

Central Florida Software Symposium 2005

Sheraton Safari Hotel - Orlando, FL

June 24 - 26, 2005

(session agenda as of 6/22/2005)

Friday, June 24					
	1	2	3	4	5
1:00 - 1:15 PM	Welcome				
1:15 - 2:45 PM	Introduction to Java Reflection Stuart Halloway	Java Metadata Jason Hunter	OpenSource Ecosystems Dave Thomas	Cascading Style Sheets: a Programmer's Perspective Eitan Suez	Making the Most of XML Ben Galbraith
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Design Patterns Revisited: Taking advantage of dynamic, reflective languages Stuart Halloway	Extreme Web Caching Jason Hunter	Introduction to Spring Bruce Tate	XML Data Binding with JiBX Eitan Suez	Creating Polished Swing Applications Ben Galbraith
4:45 - 5:00 PM	BREAK				
4:45 - 6:30 PM	Design Patterns Revisited: Taking advantage of dynamic, reflective languages Stuart Halloway	Forgotten Algorithms Jason Hunter	Lightweight Development Strategies Bruce Tate	Ruby for Java Programmers Dave Thomas	Advanced Swing: Architecture and Frameworks Ben Galbraith
6:30 - 7:30 PM	DINNER				
7:30 - 8:30 PM	Keynote: Dave Thomas (Art in Programming)				

Saturday, June 25					
	1	2	3	4	5
8:15 - 9:00 AM	Welcome and Update - Breakfast				
9:00 - 10:30 AM	Class Loading in Java: Building Dynamic Systems Without Pain Stuart Halloway	The Java Memory Model Brian Goetz	The State Machine Compiler Eitan Suez	SWT Fundamentals Ben Galbraith	Beyond Java Bruce Tate
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Class Loading in Java: Building Dynamic Systems Without Pain Stuart Halloway	An Introduction to XQuery Jason Hunter	Naked Objects Applied Eitan Suez	Advanced SWT and JFace Ben Galbraith	Ruby Persistence Strategies Bruce Tate
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL featuring Stuart Halloway, Dave Thomas, Ben Galbraith, Jason Hunter, Bruce Tate, Brian Goetz and Dion Almaer				
2:15 - 3:45 PM	The Fallacies of Enterprise Systems (Architecture) Ted Neward	XQuery By Example: Advanced Web Publishing Jason Hunter	Creating Killer Graphics and Professional PDFs with XML Ben Galbraith	Ruby on Rails Dave Thomas	Politics of Persistence Bruce Tate
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Introduction to concurrency utilities in JDK 5.0 Brian Goetz	Herding Racehorses and Racing Sheep Dave Thomas	AJAX: Creating Next-Generation, Highly Dynamic, Off-line Capable Web Applications with HTML and JavaScript Ben Galbraith	Runtime Code Generation for Java and Beyond Glenn Vanderburg	Designing and Developing Pluggable Application Architectures David Bock

Sunday, June 26					
	1	2	3	4	5
8:15 - 9:00 AM	Welcome and Update - Breakfast				
9:00 - 10:30 AM	Squashing bugs with FindBugs Brian Goetz	Introduction to Java Server Faces Neal Ford	Effective Enterprise Java: Security Ted Neward	Under the Hood of Java Memory Management Glenn Vanderburg	Building Applications with the Spring Framework Keith Donald
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Give the DB a break!: Performance and Scalability Dion Almaer	Comparison of Java Web Frameworks Neal Ford	Introduction to Web services, 2005 edition Ted Neward	Java Collections Power Techniques Glenn Vanderburg	Advanced Spring: What's New and What You Might Not Know About Keith Donald
12:30 - 1:15 PM	LUNCH				
1:15 - 2:00 PM	BIRDS OF A FEATHER SESSIONS				
2:00 - 3:30 PM	Rules Engines Dion Almaer	Power Regular Expressions in Java Neal Ford	Working with Java Metadata Ted Neward	JavaScript Exposed: There's a Real Programming Language In There! Glenn Vanderburg	Garbage Collection in the HotSpot JVM Brian Goetz
3:30 - 3:45 PM	BREAK				
3:45 - 5:15 PM	Software Metrics and the Great Pyramid of Giza David Bock	Advanced Enterprise Debugging Techniques Neal Ford	Effective Enterprise Architecture Ted Neward	Seaside: A Radical Web Framework Glenn Vanderburg	Enterprise AOP Dion Almaer

Central Florida Software Symposium 2005

Sheraton Safari Hotel - Orlando, FL

June 24 - 26, 2005

(session agenda as of 6/22/2005)

Making the Most of XML by Ben Galbraith

For many of us, XML has become a ubiquitous presence in application development, whether parsing, validating, or manipulating it. For many of us, all that XML is coupled with pain, in the form of tedious APIs (like, say, the W3C DOM API) and confusing technologies (oh, I don't know, W3C XML Schema?).

Creating Killer Graphics and Professional PDFs with XML by Ben Galbraith

You can do some pretty cool things with XML these days (despite what some curmudgeons in the technology world may claim). In the past few years, XML has solidified its place as the lingua franca of data sharing and data manipulation. But XML as a data transfer language is only marginally interesting. Things get really exciting when XML is dynamically transformed into other formats. In this session, I focus on two XML formats which can be readily transformed into high-quality presentation-centric output formats. XSL-FO is a typesetting format for XML that can be readily converted into PDF (or Postscript and some other formats). SVG is a vector graphics language in XML -- a sort of open-source version of the popular Macromedia Flash format. SVG files can be converted into beautiful, completely scalable -- and interactive -- images.

AJAX: Creating Next-Generation, Highly Dynamic, Off-line Capable Web Applications with HTML and JavaScript by Ben Galbraith

As recent high-profile web apps such as Google's GMail have shown, modern browsers are capable of natively rendering web apps with highly dynamic and compelling UIs - fetching server data without page refreshes, animating and manipulating page contents on-the-fly, even offline use. The line between web and "desktop" apps is blurring.

Advanced Swing: Architecture and Frameworks by Ben Galbraith

Are you spending more time plumbing your Swing applications than solving business problems? Has your Swing application grown out of control? This session is for you.

Advanced SWT and JFace by Ben Galbraith

This session picks up where SWT Fundamentals leaves off. Among the advanced topics I discuss are creating custom SWT widgets and exploring tight native integration. I combine another compelling topic with the advanced SWT material: JFace. SWT is a more akin to AWT than Swing; its concerned more with wrapping native functionality than providing any high-level abstractions. JFace is an API on top of SWT that provides such abstractions. The combination of SWT and JFace is comparable to Swing. My coverage of JFace includes an introduction to several of its frameworks, such as the Viewer and Window frameworks, along with many examples. Learning JFace will enable you to write complex SWT applications much faster.

Creating Polished Swing Applications by Ben Galbraith

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; its not very obvious how to make Swing sing.

SWT Fundamentals by Ben Galbraith

The Eclipse project's SWT GUI toolkit provides one of the only viable alternatives to Swing for creating so-called rich client applications in Java. Whereas Swing paints its own widgets and has distinguished itself with a complex (and often obtuse) API, SWT relies on the host operating system for widget rendering and sports a simple, clean API. If your goal is to create a Java application that "looks" like a normal Windows application (or OS X, or Linux), SWT will revolutionize your world. In this session, I introduce SWT from the ground up. I start at a high-level, but quickly move into the details of SWT's API. By the presentation's end, attendees will have a solid understanding of SWT.

Garbage Collection in the HotSpot JVM by Brian Goetz

Pop quiz: which is faster, Java or C++? If you are talking about allocation performance, the answer is Java, hands-down.

Squashing bugs with FindBugs 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 concurrency utilities 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.

Ruby Persistence Strategies by Bruce Tate

Did you ever wonder how people from dynamic languages handle persistence? Here's your chance to see. This session will talk through some important Ruby persistence framework, including basic SQL-driven access (like JDBC), Active Record in Rails, and object relational mapping in Ruby via OG.

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

Beyond Java by Bruce Tate

All programming languages have a limited life span, and Java is no different. This is a philosophical session rather than a programming session. Sooner or later, Java will lose its leadership position. This session will explore Java's strengths and weaknesses. We'll try to understand whether conditions are ripe for alternatives to emerge, and what those alternatives may be.

Lightweight Development Strategies by Bruce Tate

Based on the book Better, Faster, Lighter Java, this beginner to intermediate session will dive into philosophies for lightweight development. It's not a hardcore programming session, but we will talk about process, technologies like Spring, and design patterns like AOP and Dependency Injection. This philosophical session will talk about architectural philosophies rather than low-level programming issues.

Introduction to Spring by Bruce Tate

This session, for the Spring beginner, helps you: # Understand dependency injection and inversion of control # Know the meaning of lightweight containers and Spring # Understand the basic pieces of Spring # See core Spring modules in action, including Persistence, AOP, transactions. Attendees need not know anything about Spring. This session does talk about integration with core J2EE frameworks like JDBC and transactions.

Ruby on Rails by Dave Thomas

The Ruby on Rails framework has exploded onto the scene over the last few months. Propelled by some genuine benefits, and fueled by a whole lot of controversy, Rails seems here to stay. So, is it a Java killer? (No.) Is it a great way to develop certain classes of web application? (Yes.) Does it really deliver the 10-fold increase in developer productivity that some have claimed? (It depends...)

OpenSource Ecosystems by Dave Thomas

Open Source communities produce high quality software with little management and (typically) no pay. Most people looking at open source focus on using this software in their projects.

Ruby for Java Programmers by Dave Thomas

Ruby recently enjoyed its tenth birthday. Instead of cake and candles, the community celebrated by releasing a wave of new libraries and frameworks that make Ruby programming even easier. This talk features some of the best of these, as we explore Ruby.

Herding Racehorses and Racing Sheep by Dave Thomas

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

Designing and Developing Pluggable Application Architectures by David Bock

Pluggable application architectures are everywhere. Applications like Photoshop, Eclipse and other IDEs, and even application servers are all examples of applications that allow other developers to 'install' new functionality. There are plenty of reasons for wanting to install new functionality into an application that is already developed and deployed... from dynamic upgrading to the creation of a 'component marketplace', where end users can purchase components with 'extra' capability.

Software Metrics and the Great Pyramid of Giza by David Bock

Most software engineers hate metrics... Why? Because we know the work we do is hard to quantify # any measurement of 'software engineering' is like trying to tell how tall someone is by how much they weigh... There may be some correlation, but there is so much deviation as to make the answer practically meaningless. As a result, we often see metrics used to justify improper conclusions. There are plenty of good metrics though, and plenty of ways to use them effectively.

Rules Engines by Dion Almaer

Rules engines are powerful beasts which allow you to program in a way in which you specific rules and facts, rather than a linear set of instructions. Learn about how you can use Rules Engines in Java development to take care of complicated problems.

Enterprise AOP by Dion Almaer

Aspect-oriented programming (AOP) has become a hot topic for enterprise development, with recent news of support by IBM, JBoss, BEA, Oracle, Eclipse, and IntelliJ. Behind the news headlines, however, are critical questions: How real is AOP for the enterprise? What problems can it solve today? How does it apply to enterprise applications? How can one make an informed decision about trying to use AOP? What is the best adoption strategy? What are the long term possibilities for AOP in the enterprise? This sessions tries to tackle those questions.

Give the DB a break!: Performance and Scalability by Dion Almaer

What do we really mean by "performance" and "scalability"? This talk gets into the meat of problems which cause our applications to degrade. We will focus on issues such as problems caused by the database being a bottleneck for our application, and see how we can architect our solutions to bypass the issues, resulting in a solid system which scales with the increased load. Not only will we look at the factors, but I will delve into a couple of case studies to show how real world problems were solved!

Naked Objects Applied by Eitan Suez

Join Eitan in this hands-on session on Naked Objects. This session uses the "learning by doing" approach to learning an API or framework. Naked Objects is a powerful tool that can give you a significant advantage in the development of business systems. It gives you the ability to prototype a software application so quickly that it can be performed during information gathering phases of a project. It gives you the power to codevelop the core business model of your application with a non-developer business expert at your side. No prerequisite knowledge of Naked Objects is required.

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.

The State Machine Compiler by Eitan Suez

Classes will often bear various states. Examples include a user who may be "logged in" or "logged out," a bill that is "open" or "paid," or potentially a more complex situation where an object obeys a set of complex rules that determines which of a number of possible states that object is in. The Gang of Four gave us the State Pattern, a fairly straight-forward mechanism for developers to model and implement the behaviour of stateful objects. The State Pattern is only the beginning of the story. Robert Martin developed the State Machine Compiler and has taken the job of developing and maintaining stateful systems to a new level. Today, SMC is a well-maintained open source project hosted on sourceforge.net. Come learn about SMC, a fundamental tool for implementing stateful classes and systems that every software developer should have in his toolchest.

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.

Java Collections Power Techniques by Glenn Vanderburg

The Java Collections framework is a cornerstone of Java development. It's been a part of J2SE for six years now. Every Java developer knows it#how to create Lists, Maps, and Sets, how to put things into them and take things out, and how to iterate over the contents. But there's a lot more to the collections framework than that -- and very few programmers really know how to exploit the power that's just under the surface.

JavaScript Exposed: There's a Real Programming Language In There! by Glenn Vanderburg

Now that web browsers are settling down and Ajax applications are on the rise, it's time to take JavaScript seriously. That means learning it the right way: looking at the fundamentals of the language and surveying its strengths and weaknesses, instead of just copying other people's poorly written examples. This talk takes that approach to JavaScript. It's a "no web pages" introduction to the JavaScript language.

Runtime Code Generation for Java and Beyond by Glenn Vanderburg

Every now and then, it's really helpful to be able to generate a new Java class at runtime. Some problems just can't be solved any other way. It's one of those troublesome tasks: it's fairly tricky to do, and you only need to do it occasionally#but when you need it, you really need it (and usually you need it yesterday). So you have to start essentially from scratch, learning about how to do it on the fly, under pressure.

Under the Hood of Java Memory Management by Glenn Vanderburg

Most of the time, Java's automatic memory management works really well#it's one of the things that makes programming in Java a pleasant and productive experience, and it's nice that we don't have to worry about managing memory manually. However, although it's usually nice to ignore memory management, occasionally we have to pay close attention. Sometimes we need to take control of certain aspects of memory management. Sometimes Java programs do exhibit memory leaks, or unacceptably long garbage collection pauses, or very poor overall performance. But because Java's memory management is supposed to be "fully automatic," it can be difficult to find out what's really going on inside the VM.

Seaside: A Radical Web Framework by Glenn Vanderburg

We've been writing web applications now for 10 years, and they're still no fun. They're awkward and clumsy to write. Internally, they're overly complicated (which almost invariably means that they're buggy). Meanwhile, they're usually too primitive externally. To put it another way: the web programming model is so cumbersome for programmers that the users pay#through reduced features, clumsy interaction, bugs, and poor performance. There's a better way. I know -- who needs another web framework? But Seaside makes even Rails look primitive.

Java Metadata by Jason Hunter

Java's new Metadata facility introduced in J2SE 5.0 defines a way to attach decorations to classes, fields, methods, and even packages that can be extracted by the compiler or runtime tools to provide advanced functionality. Think of metadata as an extended @deprecated flag, or think of XDoclet++. In this tutorial session you'll learn how Metadata fits in the Java platform (and how it compares to the C# platform). We'll cover how to use the metadata attributes provided in the core J2SE libraries and how to write your own. We'll also show a bit of what's coming in JSR-181, tasked to define standard metadata attributes for web services.

Extreme Web Caching by Jason Hunter

Web Caching is very important for high traffic, high performance web site but few people know all the professional-level strategies. In this talk I'll share some of the tricks of the trade, including advanced tips from Yahoo's Mike Radwin. We'll start with the basics: using client-side caches, conditional get, and proxies. Then we'll talk about more advanced features: how best to handle personalized content, setting up an image caching server, using a cookie-free domain for static content, and using randomization in URLs for accurate hit metering or sensitive content.

XQuery By Example: Advanced Web Publishing by Jason Hunter

In this session I'll take some O'Reilly book content (encoded in Docbook XML) and show various ways that the content can be repurposed and made to sing and dance online using XQuery. I'll show several code scripts each less than a page long that do something interesting -- like combine chapters from various books to produce a dynamic table of contents or index, extract figures and graphics, perform targeted search, and

print on demand.

An Introduction to XQuery by Jason Hunter

XQuery is a new language from the W3C that lets you query and manipulate XML -- or anything that can be represented as XML, such as relational databases. As a Java developer -- especially a server-side Java developer -- XQuery is key to searching and manipulating large XML repositories or performing any XML-centric task. This talk introduces XQuery. I'll explain the XQuery language; I'll show how to call XQuery from Java; and as the creator of JDOM, I'll also explain when to use XQuery instead of JDOM, and when to use both.

Forgotten Algorithms by Jason Hunter

There are many interesting and useful algorithms that people just don't remember or never learned. The Boyer-Moore string search algorithm is one prime example. The randomized skip list is another. Both solve common problems with wonderful flair and finesse -- and performance-wise they blow the pants off brute force solutions. This session covers these two algorithms plus several others. It's like your college algorithms course but with a practical bent and absolutely zero proofs. Extra bonus: The Google PageRank algorithm.

Advanced Spring: What's New and What You Might Not Know About by Keith Donald

Spring 1.2 is out--Spring 1.3 is right on the horizon. As a broad, user-driven project with a large community, the newest releases offer a wealth of new features to be taken advantage of. This session focuses on demonstrating the most important, and how you can start leveraging them in your projects immediately.

Building Applications with the Spring Framework by Keith Donald

You'll see how to use Spring to assemble a complex system from a set of focused, loosely-coupled components. You'll experience through example how Spring enables agile development by allowing you to start simple, validate architectural choices early, and scale up as requirements demand.

Introduction to Java Server Faces by Neal Ford

This session introduces experienced Java web developers to the new framework on the block and talks about the major features of JSF. It also compares JSF to other frameworks along the way.

Advanced Enterprise Debugging Techniques by Neal Ford

Out, out, damn bugs! This session discusses techniques and tools for debugging enterprise applications.

Power Regular Expressions in Java by Neal Ford

This session shows how to fully exploit regular expressions in Java.

Comparison of Java Web Frameworks by Neal Ford

This session provides an objective comparison of Struts, Tapestry, JavaServerFaces, and WebWork.

Introduction to Java Reflection by Stuart Halloway

Reflection is writing code that manipulates itself. Well-written reflective code automates a broad class of repetitive, error-prone programming tasks. Poorly-written reflective code obfuscates programs and destroys the benefits of the type system. We'll focus on the former.

Design Patterns Revisited: Taking advantage of dynamic, reflective languages by Stuart Halloway

(3 Hour Session) Attendees should attend the Introduction to Reflection talk, or have some experience using reflection or metaprogramming in a reflective language such as Java, Objective-C, Smalltalk, Python, or Ruby. Familiarity with the GOF book is helpful but not required. Design patterns are recurring solutions to problems that consistently appear in software development. However, this does not mean that design patterns cannot be "solved", i.e. converted into language or library features. In fact, most of the original design patterns can be solved using dynamic language features such as reflection.

Class Loading in Java: Building Dynamic Systems Without Pain by Stuart Halloway

(3 Hour Session) One of Java's greatest strengths is its flexible deployment model. In this session you will learn how Class Loaders facilitate deployment, and how to troubleshoot Java and J2EE Class Loading problems.

Effective Enterprise Architecture by Ted Neward

Bring all of your enterprise Java questions to this open forum discussion hosted by the author of #Effective Enterprise Java#, Ted Neward.

Working with Java Metadata by Ted Neward

As part of JDK 1.5, Java has introduced a facility for developers to create and use custom metadata annotations, as developed by the JSR 175 committee. This represents a radical new shift for the Java programming language, quite possibly larger and farther-reaching than generics or any other language feature.

Effective Enterprise Java: Security by Ted Neward

Security's become a hot topic among enterprise developers in recent years, but to many developers, security is still the white elephant in the middle of the room. Discussions about security usually begin with, "Uh, we'll worry about that later", or, "Start with two really large prime numbers.....". Security isn't as hard as developers make it out to be, but it is something that developers need to face and recognize.

The Fallacies of Enterprise Systems (Architecture) by Ted Neward

There's a set of fallacies that every enterprise developer has fallen for at some point in their enterprise development lives, and unless they've come to realize it early enough, all cause big trouble and painful learning experiences in the long run.

Introduction to Web services, 2005 edition by Ted Neward

WSDL, and Schema and SOAP, oh my! It's 2005, and the Web services landscape looks even more confusing than it did two years ago, despite all sorts of promises to the contrary. What's it all mean, and how the heck did we get here when the original goal was to try and keep it all simple?