

New York Software Symposium 2006

Four Points by Sheraton Newark - Elizabeth, NJ

August 18 - 20, 2006

(session listing as of 8/20/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the New York Software Symposium coming to Elizabeth on August 18 - 20, 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 75 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 New York Software Symposium will offer 4 concurrent sessions over three days with over 44 sessions to choose from. The following topics will be featured:

- 1) Architecture
- 2) Groovy
- 3) Core Java
- 4) ServerSide Java
- 5) XML / Web Services

Registration Fees

Attendees	Before 7/31/2006	After 7/31/2006
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) Great NFJS Swag
- 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.

New York Software Symposium 2006

Four Points by Sheraton Newark - Elizabeth, NJ

August 18 - 20, 2006

(session listing as of 8/20/2006)

Improving Java code quality with code auditing tools by Brian Goetz

Does your program have bugs, despite unit tests, integration tests, and code reviews? You bet. Fortunately, there are some new code auditing tools that can help spot some bugs missed by other approaches.

Introduction to Java threads by Brian Goetz

The Java language included support for threads and concurrency from day 1, but writing correct multithreaded programs is not easy. This session will cover the how and why of using threads in Java.

Java Performance Myths by Brian Goetz

Performance myths about the Java platform abound, from the general "Java is slow", to the more specific "reflection is slow", "allocation is slow", "synchronization is slow", "garbage collection is slow", etc. Many of these myths have their root in fact (in JDK 1.0, everything was slow); today, not only are many of these statements not true, but Java performance has surpassed that of C in many areas, such as memory management.

Structuring concurrent applications in JDK 5.0 by Brian Goetz

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

The Java Memory Model by Brian Goetz

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

Cascading Style Sheets: a Programmer's Perspective by Eitan Suez

Today, the Cascading Style Sheets (CSS) specification is well supported by the major browsers (Mozilla, Safari, IE). CSS has become a practical tool for web content publishers that has helped turn heavy, buggy, and hard-to-maintain web sites into lean, clean, and stylish ones. CSS is sometimes stereotyped as a technology geared for graphic designers and artists. I beg to differ: I see CSS as a refactoring tool for content publishers and one that encourages content to become more strongly semantic. Come see a developer's perspective on CSS and how it can be applied to refactor your web content.

Extreme Agility with jMatter by Eitan Suez

The jMatter framework is a modern implementation of the Naked Objects Architectural Pattern using Swing, Hibernate, and deployed with Java WebStart. This open-source framework produces 2-tier workgroup apps (Swing front-ends that talk to rdbms back-ends) intended to be used in a LAN or VPN environment. Developers using a Naked Objects style framework focus on building a behaviourally complete domain model and leave everything else (UI, persistence, etc) to the framework. By focusing on the domain model only, jMatter claims to offer 10x productivity for building Swing workgroup apps. Come discover jMatter in a hands-on presentation where we'll be developing a live application and hold discussions about this new empowering style of producing business applications.

Hibernate by Example by Eitan Suez

This talk covers the core of the Hibernate Object/Relational Mapping framework by example; that is: in a hands-on manner.

XML Data Binding with JiBX by Eitan Suez

JiBX is an open source XML data binding API for Java. JiBX is younger than most other APIs in this space (Castor XML, BEA XMLBeans, JAXB). JiBX's philosophy on data binding is that: [a] databinding should be fast, and [b] databinding frameworks should allow for the divergence and evolution of your codebase from its xml representation. JiBX excels on both counts and consequently is a practical tool for the purpose of data binding. In this session, Eitan will be covering all aspects of Dennis Sosnoski's JiBX framework.

8 Steps to Struts2 by Ian Roughley

This presentation introduces the features of Struts2, and the framework differences between it and Struts, by iteratively migrating a simple application in 8 steps.

Evaluating Open Source Solutions by Ian Roughley

Many companies and most, if not all, software today utilizes open source. Whether it is databases, application servers, frameworks or libraries, these projects are fast becoming a standard commodity for building business-related functionality upon and speeding up development time. Sometimes technology evaluations are done, but frequently the library is simply slipped into the code base to address an urgent requirement - often without evaluating the technology beyond the immediate need.

EJB 3 Part 1: Core Spec and Spring Comparison by Mark Richards

The new EJB 3 specification (JSR-220) offers some great improvements over the prior EJB specs in terms of development simplicity and new features. In this session we will explore in detail some of the new features of the core EJB 3 specification. Included in this session will be defining and accessing session beans, JTA transaction management, declarative security, and interceptors. During the session I will demonstrate the new features of EJB 3 through interactive coding examples. We will then look at how the EJB 3 specification differs from the Spring Framework, where each is useful, and speculate as to what will happen in the future with these two frameworks. This session is part one of a two-part EJB 3 session (part two covers the new Java Persistence API).

EJB 3 Part 2: Java Persistence API (JPA) by Mark Richards

In addition to providing a simplified API, the new EJB 3 specification (JSR-220) defines a standard ORM Java Persistence API (JPA) that replaces those nasty Entity Beans that were part of the EJB 2.x specification. As you will see in this session, JPA bears a striking resemblance to popular ORM solutions like Hibernate and Toplink. In this session we will explore in detail the new Java Persistence API offered by JSR-220. We will start by discussing the overall design and architecture of the JPA and how the major components within JPA interact. We will then look at defining mapping objects (entities) and how to use the EntityManager to manage these entities. Through interactive coding examples we will see how to use the JPA for simple queries, complex queries, and finally stored procedures. This session is part two of a two-part EJB 3 session.

Java EE Command Pattern Architecture by Mark Richards

Tired of dealing with EJBs but cannot use other frameworks like Spring? How would you like to replace all of your remote Stateless Session Beans with POJOs and still access them remotely within Java EE? By using the Java EE Command Pattern we can write EJBs as POJOs and solve many of the issues facing EJB, including testability, configuration complexity, and performance, and still remain within our behemoth Java EE environment. The Java EE Command Pattern is a simple pattern that can significantly reduce the complexity of large-scale Java EE enterprise applications. In this session we will explore the numerous issues facing a typical EJB architecture and learn how the use of the Java EE Command Pattern can solve these issues. We will walk through the different design alternatives and see how the command pattern is implemented. Through interactive coding examples you will learn what components make up the Command Pattern framework and what simple coding changes are required to convert a complex remote EJB-based application to a much simpler remote POJO-based application. Finally, we will see how the Java EE Command Pattern helps facilitate the move towards Service Oriented Architecture (SOA)

Techniques in Architecture Agility by Mark Richards

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

Real-world Agile Development by Neal Ford

Lots of developers want to use Agile development technique but don't know where to start. This session discusses how to get started with Agility, the key benefits you can expect, and the pitfalls to avoid.

SOA: Next Wave of Enterprise Development or Return of the Son of CORBA? by Neal Ford

Is Service Oriented Architecture 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. I talk about SOA, ESB, CORBA, your MOM, and a bunch of other acronyms.

Testing with Selenium by Neal Ford

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

The Productive Programmer by Neal Ford

This session shows you how to become a more productive programmer every day by using tools that you didn't know you already had.

Web Application Security Vulnerabilities by Neal Ford

This session highlights common mistakes made by web programmers, stating the problems and avoidance techniques.

Continuous Integration using CruiseControl and Subversion by Paul Duvall

Continuous Integration (CI) is the process of continually building and testing your software under development. It is identified as a core XP practice, although it works with many software development processes.

Continuous Testing by Paul Duvall

The key to improving the reliability of your software is to run tests whenever a change occurs. Continuous Testing leverages the practice of continuous integration (CI) to ensure highly reliable code.

Domain Driven Design with AOP and DI by Ramnivas Laddad

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

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.

Testing Strategies for Web Applications by Ramnivas Laddad

Ever wondered if you can automate testing of your web application, but couldn't produce a satisfactory solution? If so, this is the session for you! Attend this session to understand the alternatives you have for unit and functional testing of web applications.

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.

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.

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.

JavaScript for Ajax Programmers by Stuart Halloway

This presentation covers JavaScript from the perspective of an Ajax programmer. We assume that you may be using an Ajax toolkit, but still need to be able to read, modify, and test the JavaScript code in your application. You will learn the common idioms of JavaScript by looking at working code from the Ajax toolkits themselves.

Prototype: Ajax and JavaScript++ by Stuart Halloway

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

Spring Dependency Injection by Stuart Halloway

Dependency Injection (DI) is the cornerstone of Spring. The core concept is quite simple, but (surprise!) actual practice can become complex. To take full advantage of Spring DI, you need to understand not only the basics on configuration, but also the container lifecycle model and the various hooks provided by the framework.

Spring Fundamentals by Stuart Halloway

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 Security with ACEGI by Stuart Halloway

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

Extend the Customization Possibilities of Your Java App with Scripts by Ted Neward

Ever wished you could just put parts of your program in end-users' hands and let them build the infinite little changes they want? Ever thought about how you might make your application more robust by writing less code, not more? Embed a scripting engine into your application--complete with the safeguards necessary to ensure that users can't do anything they shouldn't be able to--and release yourself from the Principle of Perpetual Enslavement.

Java5: The Language, The Libraries, The VM by Ted Neward

Java5 introduced a whole slew of new features, including annotations (JSR 175), new language features (the enhanced for loop, generics, static imports, and more), new library support (java.lang.instrument, among others), and some interesting enhancements to the virtual machine itself.

Effective Enterprise Java: State Management by Ted Neward

Managing state--both transient state (like your shopping cart) and your durable state (like your order placements, your inventory management forms, and so on)--is tricky in an enterprise application. In this talk, we'll examine some of the trickiness, both high-level and low-

Pragmatic XML Services by Ted Neward

There's a lot of talk about web services, and most of it falls into one of two categories: lots of low-level talk about vendor-specific tools and extensions, or lots of high-level talk that never shows you a line of code. XML services aren't that hard, and in this talk, we'll see how, why and when to do one.

Get Groovier with Grails by Venkat Subramaniam

Inspired by the Ruby on Rails project, Grails brings the ease of web development and "convention over configuration" to the Java platform. We will learn how to create web applications using Grails, how to integrate it with Hibernate, and how to Ajax it, all using the built in features of Grails. This section assumes that you are familiar with Groovy or you have attended the #Groovy for Java Programmers# session. The session will be example driven with live coding where we will build a web application from scratch.

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.

Open Source Tools for Agile Development by Venkat Subramaniam

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

Practices of an Agile Developer by Venkat Subramaniam

You have worked on software projects with varying degree of success. What were the reasons for the success of your last project? What were the reasons for those that failed? A number of issues contribute to project success - some non-technical in nature. In this presentation the speaker will share with you practices in a number of areas including coding, developer attitude, debugging, and feedback. The discussions are based on the book with the same title as the talk.

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?

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.

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.