynamic languages such as Ruby and Python. * Clojure includes Lisp's signature feature: Treating code as data through macros. * Clojure's emphasis on immutability and support for software transactional memory make it a viable option for taking advantage of massively parallel hardware.

Session #43 : Garbage-collector-friendly programming by Brian Goetz

To many developers, garbage collection is black magic. Accordingly, there are a lot of conflicting advice about what is good or bad for the garbage collector. In this talk, I look at how garbage collection is implemented in the HotSpot VM, and techniques for writing programs that exhibit good garbage collection behavior. Surprisingly, many of these techniques coincide with writing good, clean code.

Session #44 : 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 criteria 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.