



aosd.05

International Conference on
Aspect-Oriented Software Development

Conference Programme

Table of Contents

| | |
|---|----|
| Silver Sponsors..... | 3 |
| Bronze Sponsors..... | 3 |
| Academic Sponsors..... | 3 |
| Site Map..... | 6 |
| Keynotes..... | 7 |
| Grady Booch | 7 |
| Dave Thomas | 7 |
| Technical Programme..... | 8 |
| Wednesday, March 16, 2005..... | 8 |
| Thursday, March 17, 2005..... | 8 |
| Friday, March 18, 2005..... | 9 |
| Invited Industry Talks..... | 10 |
| Adopting AOP..... | 10 |
| What's new in AspectJ 5..... | 10 |
| AspectWerkz 2 and the Road to AspectJ 5..... | 10 |
| AOP with Metadata: Principles and Patterns..... | 11 |
| Comparison of the Leading AOP tools..... | 11 |
| AOP in Spring..... | 11 |
| Application Security Aspects..... | 11 |
| Implementing Middleware Using AOP..... | 12 |
| Object Persistence Aspects..... | 12 |
| AOP with C++..... | 12 |
| VM-Level AOP..... | 13 |
| Practitioner Reports..... | 14 |
| Applying AspectJ to J2EE Application Development..... | 14 |
| Using AOP for Enterprise Auditing of J2EE Applications..... | 14 |
| Using Aspect Orientation To Restructure A Model-Driven Development Framework..... | 14 |
| Demos..... | 16 |
| D1: Using AspectJ to Build a Product Line for Mobile Devices..... | 16 |
| D2: Developing Embedded Software Product Lines with AspectC++..... | 16 |
| D3: Aspect-Oriented Programming with Caesar..... | 17 |
| D4: Developing Dynamic and Adaptable Applications with CAM/DAOP: A Virtual Office Application..... | 17 |
| D5: PROSE - A middleware platform for dynamic adaptation..... | 17 |
| D6: AspectJ Development Tools (AJDT)..... | 18 |
| D7: Context Aspect-Sensitive Service Container..... | 18 |
| D8: Aspects of Rich Accessibility Experiences..... | 19 |
| D9: ActiveAspect: An Interactive Aspect Modeling Tool..... | 19 |
| D10: C-SAW and GenAWeave: Tools to Assist in Weaving Across Multiple Artifacts at Different Levels of Abstraction..... | 20 |
| D11: ASAAM-T: A Tool environment for Identifying Architectural Aspects..... | 20 |
| D12: Magellan: Program Comprehension and Analysis in Eclipse..... | 21 |
| D13: CoSMIC: Addressing Crosscutting Deployment and Configuration Concerns of Distributed Real-time and Embedded Systems via Aspect-oriented & Model-driven Software Development..... | 21 |
| D14: Aspect composition with ObjectTeams/Java in Eclipse..... | 21 |
| D15: Microsoft Phoenix: A Framework for Software Analysis and Optimization..... | 22 |
| Exhibitions..... | 23 |
| AspectWerkz Aspect Container..... | 23 |
| AspectC++ Add-In for Microsoft® Visual Studio® | 23 |
| AspectJ and AJDT..... | 23 |
| AspectMentor..... | 24 |
| Spring..... | 24 |
| Introducing AOPHP (Aspect-Oriented PHP)..... | 24 |
| Persistence as an Aspect in Oracle TopLink..... | 24 |
| Panel..... | 25 |
| Birds of a Feather Sessions (BOF)..... | 25 |
| Committees..... | 26 |

Silver Sponsors



BEA Systems, Inc. (Nasdaq: BEAS) is a world leader in enterprise infrastructure software, helping enabled companies to improve business responsiveness through service-oriented architecture (SOA), a software design approach that more closely aligns IT with business objectives. With 15,000 customers worldwide including the majority of the Fortune Global 500, BEA and its WebLogic® and Tuxedo® brands are among the most trusted names in business technology. Headquartered in San Jose, Calif., BEA has 75 offices in 36 countries. More information on BEA products and services is available at www.bea.com.

TheServerSide.COM

Your Enterprise Java Community


TheServerSide.com is an online community for enterprise Java architects and developers, providing daily news, tech talk interviews with key industry figures, design patterns, discussion forums, satire, tutorials, and more. TheServerSide.com was launched in May 2000 and has become the largest independent Java community in the world. TheServerSide Communities consist of TheServerSide.NET, TheServerSide.com, and TheServerSide Symposiums. TheServerSide.NET is a sibling community to TheServerSide.com focused on enterprise development with the Microsoft .NET framework. TheServerSide Symposium is an exclusive technical conference focused on cutting edge enterprise development issues.

Bronze Sponsors



Academic Sponsors



| Schedule | Monday | | | | Tuesday | | | | Wednesday | | | | Thursday | | | | Friday | | | |
|----------|-------------|-----------|----------|-----------------------|-------------|-----------|----------|-----------------------|--|--|---|--|---|--|--|--|---|--|---|--|
| | Research | | Industry | | Research | | Industry | | Research | | Industry | | Research | | Industry | | Research | | Industry | |
| 08:30 am | | | | | | | | | Opening | | | | | | | | | | | |
| 08:45 am | | | | | | | | | Demo Madness | | | | | | | | | | | |
| 09:00 am | T1 | Tutorials | ACP4IS | Workshops | T6 | Tutorials | DAW | Workshops | Keynote (Grady Booch) The Complexity of Programming Models <i>Grand Ballroom</i> | | | | Keynote (Dave Thomas) Transitioning AOSD from Research Park to Main Street <i>Grand Ballroom</i> | | | | Panel Challenges and Opportunities for Industry Adoption <i>Grand Ballroom</i> | | | |
| 09:30 am | T2 | | AOM | | T7 | | EA | | | | | | | | | | | | | |
| 10:00 am | | | FOAL | | | | SPLAT | | Coffee Break | | | | Coffee Break | | | | | | | |
| 10:30 am | | | LATE | | | | WTAOP | | Coffee Break | | | | Coffee Break | | | | | | | |
| 11:00 am | | | | | | | | | Paper Session 1 Modularity <i>Grand Ballroom</i> | | Invited Talks <i>Toledo Ballroom</i> | | Paper Session 3 Languages and Implementations <i>Grand Ballroom</i> | | Invited Talks <i>Toledo Ballroom</i> | | Paper Session 6 Analysing and Testing <i>Grand Ballroom</i> | | Invited Talks <i>Toledo Ballroom</i> | |
| 11:30 am | | | | | | | | | | | | | | | | | | | | |
| 12:00 pm | | | | | | | | | | | | | | | | | | | | |
| 12:30 pm | | | | | | | | | | | Exhibitions <i>Empire Ballroom</i> | | | | Exhibitions <i>Empire Ballroom</i> | | | | | |
| 01:00 pm | Lunch Break | | | | Lunch Break | | | | Lunch Break | | | | Lunch Break | | | | | | | |
| 01:30 pm | | | | | | | | | | | | | | | | | | | | |
| 02:00 pm | T3 | Tutorials | ACP4IS | Workshops – continued | T9 | Tutorials | DAW | Workshops – continued | Paper Session 2 Aspects in Middleware and System Software <i>Grand Ballroom</i> | | Invited Talks <i>Toledo Ballroom</i> | | Paper Session 4 Refactoring and Reengineering <i>Grand Ballroom</i> | | Invited Talks / Practitioner Reports <i>Toledo Ballroom</i> | |  | | | |
| 02:30 pm | T4 | | AOM | | T10 | | EA | | | | | | | | | | | | | |
| 03:00 pm | T5 | | FOAL | | | | SPLAT | | | | | | | | | | | | | |
| 03:30 pm | | | LATE | | | | WTAOP | | | | | | | | | | | | | |
| 04:00 pm | | | | | | | | | | | | | | | | | | | | |
| 04:30 pm | | | | | | | | | | | | | | | | | | | | |
| 05:00 pm | | | | | | | | | | | | | | | | | | | | |
| 05:30 pm | | | | | | | | | | | | | | | | | | | | |
| 06:00 pm | | | | | | | | | | | | | | | | | | | | |
| 06:30 am | | | | | | | | | Welcome Reception | | | | BOFs | | | | | | | |
| 07:30 pm | | | | | | | | | | | | | | | | | | | | |
| 08:00 pm | | | | | | | | | BOFs / Extravaganza | | | | Banquet | | | | | | | |
| ... pm | | | | | | | | | | | | | | | | | | | | |

(Lobby Level)

- Valencia East** T1 *abc: A Workbench for Aspect-Oriented Programming Language Research*
- Valencia West** T2 *Aspect-Oriented Programming with AspectJ (1) : Introduction*
- Valencia East** T3 *Aspect-Oriented Programming with AspectJ (2) : Hands-on Session*
- Valencia West** T4 *Practical Aspect-Oriented Programming with JBoss: The Case For Aspects*
- St. Clair** T5 *Why Should Aspect Enthusiasts Care About Software Architecture?*
- Valencia East** T6 *Enterprise Aspect-Oriented Programming with AspectJ*
- Valencia West** T7 *Feature Oriented Programming*
- T8 **(CANCELLED)** *Aspect-Oriented Programming with AspectWerkz*
- Valencia East** T9 *Aspect-Oriented Programming with C++ and AspectC++*
- Valencia West** T10 *Aspect-Oriented Refactoring: Taking Refactoring to a New Level*

- ACP4IS *Aspects, Components, and Patterns for Infrastructure Software*
- AOM *Aspect-Oriented Modeling*
- FOAL *Foundations Of Aspect-Oriented Languages*
- LATE *Linking Aspect Technology and Evolution*
- DAW *Dynamic Aspects Workshop*
- EA *Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design*
- SPLAT *Software Engineering Properties of Languages and Aspect Technologies*
- WTAOP *Workshop on Testing Aspect-Oriented Programs*

- Wright**
- Burnham**
- Sullivan**
- Sullivan**
- Holabird**
- Burnham**
- Holabird**
- Wright**
- Sullivan**

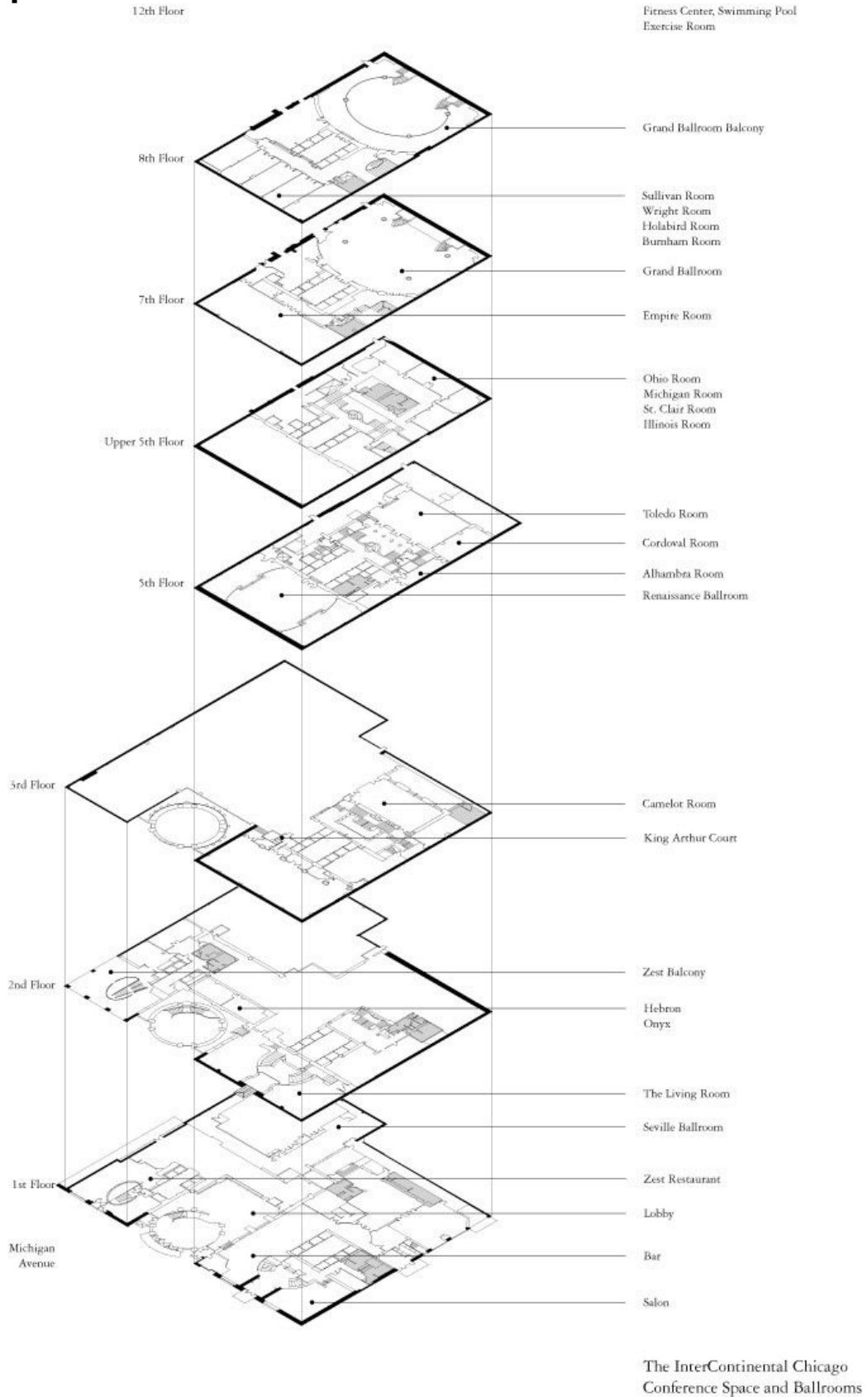
| Schedule | Monday | Tuesday | Wednesday | | Thursday | | Friday | |
|----------|--------|---------|----------------------------------|----------------------------------|----------------------------------|--------------|----------------------------------|----------------------------------|
| | | | (Lobby Level) Valencia A Room | (Lobby Level) Valencia B Room | (Lobby Level) Valencia A Room | | (Lobby Level) Valencia A Room | (Lobby Level) Valencia B Room |
| 08:30 am | | | | | | | | |
| 08:45 am | | | | | | | | |
| 09:00 am | | | | | | | | |
| 09:30 am | | | | | | | | |
| 10:00 am | | | | | | | | |
| 10:30 am | | | | | | Coffee Break | | Coffee Break |
| 11:00 am | | | Coffee Break | | Demo 8 | | Demo 13 | Demo 6 |
| 11:30 am | | | Demo 5 | Demo 9 | Demo 6 | | | Demo 11 |
| 11:45 am | | | Demo 1 | Demo 12 | Demo 5 | | Demo 15 | Demo 1 |
| 12:00 pm | | | | | | | | |
| 12:30 pm | | | | | | | Demo 3 | Demo 2 |
| 01:00 pm | | | Lunch Break | | Lunch Break | | | |
| 01:15 pm | | | | | | | Demo 14 | Demo 10 |
| 01:30 pm | | | | | | | | |
| 02:00 pm | | | Demo 7 | Demo 3 | Demo 7 | | | |
| 02:30 pm | | | | | | | | |
| 02:45 pm | | | Demo 4 | Demo 14 | Demo 4 | | | |
| 03:00 pm | | | | | | | | |
| 03:30 pm | | | Demo 8 | | | Coffee Break | | |
| 04:00 pm | | | | | | | | |
| 04:30 pm | | | Coffee Break | | Demo 12 | | | |
| 05:00 pm | | | Demo 13 | Demo 10 | Demo 9 | | | |
| 05:15 pm | | | Demo 15 | Demo 2 | Demo 11 | | | |
| 05:30 pm | | | | | | | | |
| 06:00 pm | | | | | | | | |
| 06:30 am | | | | | | | | |
| 07:30 pm | | | | | | | | |
| 08:00 pm | | | | | | | | |
| ... pm | | | | | | | | |

**AOSD.05
(Demos)**

- D1 *Using AspectJ to Build a Product Line for Mobile Devices*
- D2 *Developing Embedded Software Product Lines with AspectC++*
- D3 *Aspect-Oriented Programming with Caesar*
- D4 *Developing Dynamic and Adaptable Applications with CAM/DAOP*
- D5 *PROSE - A middleware platform for dynamic adaptation*
- D6 *AspectJ Development Tools (AJDT)*
- D7 *Context Aspect-Sensitive Service Container*
- D8 *Aspects of Rich Accessibility Experiences*
- D9 *ActiveAspect: An Interactive Aspect Modeling Tool*
- D10 *C-SAW and GenAWeave: Tools to Assist in Weaving Across Multiple Artifacts at Different Levels of Abstraction*

- D11 *ASAAM-T: A Tool environment for Identifying Architectural Aspects*
- D12 *Magellan: Program Comprehension and Analysis in Eclipse*
- D13 *CoSMIC*
- D14 *Aspect composition with ObjectTeams/Java in Eclipse*
- D15 *Microsoft Phoenix: A Framework for Software Analysis and Optimization*

Site Map



Keynotes

All keynotes will be held in:

Grand Ballroom

Grady Booch

Wednesday, March 16, 9:30 am – 11:00 am

IBM Fellow

The Complexity of Programming Models

A considerable amount of development occurs both at the edge of systems as well as in the spaces between systems. Building such applications requires that the development team have in place a conceptual programming model (encompassing all the languages and protocols against which they cut code) as well as a very concrete one (as manifest in the semantics of the interfaces against which they cut code). A fundamental problem for enterprise systems - and even for embedded systems as standards mature - is that most programming models are incredibly complex. This talk will examine what may be done to mitigate that complexity and what role aspect-oriented development may play.

Dave Thomas

Thursday, March 17, 9:00 am – 10:30 am

Bedarra Research Labs, Carleton University and University of Queensland

Transitioning AOSD from Research Park to Main Street

AOSD is motivated by the important need to reduce the complexity observed in numerous large programs where various concerns are tangled into code that is difficult to comprehend hence risky to repair or enhance. The field builds on pioneering contributions in OO including computational reflection, role modeling, subject oriented programming, separation of concerns and program transformation. AOSD has moved from research lab to practical tools supporting industrially popular languages and platforms. Recently AOSD has found early adopters in commercial application and product development. Aspects have arrived at the brink of the chasm!

In this talk we discuss the opportunities and challenges for AOSD to cross the chasm. What are the compelling benefits for commercial users? How does AOSD impact the water fall or agile software process? Where do Aspects fit in the education curricula? Where does AO belong in the tool chain and/or runtime? What are the limitations in current theory, usability and implementation which limit or retard their use?



Technical Programme

All technical papers are presented in:

Grand Ballroom

Wednesday, March 16, 2005

11:30 am - 12:30 pm Session 1: Modularity

Modularizing Design Patterns with Aspects: A Quantitative Study

Alessandro Garcia, Cláudio Sant'Anna, Eduardo Figueiredo, Uirá Kulesza, Carlos Lucena, Arndt von Staa, *Computer Science Department, PUC-Rio*

An Analysis Of Modularity in Aspect Oriented Design

Cristina Videira Lopes and Sushil Krishna Bajracharya, *University of California, Irvine, USA*

02:00 pm - 04:00 pm Session 2: Aspects in Middleware and System Software

An expressive aspect language for system applications with Arachne

Rémi Douence, Fritz Thomas, Nicolas Lorient, Jean-Marc Menaud, Marc Ségura-Devillechaise, and Mario Südholt, *EMN/INRIA Obasco group*

Empowering Configurable QoS Management in Real-Time Systems

Aleksandra Tesanovic, Mehdi Amirijoo, Mikael Björk, and Jörgen Hansson, *Linköping University*

Supporting Autonomic Computing Functionality via Dynamic Operating System Kernel Aspects

Michael Engel and Bernd Freisleben, Distributed Systems Group, *Dept. of Mathematics and Computer Science, University of Marburg*

Towards Just-in-time Middleware Architectures

Charles Zhang, Gilbert Gao, and Hans-Arno Jacobsen, *University of Toronto*

Thursday, March 17, 2005

10:30 am - 12:00 am Session 3: Languages and Implementations

Adaptive Programming in JasCo

Wim Vanderperren, Davy Suvée, Bart Verheecke, María Agustina Cibrán, and Viviane Jonckers, *Vrije Universiteit Brussel*

abc: An Extensible AspectJ Compiler

Pavel Avgustinov, *University of Oxford, United Kingdom*, Aske Simon Christensen, *University of Aarhus, Denmark*, Laurie Hendren, *McGill University, Montreal*, Sascha Kuzins, *University of Oxford, United Kingdom*, Jennifer Lhotak, *McGill University, Canada*, Ondrej Lhotak, *McGill University, Canada*, Oege de Moor, *University of Oxford, United Kingdom*, Damien Sereni, *University of Oxford, United Kingdom*, Ganesh Sittampalam, *University of Oxford, United Kingdom*, Julian Tibble, *University of Oxford, United Kingdom*

What Does Aspect-Oriented Programming Mean to Cobol?

Ralf Lämmel, *Free University, Amsterdam*, Kris De Schutter, *Ghent University*

02:00 pm - 03:30 pm Session 4: Refactoring and Reengineering

Towards a Catalog of Aspect-Oriented Refactorings

Miguel Monteiro, *Escola Superior de Tecnologia, Instituto Politécnico de Castelo Branco*, João Fernandes, *Departamento de Informática, Universidade do Minho*

Deriving Refactorings for AspectJ

Leonardo Cole and Paulo Borba, *Informatics Center, Federal University of Pernambuco, Brazil*

Role-Based Refactoring of Crosscutting Concerns

Jan Hannemann, Gail Murphy, and Gregor Kiczales, *University of British Columbia*

04:00 pm - 05:00 pm Session 5: Aspects in User-centric Processes

Weaving a Social Fabric into Existing Software

Li-Te Cheng, John Patterson, Steven Rohall, Susanne Hupfer, and Steven Ross, *IBM Research*

Mylar: a degree-of-interest model for IDEs

Mik Kersten and Gail Murphy, *University of British Columbia*

Friday, March 18, 2005

11:00 am - 12:00 am Session 6: Analysis and Testing

Crosscutting Techniques in Program Specification and Analysis

Patrick Lam, Viktor Kuncak, and Martin Rinard, *MIT Computer Science and Artificial Intelligence Laboratory*

Aspect Language Features for Concern Coverage Profiling

Hridesh Rajan and Kevin Sullivan, *University of Virginia*

Distributing Classes with Woven Concerns – An Exploration of Potential Fault Scenarios

Nathan McEachen and Roger Alexander, *Colorado State University*



Invited Industry Talks

All invited talks will be held in:

Toledo Room

Adopting AOP

Wednesday, March 16, 11:00 am – 12:00 pm

Gregor Kiczales, *University of British Columbia, lead PARC team that invented AspectJ*
Adrian Colyer, *IBM, AspectJ and AJDT project leader*

Abstract

Everyone's talking about AOP, but where do you start when first introducing it to a project, team, or organization? This talk presents a roadmap for adopting aspect-oriented programming that has successfully been used by a number of teams and projects. The phases of adoption will be described, together with examples of team organization, tool sets, and of course aspects at each phase. By the end of the talk, you will have a pragmatic strategy for getting starting with AOP right away.

What's new in AspectJ 5

Wednesday, March 16, 2:00 pm – 2:30 pm

Jonas Bonér, *BEA Systems, AspectWerkz founder*
Adrian Colyer, *IBM, AspectJ and AJDT project leader*

Abstract

The AspectJ and AspectWerkz teams recently announced that they would be joining forces to work on the development of "AspectJ 5", the forthcoming AspectJ release. This release will contain support for the Java language features introduced by Java 5, an annotation-based development style, and enhanced load-time weaving support. In this session Adrian and Jonas will give an overview and demonstration of these new features and explain how existing users of AspectJ and AspectWerkz can take advantage of them.

AspectWerkz 2 and the Road to AspectJ 5

Wednesday, March 16, 2:30 pm – 3:30 pm

Jonas Bonér, *BEA Systems, AspectWerkz founder*

Abstract

This talk gives an overview of AspectWerkz 2. In it, we describe how AspectWerkz 2 has been designed to meet the requirements of enterprise application development in terms of integration, dynamicity, performance and ease of use. Each of these requirements will be discussed, considering the benefits, drawbacks and challenges.

In the light of the recent merger between AspectWerkz and AspectJ, special attention will be given to the features which are being brought to the AspectJ 5 platform, as well as discussing why some are not being brought in, and how some are being redesigned.

The talk will also cover the concept of AspectWerkz's extensible aspect container, which allows multiple 'aspect models' (like AspectWerkz, AspectJ, Spring aspects etc.) to co-exist in the same execution

environment with consistent semantics, as well its support for pluggable weaving strategies.

AOP with Metadata: Principles and Patterns

Wednesday, March 16, 3:30 pm – 4:00 pm

Ramnivas Laddad, *Author AspectJ in Action*

Abstract

Aspect-oriented programming (AOP) and metadata make a synergistic combination. AOP presents itself as a principled producer and consumer of metadata. Metadata enhances AOP's crosscutting mechanism by allowing it to capture join points based on program element's annotations. It also makes AOP much more accessible by simplifying pointcut definitions that utilize annotations.

As good as this combination is, it carries a potential for overuse and misuse, undermining the obliviousness principle. Fully leveraging metadata-fortified AOP requires a systematic understanding of metadata from AOP's perspective and implications of employing metadata-driven crosscutting. This best practices talk will present expressing multidimensional interfaces using metadata and employing associated design patterns to create loosely decoupled system.

Comparison of the Leading AOP tools

Wednesday, March 16, 5:30 pm – 6:00 pm

Mik Kersten, *University of British Columbia, AspectJ and AJDT committer*

Abstract

AOP buzz is continuing to increase, and more tools are becoming available. It is less clear how developers should choose which tools to evaluate, and what tradeoffs those tools will impose on their projects. This talk outlines the state of the art in AOP tools, overviews the most mature approaches, and contrasts the adoption related issues of each.

AOP in Spring

Thursday, March 17, 11:00 am – 12:00 pm

Rod Johnson, *CEO Interface21, Spring Framework founder*
Adrian Colyer, *IBM, AspectJ and AJDT projects*

Abstract

This talk gives a brief introduction to the Spring lightweight container and its support for dependency injection. Spring's approach to AOP will be presented, along with examples of some of the aspects that come pre-packaged with Spring. Spring also integrates well with AspectJ, and examples will be given of how to configure and use AspectJ aspects with Spring. We will discuss when it is appropriate to use Spring's AOP framework, and when to use AspectJ aspects. The talk will conclude with a look at some ongoing developments to make the use of both Spring AOP aspects and AspectJ aspects as seamless as possible in the Spring container.

Application Security Aspects

Thursday, March 17, 12:00 pm – 12:30 pm

Ron Bodkin, *New Aspects of Software*

Abstract

Application security is becoming increasingly important topic, and as application complexity continues to increase, systematic enforcement of this crosscutting requirement is becoming critical.

AOP provides a mechanism to separate policy from implementation and to provide confidence in correct implementation. It allows expressing access control in terms of uniform rules instead of scattering access

checks through out a code base. However, there are also challenges in writing effective pointcuts and concerns about how AOP affects security policies.

We examine the benefits and risks of using AOP for application security, by looking at examples of using it for:

- enforcing role-based access control, by checking permissions wherever necessary
- data-level access control, in which access to resources is based on the relationship of a user to the data. For example, only an employee's manager can look at the employee's salary, not any manager in the company.
- filtering displays, by limiting the display of information or controls based on user entitlement
- auditing sensitive operations whenever access is made

Implementing Middleware Using AOP

Thursday, March 17, 2:00 pm – 3:00 pm

Bill Burke, *Chief Architect, JBoss Inc.*

Abstract

This talk will discuss how JBoss Inc. is using aspect-oriented programming to implement its middleware products. The first part of the presentation focuses on how JBoss Cache is using dynamic AOP for its implementation as well as a visual demo on the software. The talk will also drill down into how AOP is being used to drive JBoss' EJB 3.0 implementation.

Object Persistence Aspects

Thursday, March 17, 4:00 pm – 5:00 pm

Jim Clark, *Oracle Corporation*

Abstract

We don't create Objects just to persist them in a database (at least, not usually). We create Objects to model the behavior of a system. The need to persist the state of these Objects is "orthogonal". In that sense, Object persistence has always been an aspect of any system (ie even before we started explicitly talking about AOP). In this presentation, we will be looking at GetField/PutField JoinPoints, and at what kinds of advice can be used to enhance an existing Persistence Framework. We will also look at what sorts of state introductions are typically necessary for a persistent Object.

AOP with C++

Friday, March 18, 11:00 am – 12:00 pm

Olaf Spinczyk, *Assistant Professor, Friedrich-Alexander University, Erlangen-Nuremberg, Germany*

Abstract

The C++ programming language still dominates commercial software development in many application domains. Although mature AOP solutions for C++ would be desirable, most professional AOP tools only support aspect weaving in Java code. Does the complexity of the C++ language spoil the party? Could language-independent AOP tools do the job? Or is C++ simply powerful enough to support AOP without additional tools?

Olaf Spinczyk will address these questions and present the conclusions, which motivate the AspectC++ project. AspectC++ is an AspectJ-like AOP extension for C++. The AspectC++ development tools are aimed to bring fully-fledged AOP support into the C++ world. The talk will briefly introduce the AspectC++ language and development tools. It will also discuss the strengths and weaknesses of the current aspect weaver implementation based on real-world examples from industry and academia.

VM-Level AOP

Friday, March 18, 12:00 pm – 1:00 pm

Sam Pullara, *EIR, Accel Partners*

Abstract

Current AOP implementations in Java do their work by modifying the bytecode of the classes at the pointcuts that are specified in the particular AOP language being used or are higher level language constructs. There are several problems with this including safety, side-effects, dynamism, and performance that can be mitigated by moving the AOP machinery into the VM. I will discuss what hooks would be needed and how those hooks would affect current AOP implementations.



Practitioner Reports

All practitioner reports are presented in:

Toledo Ballroom

Applying AspectJ to J2EE Application Development

Wednesday, March 16, 4:30 pm

Nicholas Lesiecki, *VMS, Coauthor Mastering AspectJ*

Abstract

This report focuses on the application of AspectJ to the development of a J2EE web application for Video Monitoring Services of America (VMS). Aspects were used to cleanly modularize concerns ranging in scope from auxiliary (error-handling) to application-specific (shopping basket price calculation) to framework-level (object relationship management). VMS saw benefits resulting from the aspect-oriented implementation of these concerns in the areas of code size, understandability, and reduced defects. This report will detail specific areas to which AspectJ was applied, the development team's reaction to the new technology, strategies employed to ease adoption, and some of the pitfalls encountered when using the development tools.

Using AOP for Enterprise Auditing of J2EE Applications

Thursday, March 17, 15:00 pm

Matt Wizeman, *Chief Scientist, Near Infinity Corporation*
Jeff Kunkle, *Chief Operating Officer, Near Infinity Corporation*

Abstract

This report describes the use of aspect-oriented techniques to develop an enterprise auditing solution targeted for J2EE applications. Aspect-oriented programming allows us to create drop-in modules that apply consistent auditing rules across all applications in the enterprise. In this particular implementation, AspectJ was used to create a commercial product, named IntelliPrints™, that does runtime weaving of audit code for Servlet, JNDI, EJB, and JDBC API code.

Using Aspect Orientation To Restructure A Model-Driven Development Framework

Thursday, March 17, 17:00 pm

Sreedhar Reddy, *Tata Research Development and Design Centre*
Vinay Kulkarni, *Tata Research Development and Design Centre*

Abstract

We have a model-driven development framework that consists of a model repository, a set of model-based code generators and an integrated development environment. Over the years, it has been extensively used to deliver several large enterprise class business applications. Our code generators take UML class models as input and generate code for various architectural layers such as presentation layer, application logic layer, messaging layer and persistence layer in various technology platforms. There is a separate code generator for each of these layers for a technology platform. We also generate code for various design strategies such as auditing, concurrency management, logging etc. These strategies cut across several architectural layers.

As a result, change in an existing design strategy or addition of a new design strategy required modifications to multiple code generators in multiple places. Ensuring a consistent implementation of such a change required thorough knowledge of all the concerned tools on part of a tool implementer. This problem grew especially severe as the number of variants of the code generators grew requiring a large development team.

We have since restructured our code generators using aspect orientation techniques as a set of composable building blocks with each building block addressing a specific aspect such as persistence, presentation, messaging, auditing etc. A building block specification consists of three parts – model instantiation, model-to-code transformation and code weaving. The code generation process is carried out in three phases – model instantiation phase where models instantiated by various building blocks are composed, model-to-code transformation phase where model is translated to code and code weaving specifications are generated, and a code weaving phase where the generated code fragments are woven.

Use of aspect orientation has significantly reduced our tool maintenance effort leading to quick turnaround times with a much smaller team size. The building block abstraction has enabled reuse across toolset variants. A toolset variant can now be composed as a configuration of reusable building blocks. However, we faced some implementation challenges along the way. Some aspects had circular dependencies i.e. application of one aspect created model elements which made another aspect a candidate for application, which in turn created model elements that made the first aspect a candidate for re-application. We solved this problem by using 'iterate-till-saturate' approach. This was sufficient for our needs, but we feel the issue needs to be addressed in a more comprehensive manner. We found code weaving around a function call in terms of before / after advice was not adequate in some cases, for example in database queries. We solved the problem, in this limited context, by building a model for the query and performing aspect weaving at the model level. We believe a richer join point model is needed to address this problem in a more comprehensive manner. Another problem that still remains to be addressed is how aspect-orientation can be used to decompose test specifications. At present, a tester needs to be aware of all the aspects.



Demos

D1: Using AspectJ to Build a Product Line for Mobile Devices

Trevor Young & Gail Murphy, *University of British Columbia*

Wednesday, March 16, 12:00 am, (Lobby Level) Valencia A Room

Friday, March 18, 12:00 am, (Lobby Level) Valencia B Room

Building a mobile application that can run on a variety of micro devices is difficult. Sun's Java 2 Micro Edition (J2ME) technology provides a set of profiles and configurations that lets a developer target a wide range of devices. Many vendors that support J2ME on their devices also offer specific APIs to access functionality specific to their hardware.

Including functionality in an application that takes advantage of the characteristics of every device that is targeted can become prohibitively large for hardware that has memory and storage constraints. Developing a product-line to deploy across all of these devices can be hard to manage. Aspect-oriented programming techniques can make the task of building such a product-line easier.

Attendees will see a demonstration of how we used AspectJ & Apache ANT to build different versions of a product-line. In our approach, optional features are written as aspects. Selected feature aspects can then be woven with the core application code to create a final program that is customized for a targeted device. The talk will present a number of issues we encountered using AspectJ to develop mobile applications, and discuss design patterns and best practices to use when writing aspect-oriented J2MEbased mobile applications. During the demo we will build a prototype Photo Album application and run it on a variety of mobile device emulators from cell phones to palm pilots, each one with different optional features woven in as aspects.

D2: Developing Embedded Software Product Lines with AspectC++

Olaf Spinczyk & Daniel Lohmann, *Friedrich-Alexander University Erlangen*

Wednesday, March 16, 05:15 pm, (Lobby Level) Valencia B Room

Friday, March 18, 12:30 pm, (Lobby Level) Valencia B Room

AspectC++ is a general purpose aspect-oriented language extension to C++. It is aimed to bring fully-fledged AOP support in areas with strong demands on runtime efficiency and code density. This makes it possible to exploit the power of AOP for the domain of (deeply) embedded systems, where computation speed and available memory resources are strictly limited. AOP concepts are particularly useful for the development of scalable embedded system product lines. This will be demonstrated by a real world example: a small embedded device equipped with meteorological sensors and an 8-bit micro-controller running AspectC++ code.

By covering the complete build cycle of configuration, compilation and installation, participants will understand how easy it is to integrate AspectC++ with an existing tool chain. A presentation of the AspectC++ Add-Ins for common IDEs, like Eclipse and Visual Studio, rounds up the demonstration and shows that mature tool support for programming in AspectC++ is now available.

D3: Aspect-Oriented Programming with Caesar

Ivica Aracic & Vaidas Gasiunas & Klaus Ostermann, *Technische Universität Darmstadt*

Wednesday, March 16, 02:00 pm, (Lobby Level) Valencia B Room

Friday, March 18, 12:30 pm, (Lobby Level) Valencia A Room

CaesarJ is a new aspect-oriented programming language which addresses the most important goals of software design: modularity, reuse, flexibility, and correctness. It is easy to learn because it is based on object-oriented concepts and fully integrates with the Java programming language and Eclipse IDE. All new language features are compiled to efficient Java byte-code.

In comparison to other AO languages, CaesarJ provides unique support for reusable aspects, variability management, feature-oriented programming, and flexible aspect deployment. Technically, this is enabled by CaesarJ's support for virtual classes, family polymorphism, propagating mixin composition, and extensible aspect deployment strategies on top of an AspectJ-like pointcut language.

Attendees of the demonstration will see a live example that illustrates the expressiveness gained by the features of CaesarJ. We will show the new modularity and reusability features by means of a small software product line. Flexible aspect deployment will be illustrated by deployment strategies for thread-local and remote aspects. We will also present our Eclipse plug-in, which provides IDE integration and powerful visualization features for the new language concepts.

CaesarJ can be downloaded at <http://caesarj.org/>

D4: Developing Dynamic and Adaptable Applications with CAM/DAOP: A Virtual Office Application

Mónica Pinto Alarcón & Daniel Jiménez Priego & Lidia Fuentes Fernández, *University of Málaga*

Wednesday, March 16, 02:45 pm, (Lobby Level) Valencia A Room

Thursday, March 17, 02:45 pm, (Lobby Level) Valencia A Room

CAM/DAOP is a component and aspect based model and platform implemented using Java/RMI and reflective techniques. On the one hand, CAM (Component-Aspect Model) is a new model to design component and aspect based applications specified in UML, which defines the main entities of a CAM application and the relationships among them. On the other hand, the DAOP platform is a distributed component and aspect platform that implements the CAM model and provides a dynamic weaving mechanism that plugs aspects into components at runtime. Another relevant feature of this approach is the use of DAOP-ADL, an XML-based architecture description language that is used to describe the structure of CAM applications in terms of a set of components, aspects and composition constraints. The information generated with DAOP-ADL is used at runtime by the DAOP platform that consults it to obtain the composition rules that indicate how to perform the dynamic composition of components and aspects. Using CAM/DAOP we have developed several collaborative applications, where the most relevant one is a Virtual Office application, which allows dispersed users to collaborate as if they were co-located.

Attendees of the demonstration will see how to develop Dynamic and Adaptable Component and Aspect Based Applications with CAM/DAOP. The demo will go through the different phases involved in the development of the Virtual Office Application: (1) the application design using the CAM model and an UML editor; (2) the description of the application architecture using the DAOP-ADL language; (3) the reuse of pre-implemented components and aspects; (4) the execution of the application by the DAOP platform using the information provided with DAOP-ADL, and (5) the possibility to dynamically adapt the behavior of the application at runtime by modifying the application architecture described in the DAOP-ADL language.

D5: PROSE - A middleware platform for dynamic adaptation

Angela Nicoarã & Gustavo Alonso, *Swiss Federal Institute of Technology Zürich*

Wednesday, March 16, 11:30 am, (Lobby Level) Valencia A Room

Thursday, March 17, 12:00 am, (Lobby Level) Valencia A Room

PROSE (PROgrammable extenSions of sERVICES), an open source project (<http://prose.ethz.ch/>), is an

adaptive middleware platform based on Java for dynamic AOP which allows aspects to be woven, unwoven, or replaced at run time. Woven into a program, an aspect may change several units of functionality. Aspects are expressed in the same source language as the application (Java). PROSE provides middleware tools that allow run time monitoring of remote aspects. The PROSE workbench for weaving, unweaving and querying aspects is a monitoring tool which displays information about what aspects are currently inserted in the system, what join-points belong to each aspect, and what the join-points of the system are. Embedded into an Eclipse plug-in, this tool is also used to developing and running PROSE applications from Eclipse and allows run time monitoring of remote aspects, and provides a creation wizard that helps to create PROSE aspects.

In this demonstration we present PROSE and its associated middleware tools. We show the flexibility of PROSE and how it has been embedded into Eclipse, and present a variety of examples of run time adaptation. In the first part of the demonstration we show how to modify the behavior of a running application by dynamically weaving an aspect that identifies places of interest in the application. After each weaving operation, we show that the running application has been adapted and we use the PROSE middleware tools to graphically display the join-points matched by the woven aspect. In the second part of the demonstration we show how easy is to create aspects that modify a program at run time using the creation wizard of the PROSE Development Tools for Eclipse. When an aspect is created, the user can select the PROSE crosscut types and pointcutters from a list. Then the code template for a PROSE aspect that defines an advice method and a pointcut method will be generated. The aspects are then woven remotely into different VMs. After the weaving operation, we show that the running application has been adapted. In case of two running applications, we show that the aspects can be cut and pasted between distinct VMs using the PROSE workbench and also can be moved between distinct VMs by drag and drop using the PROSE plug-in for Eclipse. We also show how to insert and withdraw aspects transactionally. We start a transaction and then weave and unweave aspects into/from different VMs.

Aspect insertions and withdrawals won't be activated until the transaction is committed. The transaction can also be aborted, leaving the application as if no insertion or withdrawal ever took place.

D6: AspectJ Development Tools (AJDT)

Matt Chapman & Andy Clement, IBM Hursley

Thursday, March 17, 11:30 am, (Lobby Level) Valencia A Room

Friday, March 18, 11:00 am, (Lobby Level) Valencia B Room

The AspectJ Development Tools for Eclipse project (AJDT) aims to provide first class tooling for developing applications with AspectJ. In addition to IDE support on a par with Eclipse's Java Development Tools, AJDT provides sophisticated tools for visualizing and navigating the crosscutting nature of your aspect-oriented applications.

During the session we'll look at how to use aspects with existing Java applications, as well as exploring functionality that can also be used with new AspectJ projects. We'll be focusing on the AJDT v1.2 release where there have been significant improvements to provide better integration with Eclipse 3 and new and upcoming features will be showcased. These include better views and UIs for viewing crosscutting associations, content assist, advice ordering, inheritance hierarchy crosscutting, control-flow advice, and crosscutting- centric project exploration. This release of AJDT has itself been built with AspectJ and we'll show how this has helped to improve code quality, serviceability, add performance and resource monitoring capabilities, and enable us to support multiple versions of Eclipse.

D7: Context Aspect-Sensitive Service Container

Thomas Cottenier & Tzila Elrad & Andrew Prunicki, Illinois Institute of Technology

Wednesday, March 16, 02:00 pm, (Lobby Level) Valencia A Room

Thursday, March 17, 02:00 pm, (Lobby Level) Valencia A Room

Many of the shortcomings of current Web Service composition mechanisms stem from the difficulty of defining, modularizing and managing service behavior that is dependent on the context of service invocation.

The Contextual Aspect-Sensitive Service (CASS) model proposes a novel service composition methodology, based on dynamic service refinement through contextual aspect weaving. Contextual aspects encapsulate the distributed structure and the collaborative behavior of composite web services, as well as invocation context

dependent behavior.

The CASS container avoids the need for a centralized orchestration engine and enables on-demand service instance customization. Its service collaboration- based specification language offers a powerful alternative to static and centralized business process specification languages.

This demonstration highlights the capability of the CASS container to instantiate distributed contextual aspects on-the-fly. First, the demonstration shows how service orchestration structure and dynamic behavior are expressed in the CASS specification. Second, the context- dependent service refinement capability is illustrated by a highly dynamic search engine application in which service orchestration is dynamically reconfigured according to the client context. Finally, the demonstration points out the more advanced capabilities of the CASS model and tools and shows how complex service interaction patterns such as mobile agent-based behavior are implemented and specified in the container.

D8: Aspects of Rich Accessibility Experiences

Alison Lee & T.V. Raman, IBM TJ Watson Research Center

Wednesday, March 16, 03:30 pm, (Lobby Level) Valencia A Room

Thursday, March 17, 11:00 am, (Lobby Level) Valencia A Room

Providing rich user experiences to users with diverse abilities frequently requires modifying the application code base or maintaining multiple versions. An alternative approach is to view accessibility as a cross-cutting concern of usable and accessible software systems. This makes the task of developing accessible software both viable and manageable. Furthermore, when accessibility development is performed using an extensible and open application framework, developers can leverage the combined capabilities to facilitate the development of accessible software.

Our demonstration will illustrate three contributions of using aspects with Eclipse. First, we will show how accessibility enhancements (e.g., magnification, speech-enablement) can be added to an application like the Eclipse IDE. These same enhancements are also available for Eclipse-based rich- client applications such as the Lotus Workplace Activity Explorer. Instead of requiring a specialized version of each application, users can simply load the accessibility add-on plug-ins into the run-time environment and configure their preferences for the combination of speech and visual enablements they desire. The accessibility add-on uses aspects as well as Eclipse's frameworks to access the application context and render information in aural and visual form. Second, we will show how core functionality of the Eclipse framework can be extended with additional functionality for enriched accessibility and usability. In particular, we demonstrate how an "incremental find" capability was added to all tree widgets in Eclipse and Eclipse rich-client applications. Third, we will demonstrate two approaches for adding an additional pane such as a magnifier pane to all Eclipse perspectives that provide layouts for different tasks such as Java development or code archive browsing.

This is intended to contrast the benefits of using aspects combined with Eclipse's rich frameworks compared to simply using an Eclipse framework.

D9: ActiveAspect: An Interactive Aspect Modeling Tool

Wesley Coelho & Gail C. Murphy, University of British Columbia

Wednesday, March 16, 11:30 am, (Lobby Level) Valencia B Room

Thursday, March 17, 04:30 pm, (Lobby Level) Valencia A Room

Developers working with aspect-oriented programs can benefit from tool support that indicates how crosscutting concerns influence the underlying program. Existing tools provide features such as gutter annotations to indicate the presence of crosscutting, outline views that list the program elements advised by a particular advice, and an abstract visual representation of the lines of code advised by one or more aspects. These tools provide useful information but do not provide an overview of an aspect's effect on the program. We will present an aspect modeling tool that produces an interactive graphical model of an aspect in an existing AspectJ program. The model is displayed using a modified UML notation that has been extended to support AOP constructs and member-to- member relationships. Models are initially constructed by displaying the direct influence of an aspect such as the static shadows of its advice and the inter- type members it declares. This initial model can then be automatically expanded to show additional context such as calls made to introduced members or from advice bodies. Unique user interaction and automated abstraction techniques

are employed to limit the model's complexity as additional context is incrementally added. We believe this model-based approach to investigating aspects has the potential to improve developer productivity.

D10: C-SAW and GenAWeave: Tools to Assist in Weaving Across Multiple Artifacts at Different Levels of Abstraction

Jeff Gray & Suman Roychoudhury & Jing Zhang & Yuehua Lin, *University of Alabama at Birmingham*

Wednesday, March 16, 04:30 pm, (Lobby Level) Valencia B Room

Friday, March 18, 01:15 pm, (Lobby Level) Valencia B Room

The C-SAW and GenAWeave tools support evolution of legacy software through a two-level approach using aspects. The principle strategy of these tools is to generate low-level transformation rules from higher-level domain languages. The demonstration will show the ability to make rapid adaptations to a large cross-section of an application through simple specification changes at a high-level of abstraction. As case studies, the demonstration will highlight the transformation of two legacy commercial applications: a large mission-computing avionics framework written in C++, and a client-server enterprise management system implemented in Object Pascal. In the avionics application, transformation rules are generated from domain-specific models created in the Generic Modeling Environment (from Vanderbilt University). Using C-SAW, it will be shown that small changes in a representative model can regulate concurrency and logging policies across many C++ classes. The Object Pascal portion of the demonstration will illustrate the use of a program transformation system (the Design Maintenance System) as the underlying engine for an aspect weaver. A unique feature of the demonstration is the ability to weave aspects into various legacy languages (not just Java) at the source level using GenAWeave.

D11: ASAAM-T: A Tool environment for Identifying Architectural Aspects

Bedir Tekinerdoğan & Frank Scholten, *University of Twente*

Thursday, March 17, 05:00 pm, (Lobby Level) Valencia A Room

Friday, March 18, 11:30 am, (Lobby Level) Valencia B Room

ASAAM-T is a tool environment that implements the Aspectual Software Architecture Analysis Method (ASAAM), which is a scenario-based method for identifying architectural aspects. Like conventional software architecture analysis methods, ASAAM starts with characterizing scenarios that can be directly supported by the architecture (direct scenarios) and scenarios that require the redesign of the architecture (indirect scenarios). Some scenarios however can be scattered over different architectural components and tend to be very difficult to localize their impact in individual components. ASAAM introduces a set of heuristic rules to identify these so-called aspectual scenarios, and to derive architectural aspects. Based on the detailed impact analysis for a given set of scenarios, ASAAM also provides a categorization of the architectural components into cohesive components, composite components, tangled components and ill-defined components. The results of the detailed impact analysis can be used in aspect-oriented design and aspect-oriented programming.

ASAAM-T is implemented in the Eclipse Java platform and includes a tool for specifying the software architecture, a tool for specifying reusable scenarios, a tool for guided selection of scenarios, a tool for impact analysis of the scenarios on the given architecture, a tool for identification of architectural aspects, and finally a tool for categorization of the architectural components. Attendees of the demonstration will see a step-by-step evaluation of a software architecture for an industrial case and observe how the architectural aspects are identified using scenarios. The demonstration will be organized in three parts. The first part includes the specification of the architecture and the scenarios. The second part includes the guided selection of scenarios from a reusable scenario base. The third part of the demonstration will evaluate the impact of selected scenarios on the various architectural components and identify architectural aspects. We will also demonstrate different selections of scenarios and show the impact on the architecture and the architectural aspects.

D12: Magellan: Program Comprehension and Analysis in Eclipse

Michael Eichberg & Thorsten Schäfer, *Technische Universität Darmstadt*

Wednesday, March 16, 12:00 am, (Lobby Level) Valencia B Room

Thursday, March 17, 04:00 pm, (Lobby Level) Valencia A Room

Magellan is a query-based, extensible platform for software comprehension integrated with Eclipse. All artefacts of a project, e.g., Java code, deployment descriptors, properties files, etc., are transformed into a uniform XML-based representation and stored into a database; XQuery is used to explore the database connecting interrelated pieces of information, which are spread around different artefacts.

Using Magellan with an EJB project e.g., enables to retrieve methods declared with container-based transaction management, which create user transactions in their code: hence, violating the EJB specification. The information needed to retrieve such methods, crosscuts the artefact structure of an EJB project. Magellan lifts such crosscutting structures into modular units: the results of executing a query get integrated into one view and visualized using a graph, which can further be navigated through. This allows exploring a software project as a net of interconnected information entities, bridging the gap that exists when using trees and multiple independent views for browsing through a software project.

You will see a live demonstration of how to use pre-defined Magellan queries to unveil and explore crosscutting views of a project. Furthermore, we will demonstrate how to extend Magellan's capabilities by defining new project-specific queries. Magellan can be downloaded from: <http://www.st.informatik.tu-darmstadt.de/static/pages/projects/Magellan/Magellan.html>

D13: CoSMIC: Addressing Crosscutting Deployment and Configuration Concerns of Distributed Real-time and Embedded Systems via Aspect-oriented & Model-driven Software Development

Aniruddha S. Gokhale & Arvind S. Krishna & Doug Schmidt, *Vanderbilt University*

Wednesday, March 16, 04:30 pm, (Lobby Level) Valencia A Room

Friday, March 18, 11:00 am, (Lobby Level) Valencia A Room

The Component Synthesis using Model Integrated Computing (CoSMIC) is a Model-Driven Development (MDD) tool suite used to address key lifecycle crosscutting challenges, which includes deployment and configuration issues, of Distributed Real-time and Embedded (DRE) systems. CoSMIC consists of an integrated collection of modeling, analysis, and synthesis tools that address the crosscutting challenges of large-scale DRE systems deployed and configured using contemporary component middleware, such as lightweight CORBA Component Model.

The CoSMIC demo will provide attendees with a key understanding of the challenges and solutions for modeling crosscutting lifecycle concerns in DRE systems. The attendees will see a live demo of the CoSMIC MDD technology using a hypothetical DRE system as a guiding example. A key focus of the CoSMIC demo is the description of modeling abstractions to separate concerns that are scattered across various configuration files, such as the XML descriptions typically encountered in middleware deployment and configuration. CoSMIC generative and run-time tools are used, respectively, to synthesize these artifacts and then weave them into the deployed system.

D14: Aspect composition with ObjectTeams/Java in Eclipse

Stephan Herrmann, *Technische Universität Berlin*

Carsten Pfeiffer & Jan Wloka, *Fraunhofer FIRST*

Wednesday, March 16, 02:45 pm, (Lobby Level) Valencia B Room

Friday, March 18, 01:15 pm, (Lobby Level) Valencia A Room

The Object Teams Development Tooling (OTDT) is an integrated development environment (IDE) based on Eclipse. The OTDT provides support for practitioners who wish to apply the ObjectTeams/Java (OT/J)

programming language for aspect oriented software development. At the core of this IDE a fully incremental compiler for OT/J performs separate compilation of aspects and regular classes. The Object Teams Runtime System weaves aspects at load time for the benefits of enhanced flexibility and easier configuration management. Specialized editors and structural views give full productivity support for all new language constructs and concepts.

The demonstration will show the relevant concepts of Object Teams and the OTDT. We will walk through a szenario of integrating generalized, reusable aspects into an existing application. Structural browsing will be presented for achieving a sufficient understanding of the application and of the aspect code. A table based editor will be used for defining an additional module serving as a connector between core and aspect. This editor supports all concepts of OT/J relating to aspect integration at the levels of classes (role-base binding), methods (callin and callout bindings) and parameters (parameter mappings). For launching the integrated application a configuration technique is presented which allows to decide at program start which aspects should be included.

D15: Microsoft Phoenix: A Framework for Software Analysis and Optimization

John Lefor, *Microsoft Research*

Wednesday, March 16, 05:15 pm, (Lobby Level) Valencia A Room
Friday, March 18, 11:45 am, (Lobby Level) Valencia A Room

Phoenix is the next generation optimization and analysis framework currently under development as a joint project between Microsoft Research and the Microsoft Developer Division. Phoenix provides a framework for analysis and code generation for any .Net language as well as being extensible to support input in a variety of other forms (AST's, binaries ...). The structure of the Phoenix framework makes it useful for the implementation of language neutral weavers and could readily be adopted to offer additional opportunities for software analysis. Being based in the .Net platform Phoenix is language agnostic and offers a variety of extensibility models to adapt to novel problem areas.

Attendees of the demonstration will get a quick overview of the Phoenix architecture and extensibility model and will see compiler plug-ins and software analysis applications built using the Phoenix framework operating on conventional C# or C++ applications. We will review the source code for some simple plug-ins and see how they can be used to modify the compilation process and we will look at an application which is a version of the compiler backend modified to offer a deeper understanding of what happens during the code generation and optimization process.



Exhibitions

Wednesday, March 16, 12:30 pm - 02:00 pm, Empire Ballroom
Thursday, March 17, 12:30 pm - 02:00 pm, Empire Ballroom

AspectWerkz Aspect Container

Alexandre Vasseur & Jonas Bonér, BEA Systems

AspectWerkz is a pure Java, lightweight and high-performant AOP framework for Java. AspectWerkz has been innovating in several fields since its inception, with annotation and XML defined aspects, annotation driven AOP, atomic hot- deployment/undeployment of aspects, application server integration etc. It has been successfully implementing recent AOSD research ideas like runtime and load time weaving, and "just in time" compilation of join points. AspectWerkz is available under an open source license and is backed by BEA Systems.

This exhibition gives the attendees an opportunity to see AspectWerkz in action and discuss with AspectWerkz leads about the internals of their choice. The features presented aims at giving the practitioner a deep view on AspectWerkz programming model based on Java annotations, weaving capabilities (post compilation, load time and runtime) and application server integration.

Simple samples will provide an entry point to understand AspectWerkz plain Java aspects programming model, while more complex samples based on a broad scope of Java/J2EE technologies will illustrate how to unleash AOP in real world Java and J2EE applications.

A specific emphasis will be put on annotation-driven AOP that latest Java 5 enables in a standardized way, in which Java annotations can raise the abstraction level of the pointcut language and in which AOP allows adding behavior to the static metadata provided by the annotations.

AspectC++ Add-In for Microsoft® Visual Studio®

Danilo Beuche, pure-systems

On the AOSD exhibition pure systems presents the new version of its AspectC++ Add-In for Microsoft® Visual Studio®, now bundled with AspectC++ 1.0. The pure-systems AspectC++ Add-In is the first integrated solution for aspect-oriented software development with C++, the most widely used programming language in the world. As an Add-In it was specially designed for Microsoft® Visual Studio®, the world most popular integrated development environment.

pure-systems provides services and creates tools for aspect-oriented programming and software product line development. The company puts a special focus on the introduction of those technologies into the development of embedded software.

AspectJ and AJDT

Adrian Colyer, IBM UK

Come and meet the developers of the AspectJ language and compiler, and the AJDT (AspectJ Development Tools) for Eclipse. We'll be on hand to discuss the latest features in AspectJ 5 and in AJDT, and to demonstrate the tools. Learn about AspectJ's support for generics and annotations, see AJDTs deeper integration into the Java tools suite, and catch some of the latest demos.

AspectJ is a seamless aspect-oriented extension to the Java programming language that is Java platform

compatible and designed to be easy to learn and use. AspectJ is also an open source project hosted at eclipse.org: <http://www.eclipse.org/aspectj>. AJDT provides integration of the AspectJ compiler and language into the Eclipse IDE. It provides syntax highlighting, code assist, integrated building, structure views, crosscutting views and more. AJDT is also an open-source project on eclipse.org. You can find out more at <http://www.eclipse.org/ajdt>.

AspectMentor

Gregor Kiczales (founder) & Ron Bodkin & Mik Kersten & Ramnivas Laddad & Nicholas Lesiecki & Mira Mezini

AspectMentor is a group of experts in aspect-oriented programming (AOP) and aspect-oriented software development. We help organizations adopt and use aspect-oriented technologies with services including training, mentoring, project planning and strategy development.

We'll be available in the exhibition to talk about adopting AOSD, our services, and to demonstrate our work. We will be showing training materials from our courses, code from projects we've worked on, and signing books that we've written.

Spring

Rod Johnson, CEO Interface21, Founder Spring Framework
Rob Harrop, Lead Architect Cake Solutions, Spring Framework committer

Introducing AOPHP (Aspect-Oriented PHP)

John W. Stamey, Jr. & Bryan T. Saunders & Don Yessick, Coastal Carolina University; Matthew Cameron, SOI, Inc.

A platform that has not yet enjoyed a successful implementation of aspect-oriented programming (AOP) is PHP. With recent improvements in PHP's Object-Oriented functionality, implemented in PHP 5.0, this popular middleware is an ideal platform in which to implement AOP. Previous approaches for implementing aspects in PHP using C extensions have yet to be successful. A design implementing aspects using pre-processing has yielded a successful solution for implementing aspects in PHP.

We present our approach for implementing aspect-oriented PHP, along with some discussion about features to be available in the upcoming release of AOPHP V2.0. Released under the GNU General Public License in December 2004, AOPHP may be found on the World-Wide Web at <http://www.aopphp.net/>.

Persistence as an Aspect in Oracle TopLink

Shaun Smith, Principal Product Manager, Oracle TopLink
Merrick Schincariol, Senior Engineer, Oracle AS Containers for J2EE, Oracle Corp
Jim Clark, OracleAS Solution Architect, Oracle Corp

Oracle TopLink is the leading Java object-to-relational persistence framework which provides a highly flexible and productive mechanism for storing Java objects and Enterprise Java Beans (EJBs) in relational database tables. TopLink offers developers excellent performance and choice, working with any database, any application server, any development toolset and process, and any J2EE architecture.

In the past, TopLink has achieved a high level of transparency in its persistence by relying exclusively on reflection. However, AOP offers new ways to achieve the same or better level of transparency while opening up opportunities for even better performance. We will show how persistence mapping meta-data can be combined with AOP to simplify the development and improve the performance of applications using object-relational mapping. We will demonstrate how meta-data can be used to identify join points that can be enhanced with persistence advice to implement lazy loading and change tracking, how persistence semantics (e.g., read-only/read-write) can be enforced in persistent objects, and how factoring out persistence as an aspect results in a simpler programming model for developers.



Panel

Friday, March 18, 09:00 am - 10:30 am, Grand Ballroom

Challenges and Opportunities for Industry Adoption

Moderator: Don Almaer

Participants: Adrian Colyer, Bill Burke, Jonas Boner, Danilo Beuche, Rod Johnson

Birds of a Feather Sessions (BOF)

Birds of a Feather Sessions provide an informal forum for discussions related to a particular community or topic. No registration is required for BOFs. Rooms will be made available for BOF sessions on Wednesday and Thursday evening. A sign-up sheet will be placed near the registration desk, where attendees can reserve rooms on a first-come first-served basis.

Committees



Programme

- Mehmet Aksit, University of Twente, Netherlands
- Jonathan Aldrich, Carnegie Mellon University, USA
- Brian Barry, Bedarra Research Labs, Canada
- Shigeru Chiba, Tokyo Institute of Technology, Japan
- Siobhan Clarke, Trinity College, Ireland
- Yvonne Coady, University of Victoria, Canada
- Adrian Colyer, IBM Hursley Park, UK
- Erich Gamma, IBM Zurich, Switzerland
- Erik Hilsdale, Google, USA
- Viviane Jonckers, Vrije Universiteit Brussel, Belgium
- Gregor Kiczales, University of British Columbia, Canada
- Günter Kiesel, University of Bonn, Germany
- Shriram Krishnamurthi, Brown University, USA
- Karl Lieberherr, Northeastern University, USA
- Ralf Laemmel, Vrije Universiteit Amsterdam, The Netherlands
- Doug Lea, SUNY Oswego, USA
- Hidehiko Masuhara, University of Tokyo, Japan
- Gail Murphy, University of British Columbia, Canada
- Linda Northrop, Software Engineering Institute, USA
- Harold Ossher, IBM Research, USA
- Klaus Ostermann, Darmstadt University of Technology, Germany
- Awais Rashid, University of Lancaster, UK
- Martin Robillard, McGill University, Canada
- Christa Schwanninger, Siemens, Germany
- Mario Südholt, École des Mines de Nantes, France
- Kevin Sullivan, University of Virginia, USA
- Peri Tarr, IBM Watson Research Center, USA (**Chair**)
- Jianjun Zhao, Fukuoka Institute of Technology, Japan

Conference

- General Chair: Mira Mezini (Darmstadt University of Technology)
- Program Chair: Peri Tarr (IBM Watson Research Center)
- Organizing Chair: Tzilla Elrad (Illinois Institute of Technology)
- Tutorials and Workshops Chairs: Hidehiko Masuhara (University of Tokyo), Klaus Ostermann (Darmstadt University of Technology)
- Demonstrations Chair: Christa Schwanninger (Siemens AG)
- Industry Chairs: Ron Bodkin (New Aspects of Software), Ramnivas Laddad
- Web Chair: Doug Janzen (University of British Columbia)
- Publicity Chair: Eric Eide (University of Utah)
- Student Extravaganza Chairs: Mik Kersten (University of British Columbia), Kris De Volder (University of British Columbia)
- BOFs Chair: Mik Kersten (University of British Columbia)