

New England Software Symposium 2006

Sheraton Framingham - Framingham, MA

Sep. 29 - Oct. 01, 2006

(session listing as of 10/1/2006)

The No Fluff Just Stuff Software Symposium 2006 tour is pleased to announce the New England Software Symposium coming to Framingham on Sep. 29 - Oct. 01, 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 England Software Symposium will offer 5 concurrent sessions over three days with over 55 sessions to choose from. The following topics will be featured:

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

Registration Fees

Attendees	Before 9/8/2006	After 9/8/2006
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: jzimmerman@nofluffjuststuff.com or (303)469-0486.

New England Software Symposium 2006

Sheraton Framingham - Framingham, MA

Sep. 29 - Oct. 01, 2006

(session listing as of 10/1/2006)

Introduction to TestNG, the next generation testing framework for developers by Andrew Glover

No one will argue that JUnit has positively affected the quality of thousands of Java applications around the world. JUnit's simplicity and ease of use ushered in a whole new era of code quality; however, as many developers have found, its simplicity has also limited its use. TestNG was designed from the ground up to overcome some of JUnit's limitations; moreover, TestNG's features make it a great tool to complement your JUnit tests.

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.

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.

Pragmatic Learning by Andy Hunt

How you learn new technology and acquire new skills is key to your personal success. But how do you learn how to learn? What tricks tips can you use to learn more faster, and retain more of what you learn?

Refactoring Your Wetware by Andy Hunt

Software development happens in your head; not in an editor, IDE, or design tool. We're well educated on how to work with software and hardware, but what about wetware -- our own brains?

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.

Applied Object-Oriented Metrics by Brian Sletten

Object-oriented code metrics are a little like Artificial Intelligence: those who did it twenty years ago roll their eyes at the thought and prophesy the same ultimate failure at applicability now. Those who grew up with Java are approaching the topic with new eyes and are finding useful ways of incorporating metrics into their projects. Come hear about tools and ways to measure properties of software, how they might be beneficial and where you are likely to go astray with this approach.

Applied REST by Brian Sletten

REST sounds like such a simple thing. But, what is it really? How do you convince your boss to let you try it when she has been sold on the equation $SOAP = SOA + P(rofit)$? How do you go about building, deploying, publishing and orchestrating web services without the (Un)Holy Trinity of SOAP, WSDL and UDDI?

NetKernel : XML Processing for the 21st Century by Brian Sletten

A wise man once said, "XML is like lye. It is very useful, but humans shouldn't touch it." If you've had to incorporate XML into your project by hand, you have probably been burned by getting too close. NetKernel turns this wisdom on its head and encourages you to use XML like the liquid data stream you want it to be. Imagine the simplicity of REST married to the power of Unix pipes. Come see how this open source / commercial product built on a compelling modern architecture can be used to create, manipulate and transform XML.

Effective Teams: The dirty little secret by Bruce Tate

Most conferences will try to tell you that the secret to good software development lies with a process, or a technology, or an architecture. Here's a dirty little secret. You can build working software with an outdated two tier architecture, a waterfall process and COBOL. How? By building a great team.

Java/Ruby Integration with JRuby and ReST by Bruce Tate

You can have rapid web development with Rails without losing access to your critical Java code. With the explosion of the Ruby programming language, more developers will need a strategy for letting Java and Ruby interoperate. This session explores two strategies: JRuby and Rails-based web services.

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

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.

Tapestry In Action (Part One) by Howard Lewis Ship

An introduction to the Apache Tapestry web application framework, which will explain the concepts and features of the framework with some simple applications. We'll discuss Tapestry forms, request cycle, component object model. The use of several important components (including the powerful Table data grid) will be featured.

Tapestry In Action (Part Two) by Howard Lewis Ship

In Tapestry, components are not an add-on; in fact, anything but! Tapestry components are integral to the entire framework # if something dynamic is going on in a page, there's a component involved.

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.

Developing Portlets with Spring Portlet MVC by Mark Fisher

Spring's Portlet MVC framework is one of the major new additions to Spring 2.0, bringing the proven benefits of the servlet-based Spring MVC framework to JSR-168 Portlet development. In this presentation Mark will show you how to use Portlet MVC to develop enterprise portlet applications. The presentation will cover basic portlet architecture, configuration, and deployment. It will also include a walkthrough of a sample portlet application developed using the Spring Portlet MVC framework. Mark will highlight a number of the framework's compelling features as well as its integration with Spring Web Flow.

The Role of Spring in an ESB by Mark Fisher

An Enterprise Service Bus (ESB) brings flow-related concepts such as transformation and routing to a Service-Oriented Architecture. An ESB can also provide an abstraction for endpoints. This promotes flexibility in the transport layer and enables pluggability of POJO services.

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.

Effective Java Persistence Using Spring and iBATIS by Mark Richards

Hibernate has evolved as the de facto standard for persistence in most Spring-based applications. However, many people are turning to iBATIS as an open source persistence alternative for Spring. iBATIS is a powerful open source persistence framework that is rapidly gaining in popularity, particularly within the Spring community. In this session you will learn why iBATIS is becoming so popular, how iBATIS differs from Hibernate and JPA (JSR-220), and how to use iBATIS as the persistence framework for spring-based applications. Through interactive coding examples I will demonstrate how to configure iBATIS within Spring and show how to define mapping files for simple SQL statements, complex SQL statements, and Stored Procedures. I will also discuss the various caching strategies available within iBATIS and the techniques, design strategies, and best practices for using iBATIS in small and large-scale spring-based applications.

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.

Continuous Database Integration by Paul Duvall

Performing daily or continuous builds is essential for ensuring working software. Yet, most consider only the source, not the database, as a part of these builds. What's good for the source code is also good for your database.

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.

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.

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.

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.

Extreme Swing by Romain Guy

With MacOS X and the incoming Windows Vista, rich client applications have never looked so attractive. User interfaces are slowly evolving towards appealing 2.5D and 3D worlds. Although this might seem to be a daunting task, Java offers all the tools you need to let you easily create slick and modern GUI.

Filthy Rich Clients by Romain Guy

Animation and whizzy graphical effects can be totally gratuitous, but they can also be used to make applications more effective and users more productive.

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.

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 giving 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 applications 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.

Effective Hibernate by Scott Leberknight

Hibernate seems simple on the surface yet when you go beyond very simple use cases it can become much more complex. In addition, Hibernate is only part of an overall application architecture. This session shows

ways to use Hibernate effectively including creating rich domain models, managing sessions and transactions, querying for objects, using interceptors and the event model, and handling lazy-loading.

Spring/Hibernate Integration by Scott Leberknight

Hibernate is a very popular Java transparent persistence framework, but you often need to create additional infrastructure to manage sessions, transactions, and lazy-loading in a clean and elegant manner.

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.

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.

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 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.

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.

Java Annotations: From Definition to Consumption by Ted Neward

Want to get the soup-to-nuts story on Java annotations? In this presentation, we'll first talk about what annotations provide to the Java language. After setting ourselves a conceptual basis to operate from, we'll look at the language definition for Java annotations, from how to use them to how to define them. Finally, we'll take a look at the other side of annotations, consuming them at source-level (using "apt", the annotation processing tool), class-level (using a bytecode toolkit such as BCEL), and at runtime (using enhancements to the Reflection API made in Java5).

Java6: Exploring Mustang by Ted Neward

Mustang, the forthcoming Java6 release, is just around the corner, and even if you're not looking to adopt the new platform right away, it's important to know what's there so you can start to plan for it. In this presentation, we'll go over the major new features of the Java6 platform, including the new integrated XML services capabilities (JAX-WS and JAXB), dynamic/scripting language support (javax.script), new JVM "attach" capabilities, new annotations supported by the javac compiler, and more.

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.

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.

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?

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.