

# Greater Nebraska Software Symposium

Crowne Plaza Omaha

April 04 - 06, 2008

<http://www.nofluffjuststuff.com/conference/omaha/2008/04/index.html>

(event schedule as of April 6, 2008)

Fri, Apr. 04, 2008				
	Rose	Dodge I	Dodge II	Dodge III
12:00 - 1:00 PM	REGISTRATION			
1:00 - 1:15 PM	WELCOME			
1:15 - 2:45 PM	JavaServer Faces: A Whirlwind Tour David Geary	Architecture and Scaling Ken Sipe	Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy Scott Davis	Java Concurrency Idioms Alex Miller
2:45 - 3:15 PM	BREAK			
3:15 - 4:45 PM	Facelets David Geary	SOAs Challenges Ken Sipe	Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned- Down Java Developer's Mind Scott Davis	Java Collections API Alex Miller
4:45 - 5:00 PM	BREAK			
5:00 - 6:30 PM	Rich Faces David Geary	7 Habits of Highly Effective Developers Ken Sipe	Groovy, Grails and Google Maps: Mashups 101 Scott Davis	Design Patterns Reconsidered Alex Miller
6:30 - 7:15 PM	DINNER			
7:15 - 8:00 PM	Keynote: by Scott Davis			

Sat, Apr. 05, 2008				
	Rose	Dodge I	Dodge II	Dodge III
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Exploring Terracotta Alex Miller	Java Memory, Performance and the Garbage Collector Ken Sipe	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Grails for Struts Developers: A Groovy Alternative Scott Davis
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	Cluster your Cache with Hibernate and Terracotta Alex Miller	The Busy Java Developer's Guide to ClassLoaders Ted Neward	Iteration 0 Ken Sipe	Seam David Geary
12:30 - 1:30 PM	LUNCH			
1:30 - 3:00 PM	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	The Busy Java Developer's Guide to Debugging Ted Neward	Spring 2.5 - Spring without XML Ken Sipe
3:00 - 3:15 PM	BREAK			
3:15 - 4:45 PM	Boosting Programmer productivity with Mylyn Brian Sam-Bodden	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	The Busy Java Developer's Guide to Monitoring Ted Neward	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSIONS			

Sun, Apr. 06, 2008				
	Rose	Dodge I	Dodge II	Dodge III
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Real World Groovy Scott Hickey	Beginning Object-Relational Mapping with Hibernate Brian Sam-Bodden	Improving Code Quality Nathaniel Schutta	YSlow: Building Your Website for Speed Scott Davis
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	The 90-Minute Startup Michael Nygard	10 ways to use Hibernate effectively Brian Sam-Bodden	JavaScript: the Good, the Bad, and the Ugly Nathaniel Schutta	The Busy Java Developer's Guide to Annotations Ted Neward
12:30 - 1:15 PM	LUNCH			
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION			
2:15 - 3:45 PM	Failures Come In Flavors (part 1) Michael Nygard	Beginning Drools - Rule Engines in Java Brian Sam-Bodden	Designing for Ajax, part 1 Nathaniel Schutta	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward
3:45 - 4:00 PM	BREAK			
4:00 - 5:30 PM	Failures Come In Flavors (part 2) Michael Nygard	Advanced Rules Programming with Drools Brian Sam-Bodden	Designing for Ajax, part 2 Nathaniel Schutta	The Busy Developer's Guide to Scala Ted Neward

# Greater Nebraska Software Symposium

Crowne Plaza Omaha

April 04 - 06, 2008

<http://www.nofluffjuststuff.com/conference/omaha/2008/04/index.html>  
(event schedule as of April 6, 2008)

## **Java Concurrency Idioms by Alex Miller**

This presentation will look at the many new additions in Java 5 and 6 for concurrent programming such as `Atomic`s, `Lock`s, `synchronizers`, and `concurrent collections`. In particular, we will be looking at common concurrency idioms around locking and access to shared state, thread coordination, thread pooling, and work execution. Each of these topics will be presented with code examples demonstrating common idioms and the usage of these new concurrency primitives.

## **Java Collections API by Alex Miller**

Did you know that Java 5 and 6 added 8 new interfaces and 16 new collection implementations to the JDK, more than doubling the size of the collection API? `Collections 201` gives you an update on all of the interfaces, implementations, and utilities and gives you guidance on picking the perfect collection. In particular, Java 5 introduced a new major collection type `Queue` and a whole new `java.util.concurrent` package with data structures optimized for concurrent use.

## **Design Patterns Reconsidered by Alex Miller**

The `Design Patterns` book launched a revolution in object-oriented design and provided a vocabulary for OO developers to communicate their ideas. However, in some cases, patterns used blindly can lead to awkward, confusing, or hard to maintain code. It is time for some common patterns used in Java to be reconsidered so that we can derive the benefits from patterns while minimizing their concerns. This talk will re-evaluate key patterns like `Singleton`, `Template Method`, `Visitor`, and `Proxy`. These patterns have downsides and in some cases, do more harm than good. Examples of each pattern will be given in Java and examined for clarity, testability, and flexibility. Important problems will be discussed and examples of alternate solutions will be given.

## **Exploring Terracotta by Alex Miller**

`Terracotta` is an open-source Java clustering technology. It creates a virtual, durable Java heap that is shared across a cluster of Java Virtual Machines. This is done by dynamically instrumenting bytecode at load time to intercept calls to read and write fields, and also to enter and exit monitor locks. Information about these calls is then transmitted to the `Terracotta Server` (which can also be clustered) and out to other nodes in the cluster as needed. The advantage of this approach is that many Java programs can be clustered without code changes by providing just external `Terracotta` configuration. Many performance optimizations are performed to minimize communication and locking costs. `Terracotta` is commonly used for session sharing in web applications, distributed caching, and distributed workflow processing. This presentation will give an overview of the `Terracotta` technology, how it's implemented, and common use cases that can benefit from the technology. We will look at some code and cluster some Java applications during the presentation.

## **Cluster your Cache with Hibernate and Terracotta by Alex Miller**

`Terracotta` (an open source technology) provides a clustered, durable, virtual heap. You can reduce the load on your database by allowing `Terracotta` to handle sharing and persistence of temporary conversational state in your web application. One option is to simply cluster your existing `Hibernate L2 cache` (for instance with `ehcache`). A higher performance option is to disconnect your `POJOs` from the `Hibernate session` and manage them entirely in `Terracotta shared heap` until they are ready to be written back to the system of record. This option can yield extremely high performance while simultaneously reducing the load on your database, allowing you to scale your system with significantly less hardware.

## **Professional Java UI development with the Eclipse RPC by Brian Sam-Bodden**

Learn how to build featured rich applications using the `Eclipse Rich Client Platform`. The `Eclipse platform` is an open tools platform, on top of this platform you can build your own applications (which do not need to be IDE like or IDE related). Yet you can enjoy the benefits of working with a mature and featured rich platform

that can greatly reduce the amount of time required to create a professional-looking and robust Java UI application.

### **Boosting Programmer productivity with Mylyn by Brian Sam-Bodden**

Mylyn is a task-focused toolkit for the Eclipse IDE that allows developers to focus on tasks in a way that they never been able to do before. Mylyn eliminates the constant context switching produced by typical ways IDEs are used. No more scrolling/browsing/searching/tagging/sending emails with progress updates... Mylyn provides a new way of working that allows you to focus on specific tasks by reducing information overload. Mylyn also provides a framework for integrating with the most commonly usage task tracking systems and version control systems. In this talk you'll learn how Mylyn can boost your productivity as a Java developer by letting you get the most out of your IDE.

### **Beginning Object-Relational Mapping with Hibernate by Brian Sam-Bodden**

Hibernate is an open source Object-Relational Mapping Framework that mostly automates the tedious and time-consuming task of persisting Java objects to a relational database. Hibernate is quickly becoming the preferred way for enterprise developers to overcome the object-relational impedance mismatch and a good alternative to the coarse-grained Entity EJBs, low-level raw JDBC, and by-committee specifications like JDO. Learn what your choices in the ORM arena, what to look for in an ORM tool, and how to get started with Hibernate for your next J2SE or J2EE project.

### **10 ways to use Hibernate effectively by Brian Sam-Bodden**

Learn 10 tried and true ways to improve the way you use Hibernate today. In this session you would learn about a collection of 10 tips, tricks, practices and tools that will make you more effective at designing, implementing, testing and tuning your application's Hibernate-powered object-relational layer.

### **Beginning Drools - Rule Engines in Java by Brian Sam-Bodden**

Drools is an open source pure-Java implementation of a forward chaining rules engine. Drools can be used in a J2SE or J2EE application and allows you to express rules programatically or by building domain specific rule languages. Learn how Business Rules with Drools can make your Java applications more flexible and robust.

### **Advanced Rules Programming with Drools by Brian Sam-Bodden**

In this session you'll learn some of the more advanced features of Drools; a pure-Java Rule Engine. This session will walk through the construction of an advanced Rules application covering such topics as: - Fine control and monitoring of a Working Memory session - Using Decision Tables - Advanced Rule Language Features - Building Domain Specific Languages - Managing your Rules **Prerequisite:** *Beginning Drools*

### **JavaServer Faces: A Whirlwind Tour by David Geary**

In April 2005, annual growth rates for jobs in JavaServer Faces, Struts, and Ruby on Rails were all at about 0%. Today, Struts' growth rate still hovers around 0%, but JSF and Rails have taken off. At the end of 2007, both JSF and Rails were growing at a rate of between 400-500% annually (according to indeed.com). JSF has passed the adoption tipping point, and is now the Java-based framework of choice, as is evidenced by its ecosystem. From vendors such as MyEclipse and RedHat to open source projects such as Seam, Facelets, and Ajax4JSF, JSF is where the action is. Come see why JSF is so popular. In this code- and demo-intensive session, I'll show you the fundamentals of JSF. **Prerequisite:** *Some knowledge of Java-based web applications, such as Struts, is a plus, but is not required. If you have a significant experience with JSF, you probably already know most of what's covered in this session.*

### **Facelets by David Geary**

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet. It's popularity is well deserved, and in fact, much of what is in Facelets today will make its way into the JSF 2.0 spec due out in 2008. So not only can you come to this session and see some really cool demos that you can put to use in the real world, but you'll also be learning JSF 2.0 before it's even been defined! How's that for a ROI? **Prerequisite:** *Some knowledge of JSF is essential. If you're familiar with a templating framework, such as Velocity or Tiles, that's a plus, but not required.*

### **Rich Faces by David Geary**

This talk explores the RichFaces Ajax framework, which is really two frameworks: Ajax4jsf and RichFaces components. In this session you will see how to implement low-level Ajax functionality using Ajax4JSF,

and how to use high-level Ajax components from RichFaces. **Prerequisite:** *Some knowledge of JSF is required, in addition to familiarity with Ajax.*

### **Seam by David Geary**

Have you ever stopped to think that you need to learn two frameworks to develop a non-trivial, database-backed, web application? Struts and iBatis; JSF and Hibernate; Tapestry and EJB3.0. Two frameworks. And then you have to learn to use them together. Why do we have to learn two frameworks just to retrieve "Hello World" from a database and show it in a view. Isn't that crazy? Now you can use one framework, and use one component model. One. Isn't that nice? Seam, a framework built on JSF and EJB3.0, unifies the JSF and EJB component models. Seam is a steam roller, quickly gathering market share among JSF newbies and longtime believers alike. Come see what it's all about. **Prerequisite:** *Some knowledge of JSF is required. If you don't know what a managed bean is, for instance, then attend JSF Whirlwind before this session.*

### **Filthy Rich Clients with the Google Web Toolkit, Part I by David Geary**

The Google Web Toolkit (GWT) is truly a revolutionary framework that lets you develop Ajaxified web applications without knowing anything about Ajax or JavaScript. But the GWT goes way beyond basic Ajax by letting you implement desktop-like applications that run in the ubiquitous browser.

### **Filthy Rich Clients with the Google Web Toolkit, Part II by David Geary**

In the second part of this talk, you will learn how to extend the GWT by implementing custom widgets, including a scrolling viewport and a drag and drop framework. After discussing custom widgets, you will see how to integrate database access into your GWT applications, and how to deploy your GWT applications to external servers.

### **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?

### **SOAs Challenges by Ken Sipe**

SOA... Is it hype? What's real... and what's not? What is the right abstraction level?

### **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.

### **Java Memory, Performance and the Garbage Collector by Ken Sipe**

You are using Java, whew!!! No need to worry about memory, the garbage collector will handle that. Those who have had a memory issue in Java are not so naive any more. Often memory utilization and heap sizes are an after thought and are not recognized until the application is in production, often caused by application uptime, production request volume or production sets of data. When the OutOfMemory Error occurs, often the science of development seems to brake down and knobs are turned. First the (-mx) maximum heap space gets adjusted... More is better right. The next OutOfMemory, heads start scratching, code reviews start in earnest, and Google gets several new hits. Did you know that it is possible to get an OutOfMemory error without running out of heap space?

### **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 certain 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.

### **Spring 2.5 - Spring without XML by Ken Sipe**

Spring 2.5 is brand spanking new, with a number of fantastic features. With growth of large and complex Spring applications which struggle with xml manageability and with the added pressure of Guice and SEAM there is a push for less XML, with solution leaning towards annotations. Spring 2.5 adds to the toolset

provided in Spring 2.0 to provide a development environment where XML is greatly reduced... or eliminated if you so choose.

### **Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence by Ken Sipe**

Well the standards created EntityBeans.... yea. and the community created Hibernate. Fortunately the standards body learned some lessons and created JPA. JPA requires a vendor implementation and none make a better choice than Hibernate. Combined with Spring this trio is a powerhouse when it comes to developer productivity on applications requiring persistence.

### **The 90-Minute Startup by Michael Nygard**

What do you get when you add agile programming, automated deployment, self-describing systems, and virtualization? In this session, Michael will create and deploy a fully-functional web site. By the end of 90 minutes, you will be able to access the site live on the 'Net.

### **Failures Come In Flavors (part 1) by Michael Nygard**

The typical JEE application does not reach the fabled "five nines" of availability. Far from it. It's more like "double eights". Come see why enterprise applications and web sites are only serving users 88% of the time instead of 99.999%. Part 1 of 2

### **Failures Come In Flavors (part 2) by Michael Nygard**

What can we do about the dismal uptime of typical applications? We are asked to provide "five nines", but only reach 88%, on average. Come learn how to prevent the Stability Antipatterns from biting you. Apply these Stability Patterns to contain damage, recover from shocks, and survive disasters. Part 2 of 2

### **Improving Code Quality by Nathaniel Schutta**

It seems that software follows the second law of thermodynamics - in other words, code tends towards disorder. Of course it doesn't have to be that way, and we have a number of tools and techniques that we can apply to keep our code in tip top shape. This talk will discuss ten things you can do to fight back!

### **JavaScript: the Good, the Bad, and the Ugly by Nathaniel Schutta**

Thanks to Ajax, JavaScript is cool again and developers are taking a second look at this much maligned language. This session will give you an overview of this misunderstood language as well as opening your eyes to some of the excellent tools available to ease the pain of developing in this dynamic language.

### **Designing for Ajax, part 1 by Nathaniel Schutta**

So you've convinced the boss that your new web application just has to have Ajax...but now what? With dozens of libraries making even the most blinkish of interactions trivial, how do you decide where to sprinkle the magic Ajax dust? This talk will give a plain old boring "web 1.0" an Ajax facelift with a focus on improving the user experience providing you with a game plan for introducing Ajax to your world.

### **Designing for Ajax, part 2 by Nathaniel Schutta**

We'll pick up where Part 1 left off working in even more advanced approaches such as offline support with Google Gears.

### **Groovy, the Blue Pill: Writing Next Generation Java Code in Groovy by Scott Davis**

There are wild-eyed radicals out there telling you that Java is dead, statically-typed languages are passe, and your skills are hopelessly out-of-date. Those extremists are the same ones who don't bat an eye at throwing out years of experience to learn a new language from scratch, pushing aside a familiar IDE for a new one, and deploying to a whole new set of production servers with little regard to legacy integration. While this "burn the boats" approach to software development might sound exciting to some folks, it's giving your manager the cold shakes right now. What if I told you that there was a way that you could integrate seamlessly with your legacy Java code, continue to use your trusty IDE and stable production servers, and yet take advantage of many of the exciting new dynamic language features that those fanatics keep prattling on about? You'd probably say, "Groovy!" I would, too...

### **Groovy, The Red Pill: Metaprogramming, the Groovy Way to Blow a Buttoned-Down Java Developer's Mind by Scott Davis**

This talk focuses on the ways that Groovy can turn a traditional Java developer's world-view upside down. We'll start by talking about how you can thumb your nose at The Man by leaving out many of the main syntactic hallmarks of Java: semicolons, parentheses, return statements, type declarations (aka Duck-typing), and the ever-present try/catch block. Then we'll look at features like operator overloading and method pointers that Groovy welcomes back into the language with open arms.

### **Groovy, Grails and Google Maps: Mashups 101 by Scott Davis**

Groovy is a new dynamic language that dramatically speeds up Java development. Grails is a complete web framework in a box, including a web server and a database. Google Maps allows you to add maps to your webpage in a few lines of code. Put all three together and you are built for speed.

### **KEYNOTE: How to Lie with Open Source by Scott Davis**

Following in the fine tradition of such books as "How to Lie with Statistics", "How to Lie with Charts", and "How to Lie with Maps", this provocative talk explores our most cherished myths, misunderstandings, and outright lies about Open Source software. Without a firm grasp of the truth, it is difficult to recognize FUD (Fear, Uncertainty, and Doubt) as "Facts Under Duress". And if you don't know the truth, how can you lie about it?

### **Grails for Struts Developers: A Groovy Alternative by Scott Davis**

Struts is the defacto web framework for Java web developers. It has been with us since 2001. Struts enjoys unprecedented success -- most surveys place its market share between 60% and 70%. It introduced a whole generation of web developers to the phrase 'MVC' (Model / View / Controller). Grails 1.0 was released in 2008. It marries the modern features of Rails with the need for legacy support for Struts. Grails is inspired by Rails, but it is not a simple port of the project to Java. It takes the ideas of Rails, but expresses them in familiar Java libraries like Spring and Hibernate. It also leverages a new dynamic language for the JVM called Groovy.

### **YSlow: Building Your Website for Speed by Scott Davis**

How optimized is your website? YSlow, a FireFox/FireBug plugin, doesn't pull any punches. It gives any website an A, B, C, D, or F rating based on 14 individual analysis points. You'll be amazed (or depressed) at what YSlow thinks of your site. In this talk, we'll walk through these points step by step, learning what Yahoo! (the creator of this utility) does to keep its web properties running as quickly as possible.

### **Real World Groovy by Scott Hickey**

Now that Groovy has arrived, you are considering Groovy for your next project. You search the web for more information only to get conflicting or incomplete information. It's hard to distinguish FUD from fact. In this session, we go beyond the theoretical and discuss everything you want to know about using Java in a real world corporate Java project. This presentation is based upon three years of actual experience using Groovy as part of a large J2EE project at a Fortune 500 Insurance company.

### **The Busy Java Developer's Guide to Java Platform Security by Ted Neward**

Permissions, policy, SecurityExceptions, oh my! The Java platform is a rich and powerful platform, complete with a rich and powerful security mechanism, but sometimes understanding it and how it works can be daunting and intimidating, and leave developers with the basic impression that it's mysterious and dark and incomprehensible. Nothing could be further from the truth, and in this presentation, we'll take a pragmatic, code-first look at the Java security platform, including Permissions, the SecurityManager and its successor, AccessController, the Policy class and policy file syntax, JAAS, and more.

### **The Busy Java Developer's Guide to ClassLoaders by Ted Neward**

If you've ever gotten a ClassCastException and just knew the runtime was wrong about it, or found yourself copying .jar files all over your production server just to get your code to run, then you probably find the Java ClassLoader mechanism to be deep, dark, mysterious, and incomprehensible. Take a deep breath, and relax--ClassLoaders aren't as bad as they seem at first, once you understand a few basic rules regarding their operation, and have a bit more tools in your belt to diagnose ClassLoader problems. And once you've got that, and hear about ClassLoaders' ability to run multiple versions of the same code at the same time, and to provide isolation barriers inside your application, or even compile code on the fly from source form, you might just find that you like ClassLoaders after all... maybe.

### **The Busy Java Developer's Guide to Debugging by Ted Neward**

Bugs? We all know your code has no bugs, but someday, you're going to find yourself tracking down a bug in somebody else's code, and that's when it's going to be helpful to have some basic ideas about bug-tracking in your toolbox. Learn to make use of the wealth of tools that the Java Standard Platform makes available to you--tools that your IDE may not know exist, tools that you can make use of even within a production environment.

#### **The Busy Java Developer's Guide to Monitoring by Ted Neward**

Crashes? Outages? Slow response? We all know that it's never your code that causes these things, but for some reason those pesky system administrators still insist on paging you at 4AM to come in and fix those problems, anyway. For some reason, they just keep expecting you to support this thing, even after QA said it was OK!

#### **The Busy Java Developer's Guide to Annotations 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).

#### **The Busy Java Developer's Guide to Hacking (on) the JDK by Ted Neward**

Ever since its 1.1 release, the Java Virtual Machine steadily becomes a more and more "hackable" (configurable, pluggable, customizable, choose your own adjective here) platform for Java developers, yet few, if any, Java developers take advantage of it. Time to take the kid gloves off, crack open the platform, and see what's there. Time to play.

#### **The Busy Developer's Guide to Scala by Ted Neward**

Scala is a new programming language incorporating the most important concepts of object-oriented and functional languages and running on top of the Java Virtual Machine as standard "dot-class" files.