

# Welcome



## Conference Chair: John Vlissides, *IBM Research*

It's a pleasure to welcome you to OOPSLA 2004, the 19th annual conference on object-oriented programming, systems, languages, and applications. OOPSLA is the premier forum for practitioners, researchers, and students in diverse disciplines whose common thread is object technology. From its inception, OOPSLA has served as an incubator for advanced technologies and practices. Dynamic compilation and optimization, patterns, refactoring, aspect-oriented programming, and agile methods (to name a few) all have OOPSLA roots.

OOPSLA 2004 continues that tradition. Researchers and practitioners from around the world have come to showcase their latest work. Presentations from invited speakers dovetail with technical papers, practitioner reports, expert panels, demonstrations, formal and informal educational symposia, workshops, and diverse tutorials from world-class lecturers. The popular Onward! track presents out-of-the-box thinking at the frontiers of computing. You can discuss late-breaking results with the researchers themselves at poster sessions, which culminate in the Third Annual SIGPLAN Student Research Competition. DesignFest<sup>®</sup> provides hands-on design experience in an expert-mentored environment. As always, the opportunities for mingling and professional networking are boundless.

Again this year, we're privileged to host the ACM Turing Lecture, delivered on Tuesday evening by Alan Kay, 2003 Turing Award winner. Each year, ACM bestows its most prestigious technical award on an individual who has made major and lasting contributions to computer technology. Alan's work in object-oriented programming and personal computing are legendary, making it especially fitting that he give the Turing Lecture here at OOPSLA. Don't miss this historic event!

In fact, there's so much to do and see at OOPSLA that planning a schedule can be daunting, particularly for newcomers. So whether this is your first OOPSLA or you're just feeling overwhelmed, be sure to join us Monday evening for the Newcomer Orientation right after the Welcome Reception. We'll answer your questions and give you tips on how to navigate the conference.

All these activities produce a unique environment for learning and collaboration. Take advantage. Catch up with your far-flung colleagues. Ask questions of the speakers. See the demos and interact with poster presenters. Don't be shy about investigating areas outside your expertise. OOPSLA's diversity means there are lots of opportunities for cross-pollination, and cross-pollination is a key to innovation.

Most of all, take what you learn here and apply it to your work-life. OOPSLA is all about impact, not just on the research community but on practitioners, educators, and students as well. It's our work as a community that gives rise to broad advancements in computing; no person or organization can do it alone. I believe every one of us can look forward to greater success at work, individually and collectively, through our experience at OOPSLA.

Thanks for attending, and enjoy the conference!

# Corporate Supporters

## Gold Corporate Supporters

Founded in 1983, Borland® Software Corporation is a global leader in platform independent solutions for software delivery optimization. Borland provides the software and services to optimize the planning, execution and management of software delivery, across roles, teams and technology platforms. Borland solutions fully address application lifecycle management, as well as emphasizing the alignment of business objectives with software strategies to maximize the business value of software. By leveraging Borland solutions, customers can accelerate the delivery of quality software, on time and within budget. Leveraging global partners and its category leading StarTeam®, CaliberRM™, Together®, JBuilder®, Delphi™ and VisiBroker® product lines, Borland is helping to transform software delivery within the world's best organizations. For more information, please visit <http://borland.com>.



IBM is the world's largest information technology company with 80 years of leadership in helping businesses innovate. We create tools and technologies that will enable the continued evolution of computing and computing services over the network. Our work across many disciplines is often done in concert with our colleagues in academic and government research centers, as well as "in the marketplace" with customers who provide us with challenging research problems.



IBM Software offers the widest range of applications, middleware and operating systems for all types of computing platforms, allowing customers to take full advantage of the new era of on-demand e-business. The fastest way to get more information about IBM software is through: <http://www.software.ibm.com>.

IBM Research is the world's largest information technology research organization, with more than 3,000 scientists and engineers at eight labs in six countries. IBM has produced more research breakthroughs than any other company in the IT industry. For more information on IBM Research, visit <http://www.research.ibm.com>.

Founded in 1975, Microsoft is the worldwide leader in software, services and Internet technologies for personal and business computing. The company offers a wide range of products and services designed to empower people through great software -- any time, any place and on any device.



Microsoft Visual Studio .NET is the comprehensive tool set for rapidly building and integrating Web services, Microsoft Windows®-based applications, and Web solutions. For more information, please visit <http://www.microsoft.com>.

A singular vision-The Network is The Computer-has propelled Sun Microsystems, Inc., since its inception in 1982 to its position as a leading provider of industrial-strength hardware, software, and services that make the Net work. Sun Microsystems is recognized for its development and stewardship of Java technology, the prevailing object-oriented standard for the development and deployment of the secure, portable, reliable, and scalable applications required by today's networked economy. Sun can be found in more than 100 countries and on the World Wide Web. For more information, please visit <http://www.sun.com>.



## Silver Corporate Supporter

Addison-Wesley Professional and Prentice Hall PTR are the leading publishers of high-quality and timely resources for programmers, managers, engineers, and system administrators. Our readers rely on us to provide up-to-date information on the latest technologies, as well as new approaches to current technologies. Written by the industry's most respected experts, our educational materials teach technology professionals essential new skills and timesaving techniques to help them do their jobs more effectively. Addison-Wesley Professional and Prentice Hall PTR are part of the Pearson Technology Group. As a division of Pearson Education, our readers and authors benefit from our global resources. For more information, please visit <http://www.awprofessional.com> and <http://www.phptr.com>.

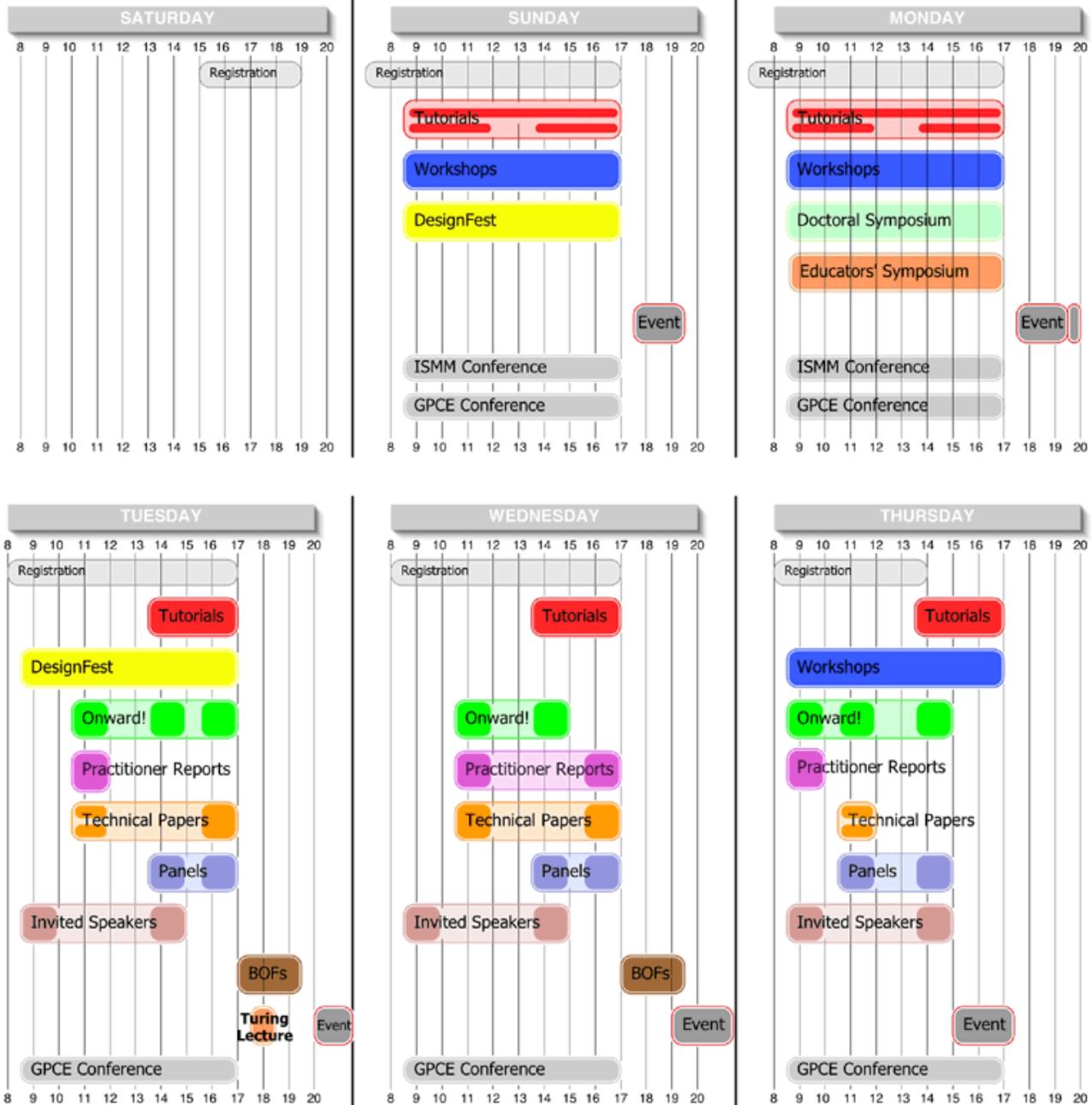


# Table of Contents



OOPSLA 2004 at a Glance .....	4
Schedule .....	5
Keynotes & Invited Speakers.....	10
Technical Papers .....	14
Panels .....	23
Practitioner Reports .....	25
Onward! .....	28
Tutorials .....	31
Workshops .....	35
DesignFest.....	38
Educators' Symposium.....	40
Doctoral Symposium .....	44
Posters .....	47
The ACM SIGPLAN Student Research Competition .....	58
Demonstrations .....	60
Special Events.....	70
FlashBoF.....	72
Maps.....	73
ACM/SIGPLAN/SIGSOFT .....	78
OOPSLA 2004 Committees.....	79
OOPSLA 2005 .....	82

# OOPSLA 2004 at a Glance



## Events

- Sunday: eclipse Technology eXchange (eTX) Reception**
- Monday: Welcome Reception, Poster Session, and Newcomer Orientation**
- Tuesday: GoF 10th Anniversary Commemorative**
- Wednesday: Special Event at the Vancouver Aquarium**
- Thursday: OOPSLA 2005 Kick-off and Ice Cream Social**

# Schedule

Registration at the Convention Level Lobby and Box Office is open at the following times:

Saturday	3:00 pm - 7:00 pm
Sunday	7:00 am - 5:00 pm
Monday	7:00 am - 5:00 pm
Tuesday	8:00 am - 5:00 pm
Wednesday	8:00 am - 5:00 pm
Thursday	8:00 am - 2:00 pm

Demonstrations occur throughout Tuesday, Wednesday and Thursday. Please see page 60 for times and topics.

Posters will be on display at the Welcome Reception on Monday. They will remain on display from 10:00-17:00 on Tuesday and Wednesday, and from 10:00-14:00 on Thursday. Please see page 47 for times and topics.

The following tables provide high-level session views of the conference. Please refer to the specific pages in the Final Program for session details. The session page numbers are noted after each session.

## Sunday, October 24

Full-Day Tutorials - p. 31 Morning Tutorials - p. 31 Full-Day Workshops - p. 35 Morning Workshops - p. 35 Full-Day DesignFest <sup>®</sup> - XPFeSt - p. 38	<b>8:30-12:00</b>
---	-------------------

Full-Day Tutorials Continued - p. 31 Afternoon Tutorials - p. 31 Full-Day Workshops Continued - p. 35 Afternoon Workshops - p. 35 Full-Day DesignFest <sup>®</sup> - XPFeSt Continued - p. 38 Half-Day DesignFest <sup>®</sup> - p. 38	<b>13:30-17:00</b>
--	--------------------

eclipse Technology eXchange (eTX) Reception - p. 70	<b>17:30-19:30</b>
--	--------------------

# Schedule

## Monday, October 25

Full-Day Tutorials - p. 31 Morning Tutorials - p. 31 Full-Day Workshops - p. 35 Full-Day Doctoral Symposium - p. 44	<b>8:30-10:00</b>	Educators' Symposium Welcoming Remarks and Keynote Address: Alan Kay - The Computer Revolution Hasn't Happened Yet! - p. 40
	<b>10:00-10:30</b>	Educators' Symposium: Posters and Demonstrations - p. 40
	<b>10:30-12:00</b>	Educators' Symposium: New Ideas for Old Problems - p. 40
	<b>12:00-13:30</b>	Educators' Symposium: Lunch
Full-Day Tutorials Continued - p. 31 Afternoon Tutorials - p. 31 Full-Day Workshops Continued - p. 35 Full-Day Doctoral Symposium Continued - p. 44	<b>13:30-15:00</b>	Educators' Symposium: Nifty Assignments - p. 40
	<b>15:00-15:30</b>	Educators' Symposium: Posters and Demonstrations - p. 40
	<b>15:30-17:00</b>	Educators' Symposium: Techniques for Teaching - p. 40
	<b>17:00-17:30</b>	Educators' Symposium: Closing - p. 40
Welcome Reception - p. 70	<b>17:30-19:30</b>	Posters at the Welcome Reception - p. 47 ACM Student Research Presentations at the Welcome Reception - p. 58
Newcomer Orientation - p. 70	<b>19:30-20:00</b>	

# Schedule

## Tuesday, October 26

<b>8:30-10:00</b>	OOPSLA Keynote: The Future of Programming, Richard F. Rashid - p. 10
-------------------	--

Full-Day DesignFest <sup>®</sup> - XPFeSt - p. 38	<b>10:30-12:00</b> Practitioner Reports: Starting on the Right Foot - p. 25 Technical Papers: Generics - p. 14 Technical Papers: Garbage Collection - p. 14 Onward!: Critiquing the Present - p. 28
---	--

Afternoon Tutorials - p. 31 Full-Day DesignFest <sup>®</sup> - XPFeSt Continued - p. 38 Half-Day DesignFest <sup>®</sup> - p. 38	<b>13:30-15:00</b> Invited Talk: Objects, Patterns, Wiki and XP: All Systems of Names, Ward Cunningham - p. 10 Panel: Model Driven Architecture: The Realities, A Year Later - p. 23 Onward!: Panel: The Last New Language Feature - p. 28
	<b>15:30-17:00</b> Technical Papers: Inheritance - p. 14 Panel: The Role of the Customer in Software Development —The XP Customer: Fad or Fashion? - p. 23 Onward!: Large Scale Reality Construction - p. 28 ACM Student Research Finalists Presentations - p. 58
FlashBoF - p. 72	<b>17:00-17:30</b>
	<b>17:30-18:30</b> ACM Turing Lecture, Alan Kay - p. 13
	<b>18:30-19:30</b>

<b>20:00-21:30</b>	GoF 10th Anniversary Commemorative - p. 70
--------------------	--

# Schedule

## Wednesday, October 27

<b>8:30-10:00</b>	Invited Talk: Code Complete 2: Realities of Modern Software Construction, Steve McConnell - p. 10
-------------------	---

<b>10:30-12:00</b>	Practitioner Reports: Staying the Course with Methodology and Process - p. 25 Technical Papers: Aspects in the Middle - p. 14 Technical Papers: Java Technologies - p. 14 Onward!: Changing the Experience of Coding - p. 28
--------------------	---

Afternoon Tutorials - p. 31	<b>13:30-15:00</b>	Invited Talk: Concrete Languages on Virtual Platforms, Herb Sutter - p. 10 Panel: The View: The Ultimate IT Chat - p. 23 Onward!: Panel: Transforming the Lifecycle - p. 28
	<b>15:30-17:00</b>	Technical Papers: Performance - p. 14 Technical Papers: Advanced Design - p. 14 Panel: Software Development: Math & Science or Arts & Crafts? - p. 23 Practitioner Reports: Working with Language Features - p. 25
FlashBoF - p. 72	<b>17:00-19:00</b>	
	<b>19:00-23:00</b>	Special Event at the Vancouver Aquarium - p. 70

# Schedule

## Thursday, October 28

Full-Day Workshops - p. 35	<b>8:30-10:00</b>	Invited Talk: Inside Amazon Web Services, Allan Vermeulen - p. 10 ACM Student Research Award Presentation - p. 58 Onward!: Rethinking Languages - p. 28 Practitioner Reports: Building Distributed Systems - p. 25
	<b>10:30-12:00</b>	Technical Papers: Languages - p. 14 Technical Papers: Verification and Validation - p. 14 Onward!: Breakthrough Ideas - p. 28 Panel: Challenges in Outsourcing and Global Development: How Will Your Job Change? - p. 23
Afternoon Tutorials - p. 31 Full-Day Workshops Continued - p. 35 Afternoon Workshops - p. 35	<b>13:30-15:00</b>	Onward! Keynote: Exocomputing in the Year 2304: A Survey of Confirmed Alien Information Technologies, Jaron Lanier - p. 10 Panel: The Great J2EE vs. Microsoft .NET Shootout - p. 23
	<b>15:00-17:00</b>	Ice Cream Social/OOPSLA 2005 Invitation - p. 70

# Keynotes & Invited Speakers

The Keynotes & Invited Speakers Track contains talks from speakers invited by the Conference Committee on new, different or otherwise interesting perspectives on topics of broad, general interest. In his opening OOPSLA Keynote talk, Richard Rashid will discuss how the development process is changing and how those changes will affect the way we think of programming over the next 10 years. Ward Cunningham will unravel his experience building tools for the not quite tangible. Steve McConnell's talk explores the software issues of yesterday and today and arrives at a few lasting truths about software development. Herb Sutter talks about the importance and viability of environments based on virtual machines and garbage collection, and demonstrates the challenges and rewards of evolving existing performance and hardware-oriented languages to operate seamlessly and with first-class status in modern virtual environments. The talk by Allan Vermeulen describes how developers are building stand-alone businesses as software vendors, web storefront aficionados, and online retail service providers. The Onward! Keynote, by Jaron Lanier, will present an overview of a variety of alien information technologies and to glean insights into how they might inform the future of human IT as well as what might be expected from future alien encounters.

## Tuesday, 8:30, Ballroom A-B The Future of Programming



*Richard F. Rashid, Microsoft Research*

The nature of the software development process is changing in significant ways. New tools for program specification, program analysis, program proof, and testing are coming out of the laboratory and into practice. Languages are being adapted to new task domains and new problem areas. There are new ways to track and manage software and hardware errors in the field and new techniques for datamining that information to reduce the time from fault to fix. At the same time, fundamental questions about the computational process are being raised by new processors. This talk will address these changes and look at how they will affect the way we think of programming over the next 10 years.

*Dr. Rashid founded and heads Microsoft Research—one of the largest industrial research organizations in the field of computer science. Before joining Microsoft in 1991, Dr. Rashid was a professor for 12 years at Carnegie-Mellon University and is best known for his work in developing the Mach operating system, which is the basic kernel underlying MacOS 10 and several commercial versions of Unix. Dr. Rashid has published papers and holds patents in many areas of computer science, including operating systems, computer vision, networking, programming languages, data compression, AI, and security. He is also the author of two computer games: AltoTrek—one of the first network computer games—and Allegiance—a massive multiplayer space strategy combat game published in 2000. He is a member of the National Academy of Engineering.*

## Tuesday, 13:30, Ballroom A-B Objects, Patterns, Wiki and XP: All Systems of Names



*Ward Cunningham, Microsoft*

What do Design Patterns in a textbook, CRC Cards on a conference room table and pages in a WikiWikiWeb site have in common? Each is an artfully incomplete specification. Each asserts some fact or decision while leaving room for complementary or evolving ideas. Unlike some communication tools, they work well at that delicious moment just before a design crystallizes. And, finally, they all share a tangled history rooted in a HyperCard stack Ward Cunningham wrote years ago. In this talk Ward unravels his experience building tools for the not quite tangible. His reflection stands as a motivation for the software practice of Refactoring which itself enables the agile discipline of Extreme Programming.

*Ward Cunningham is an Architect in the Patterns & Practices group at Microsoft Corp. He founded Cunningham & Cunningham, Inc., served as Director of R&D at Wyatt Software and as Principle Engineer in the Tektronix Computer Research Laboratory. Ward is well known for his contributions to the developing practice of object-oriented programming, the variation called Extreme Programming, and the communities hosted by his WikiWikiWeb. He is active with the Hillside Group and has served as program chair of the Pattern Languages of Programs conference which it sponsors. Ward created the CRC design method which helps teams find objects. Ward has written for PLoP, JOOP and OOPSLA on Patterns, Objects, CRC and related topics.*

# Keynotes & Invited Speakers

## Wednesday, 8:30, Ballroom A-B Code Complete 2: Realities of Modern Software Construction



*Steve McConnell, Construx*

For more than 10 years, Code Complete has been a computing-industry bestseller. The past decade has seen the emergence of object-oriented programming, the Internet, the CMM, offshore outsourcing, Java, Visual Basic, and the Agile movement as well as staggering advances in computing power and numerous other developments. How have 10 years of advances changed software construction? Has the general state of programming improved? Are any of the old techniques still relevant? Has anything moved backward? Award-winning author Steve McConnell explores the software issues of yesterday and today and arrives at a few lasting truths about software development. This talk is based on Code Complete Second Edition, available in June 2004.

*Steve McConnell is Chief Software Engineer and CEO at Construx Software where he oversees Construx's software engineering practices. Steve has worked on software projects at Microsoft, Boeing, and other Seattle-area companies. Steve is the author of Rapid Development (1996), Software Project Survival Guide (1998), and Professional Software Development (2004). His books have twice won Software Development magazine's Jolt Excellence award for outstanding software development book of the year. In 1998, readers of Software Development magazine named Steve one of the three most influential people in the software industry along with Bill Gates and Linus Torvalds. Steve is Editor in Chief Emeritus of IEEE Software magazine, serves on the Panel of Experts of the SWEBOK project, and is Chair of the IEEE Computer Society's Professional Practices Committee. Code Complete, 2d Edition was published in June 2004.*

## Wednesday, 13:30, Ballroom A-B Concrete Languages on Virtual Platforms



*Herb Sutter, Microsoft*

From the JVM to .NET, virtual machine runtime environments with garbage collection are now the mainstream of computing rather than the exception, and they are here to stay. Microsoft has bet its medium-term future on it: The successor to Win32 as the API for Windows Longhorn is WinFX, which is based on today's .NET Frameworks and puts the garbage-collected VM squarely at the heart of a major operating system. In this talk, the lead architect of C++/CLI talks about the importance and viability of environments based on virtual

machines and garbage collection, even for performance-driven applications, and demonstrates the challenges and rewards of evolving existing performance- and hardware-oriented languages to operate seamlessly and with first-class status in modern virtual environments.

*Herb Sutter is a leading authority on software development in general and C++ and .NET in particular, and the creator of dozens of innovations in the fields of distributed database systems architecture, distributed network adaptive self-organization, and cryptographic security for databases. He is currently a software architect at Microsoft, chair of the ISO C++ Standards Committee, and the author of four acclaimed books and hundreds of papers and articles on software development topics.*

## Thursday, 8:30, Ballroom A-B Inside Amazon Web Services



*Allan Vermeulen, Amazon*

After nearly a decade of building one of the most powerful e-commerce applications on the Web, Amazon.com handed developers the keys in July 2002. The results have been nothing short of astonishing as developers everywhere race to build the next big thing using Amazon.com's technology platform and product data. Al Vermeulen will show how developers are building stand-alone businesses as software vendors, web storefront aficionados, and online retail service providers.

*Allan Vermeulen is Chief Technology Officer and Vice President of Web Services Engineering for the Amazon Web Services (AWS) group of Amazon Services, Inc., a subsidiary of Amazon.com, Inc. Vermeulen oversees the building of Web services that allow third-party developers to access the atomic-level components of Amazon.com's technology platform and product data. Before joining the AWS group in 2003, Vermeulen ran the Amazon.com Platform Technologies team which is responsible for the company's technology architecture. Vermeulen was previously Chief Technology Officer and Vice President of Development at Rogue Wave Software, which builds reusable software components and services for application development. Vermeulen earned his doctorate in Systems Design Engineering from the University of Waterloo.*

# Keynotes & Invited Speakers

**Thursday, 13:30, Ballroom A-B**  
**Onward! Keynote—Exocomputing in the Year 2304: A Survey of Confirmed Alien Information Technologies**



*Jaron Lanier, Interstellar Computer Science Institute*

As more alien civilizations have been encountered in recent decades, a variety of exotic information technology strategies have come to light. It has often been difficult to analyze these technologies, as alien cognitive and social factors must be taken into account, and these are in themselves challenging to interpret. It is now becoming possible to present an overview of a variety of alien information technologies and to glean insights into how they might inform the future of human IT as well as what might be expected from future alien encounters.

*Jaron Lanier joined the Interstellar Computer Science Institute, based in Berkeley, CA, Earth, in 2003 as a Senior Research Scientist. He specializes in Virtual Reality, General Bio-information Theory, and Exocomputing.*

# ACM Turing Lecture

**Tuesday, 17:30, Ballroom A-B**  
**ACM Turing Lecture: Introductions To Computing Should Be Child's Play**



*Alan Kay, Viewpoints Research Institute*  
The 2003 Turing Award has been given to Alan Kay "For pioneering many of the ideas at the root of contemporary object-oriented programming languages, leading the team that developed Smalltalk, and for fundamental contributions to personal computing."

Perhaps the most disturbing "trend which became reality" over the last 25 years has been a recharacterization and professing of the various computing fields as though Computer Science and Software Engineering have actually been invented and can be taught in ways that parallel fields such as physics and structural engineering. This is "science & engineering envy" pure and simple!

The result is that so much of what is taught in high schools and universities looks backwards—not for historical interest, which is almost absent, or even to great ideas of the past—but (a) to emphasize what all too often have been workarounds for what we don't yet know how to do, and (b) to substitute vocational training for real knowledge and perspective.

One of the most interesting characteristics of computing in the best universities of the 1960s was that the professors told the students that nothing much of importance was known, and it was the duty of all to try to invent a real computing science and software engineering. This was a very healthy attitude and led to many good starts towards qualitatively better approaches to our exciting area of interest. Just as "civilization" is not a place or state, but a process of people who are trying to be more civilized, real computing is the process of people trying to make a better notion of computing. The most progress will be made by young people who have been encouraged to criticize old conceptions and invent new ones with an elevated notion of what constitutes a high threshold for a good idea.

It is the duty of all enfranchised computerists to help this happen. Since our paths of thinking are so conditioned by the early environments we put so much effort into learning, it is of critical importance to pay the highest attention to the introductions to our field for children, young adults and college students. This talk is about how we might introduce computing to beginners to help them see the real beauties and possibilities of our field in a way that will both get them fluent in the small amount of good stuff that is known, and most importantly to encourage them to make qualitative improvements in computing.

*Alan Kay is one of the earliest pioneers of object-oriented programming, personal computing and graphical user interfaces. His contributions been recognized with the Charles Stark Draper Prize of the National Academy of Engineering (with Robert Taylor, Butler Lampson, and Charles Thacker), the A.M. Turing Award from the Association of Computing Machinery, and the Kyoto Prize from the Inamori Foundation. This work was done in the rich context of ARPA and Xerox PARC with many talented colleagues.*

*While at the ARPA project at the University of Utah in the late 60s, he invented dynamic object-oriented programming, was part of the original team that developed continuous tone 3D graphics, was the codesigner of the FLEX Machine, an early desktop computer with graphical user interface and object-oriented operating system, conceived the Dynabook, a laptop personal computer for children of all ages, and participated in the design of the ARPAnet.*

*At Xerox PARC, inspired by children, he invented Smalltalk (with important contributions by Dan Ingalls), the first completely object-oriented authoring and operating system (which included the now ubiquitous overlapping window interface), instigated the bit-map screen, screen painting and animation, participated in desk-top publishing, other desktop media, and the development of the Alto, the first modern networked personal computer. This was part of the larger process at PARC that created an entire genre of personal computing including: the Ethernet, Laserprinters, modern word processing, client-servers and peer-peer networking.*

*He has been a Xerox Fellow, Chief Scientist of Atari, Apple Fellow, and Disney Fellow. In 2001 he founded Viewpoints Research Institute, a non-profit organization dedicated to children and learning. He is currently a Senior Fellow at HP Labs, an Adjunct Professor of Computer Science at UCLA and a Visiting Professor at Kyoto University.*

*He has been elected a Fellow of the American Academy of Arts and Sciences, the National Academy of Engineering, the Royal Society of Arts, and the Computer History Museum.*

*He has a BA in Mathematics and Biology with minor concentrations in English and Anthropology from the University of Colorado, 1966. MS and PhD in Computer Science (both with distinction) from the University of Utah, 1968 and 1969, and an Honorary Doctorate from the Kungl Tekniska Hoegskolan in Stockholm.*

*Other honors include: J-D Warnier Prix d'Informatique, ACM Systems Software Award, NEC Computers & Communication Foundation Prize, Funai Foundation Prize, Lewis Branscomb Technology Award, and the ACM SIGCSE Award for Outstanding Contributions to Computer Science Education.*

# Technical Papers

Chair: Douglas C. Schmidt, Vanderbilt University



The OOPSLA technical paper sessions showcase the premier research contributions and empirical results in object-oriented languages, systems, and applications. The 27 technical papers presented in nine sessions this year cover a range of interesting and important topics: novel software design techniques, advanced programming language features, analysis tools, run-time systems, and new approaches to validating and optimizing programs.

These papers were selected after a rigorous peer review of 173 submissions by an international program committee comprising 27 experts from all areas of object technology. Each paper was assigned at least four reviewers; papers co-authored by members of the program committee were held to even higher standards, with as many as eight reviewers assigned under strict anonymity. The program committee met

in person for two days to discuss each paper in detail, assessing their respective novelty and contribution to object technology. This peer review process has yielded a collection of papers that advance the state-of-the-art in object technology and its related fields.

## Tuesday, 10:30-12:00, Ballroom A-B Generics

Session Chair: Torsten Layda, SWX Swiss Exchange

### **Converting Java Classes to Use Generics**

Daniel von Dincklage, *University of Colorado*  
Amer Diwan, *University of Colorado*

Generics offer significant software engineering benefits since they provide code reuse without compromising type safety. Thus generics will be added to the Java language in the near future. While this extension to Java will help programmers when they are writing new code, it will not help legacy code unless it is rewritten to use generics. In our experience, manually modifying existing programs to use generics is error prone and labor intensive.

We describe a system that (i) converts non-generic classes to generic classes and (ii) rewrites the clients of the previously non-generic class to use the generic class. Our preliminary to generic classes and (ii) rewrites their clients to use the newly generified classes. Our experiments with a number of Java container classes show that our system is effective in modifying legacy code to use generics.

### **Converting Java Programs to Use Generic Libraries**

Alan Donovan, *MIT*  
Adam Kiezun, *MIT*  
Michael D. Ernst, *MIT*

Future versions of Java's type system (called JSR-14) will include support for parametric polymorphism, or generic classes. This will bring many benefits to Java programmers, not least because current Java practice makes heavy use of logically-generic classes, including container classes.

Translation of Java source code into semantically equivalent JSR-14 source code requires two steps: parameterization (adding type parameters to class definitions) and instantiation (adding the actual parameters at each use of a parameterized class). Parameterization need be done only once for a class, whereas instantiation must be performed for each client, of which there are potentially many more. Therefore, this work focuses on the instantiation problem. We present a technique to determine sound and precise JSR-14 types at each use of a class for which a generic type specification is available. Our approach is to use a precise and context-sensitive concrete class analysis to determine possible types at allocation sites, and a set-constraint-based analysis to choose consistent types for both allocation and declaration sites. We have implemented our analysis in a tool that automatically inserts type parameters into Java code, and we report its performance when applied to a number of real-world Java programs.

### **A Practical Type System and Language for Reference Immutability**

Adrian Birka, *MIT*  
Michael D. Ernst, *MIT*

This paper describes a type system that is capable of expressing and enforcing immutability constraints. The specific constraint expressed is that the abstract state of the object to which an immutable reference refers cannot be modified using that reference. The abstract state is (part of) the transitively reachable state: that is, the state of the object and all state reachable from it by following references. The type system permits explicitly excluding fields or objects from the abstract state of an object. For a statically type-safe language, the type system guarantees reference immutability. If the language is extended with downcasts, then run-time checks enforce the reference immutability constraints.

# Technical Papers

In order to better understand the usability and efficacy of the type system, we have implemented an extension to Java, called Javari, that includes all the features of our type system. Javari is interoperable with Java and existing JVMs. This paper describes the design and implementation of Javari, including the type-checking rules for the language. This paper also discusses experience with 160,000 lines of Javari code; Javari was easy to use and provided a number of benefits, including detecting errors in well-tested code.

**Tuesday, 10:30-12:00, Meeting Rooms 1-3  
Garbage Collection**

**Session Chair: Guy Steele, Sun  
Microsystems**

## ***A Unified Theory of Garbage Collection***

David F. Bacon, *IBM T.J. Watson Research Center*  
Perry Cheng, *IBM T.J. Watson Research Center*  
V.T. Rajan, *IBM T.J. Watson Research Center*

Tracing and reference counting are uniformly viewed as being fundamentally different approaches to garbage collection that poses very distinct performance properties. We have implemented high-performance collectors of both types, and in the process observed that the more we optimized them, the more similarly they behaved—that they seem to share some deep structure.

We present a formulation of the two algorithms that shows that they are in fact duals of each other. Intuitively, the difference is that tracing operates on live objects, or "matter," while reference counting operates on dead objects, or "anti-matter." For every operation performed by the tracing collector, there is a precisely corresponding anti-operation performed by the reference counting collector.

Using this framework, we show that all high-performance collectors (for example, deferred reference counting and generational collection) are in fact hybrids of tracing and reference counting. We develop a uniform cost-model for the collectors to quantify the tradeoffs that result from choosing different hybridizations of tracing and reference counting. This allows the correct scheme to be selected based on system performance requirements and the expected properties of the target application.

## ***The Garbage Collection Advantage: Improving Program Locality***

Xianglong Huang, *University of Texas at Austin*  
Stephen Blackburn, *Australian National University*  
Kathryn McKinley, *University of Texas At Austin*  
J. Eliot B. Moss, *The University of Massachusetts, Amherst*  
Zhenlin Wang, *Michigan Technological University*  
Perry Cheng, *IBM T.J. Watson Research Center*

As increases in processor speed continue to outpace increases in cache and memory speed, programs are losing more performance to poor locality. Because copying garbage collectors move objects, they have the opportunity to improve locality for languages such as Java. This paper introduces a new dynamic, online class analysis for finding and exploiting locality in a copying collector. The analysis exploits method sampling in a JIT (just-in-time) optimizing compiler. For each hot (frequently accessed) method, object reordering analysis marks the class fields that the method accesses as hot. Then at garbage collection time, the collector copies referents of hot fields together with their parent. Enhancements to this basic technique include heuristics that decay heat to respond to phase changes, group objects of hot classes together in a separate copy space, and static analysis to exclude cold basic blocks from the reordering analysis. In experiments with JikesRVM using MMTk on a range of Java programs, the overhead of dynamic class reordering is on average negligible and at most 1.9%. We compare class reordering with a number of static class oblivious orderings (e.g., breadth and depth first). The overall time variation between static orderings can be up to 25% and there is no consistent winner. In contrast, dynamic class reordering always matches or improves over the best static ordering since its history-based copying order tunes memory layout to program traversal.

## ***MC?: High-Performance Garbage Collection for Memory-Constrained Environments***

Narendran Sachindran, *University of Massachusetts, Amherst*  
Eliot Moss, *University of Massachusetts, Amherst*  
Emery Berger, *University of Massachusetts, Amherst*

Java is becoming an important platform for memory-constrained consumer devices such as PDAs and cellular phones, because it provides safety and portability. Since Java uses garbage collection, efficient garbage collectors that run in constrained memory are essential. Typical collection techniques used on these devices are mark-sweep and mark-compact. Mark-sweep collectors can provide good throughput and pause times but suffer from fragmentation. Mark-compact collectors prevent fragmentation, have low space overheads, and provide

# Technical Papers

good throughput. However, they cannot be made fully incremental and so can suffer from long pause times.

Copying collectors can provide higher throughput than either of these techniques, but because of their high space overhead, they previously were unsuitable for memory-constrained devices. This paper presents MC<sup>2</sup> (Memory-Constrained Copying), a copying, generational garbage collector that meets the demands of memory-constrained devices with soft real-time requirements. MC<sup>2</sup> has low space overhead and tight space bounds, prevents fragmentation, provides good throughput, and yields short pause times. These qualities make MC<sup>2</sup> also attractive for other environments, including desktop and server systems.

**Tuesday, 15:30-17:00, Ballroom A-B  
Inheritance**

**Session Chair: David Ungar, Sun  
Microsystems**

## ***Scalable Extensibility via Nested Inheritance***

Nathaniel Nystrom, *Cornell University*  
Stephen Chong, *Cornell University*  
Andrew Myers, *Cornell University*

Inheritance is a useful mechanism for reusing and factoring code. However, it has limitations for building extensible systems. We describe nested inheritance, a mechanism that addresses some of the limitations of ordinary inheritance and also provides much or all of the functionality of previous code reuse mechanisms. Using our experience with an extensible compiler framework, we show how nested inheritance can be used to construct this and other highly extensible software frameworks. The essential aspects of nested inheritance have been formalized in a simple object-oriented language with an operational semantics and type system; the type system of this language is sound, so no run-time type checking is required to implement it. Nested inheritance has been added to the Java language as an unobtrusive extension; we describe how our prototype implementation translates code using this new feature to ordinary Java code, without duplicating inherited code.

## ***Super and Inner—Together at Last!***

David Goldberg, *University of Utah*  
Robert Findler, *University of Chicago*  
Matthew Flatt, *University of Utah*

In an object-oriented language, a derived class may declare a method with the same signature as a method in the base class. The meaning of the re-declaration depends on the language. Most commonly, the new declaration

overrides the base declaration, perhaps completely replacing it, or perhaps using *super* to invoke the old implementation. Another possibility is that the base class always controls the method implementation, and the new declaration merely augments the method in the case that the base method calls *inner*. Each possibility has advantages and disadvantages. In this paper, we explain why programmers need both kinds of methods, and we present a language that integrates them. We also present a formal semantics for the new language, and we describe an implementation for MzScheme.

## ***Object-Oriented Encapsulation for Dynamically Typed Languages***

Nathanael Schaerli, *University of Bern*  
Andrew P. Black, *OGI School of Science & Engineering,  
Oregon Health & Science University*  
Stephane Ducasse, *University of Bern*

Encapsulation in object-oriented languages has traditionally been based on static type systems. As a consequence, dynamically-typed languages have only limited support for encapsulation. This is surprising, considering that encapsulation is one of the most fundamental and important concepts behind object-oriented programming and that it is essential for writing programs that are maintainable and reliable, and that remain robust as they evolve.

In this paper we describe the problems that are caused by insufficient encapsulation mechanisms and then present object-oriented encapsulation, a simple and uniform approach that solves these problems by bringing state of the art encapsulation features to dynamically typed languages. We provide a detailed discussion of our design rationales and compare them and their consequences to the encapsulation approaches used for statically typed languages. We also describe an implementation of object-oriented encapsulation in Smalltalk. Benchmarks of this implementation show that extensive use of object-oriented encapsulation results in a slowdown of less than 15 percent.

# Technical Papers

**Wednesday, 10:30-12:00, Ballroom A-B  
Aspects in the Middle**

**Session Chair: Elisa Baniassad, Trinity College**

## ***Measuring the Dynamic Behaviour of AspectJ Programs***

Bruno Dufour, *McGill*

Christopher Goard, *McGill*

Laurie Hendren, *McGill*

Oege de Moor, *Oxford*

Ganesh Sittampalam, *Oxford*

Clark Verbrugge, *McGill*

This paper proposes and implements a rigorous method for studying the dynamic behaviour of AspectJ programs. As part of this methodology several new metrics specific to AspectJ programs are proposed and tools for collecting the relevant metrics are presented. The major tools consist of: (1) a modified version of the AspectJ compiler that tags bytecode instructions with an indication of the cause of their generation, such as a particular feature of AspectJ; and (2) a modified version of the \*J dynamic metrics collection tool which is composed of a JVMPI-based trace generator and an analyzer which propagates tags and computes the proposed metrics. This dynamic propagation is essential, and thus this paper contributes not only new metrics, but also non-trivial ways of computing them.

We furthermore present a set of benchmarks that exercise a wide range of AspectJ's features, and the metrics that we measured on these benchmarks. The results provide guidance to AspectJ users on how to avoid efficiency pitfalls, to AspectJ implementors on promising areas for future optimization, and to tool builders on ways to understand runtime behaviour of AspectJ.

## ***Decentralizing Execution of Composite Web Services***

Mangala Gowri Nanda, *IBM India Research Laboratory*

Satish Chandra, *IBM India Research Laboratory*

Vivek Sarkar, *IBM T. J. Watson Research Center*

Distributed enterprise applications today are increasingly being built from services available over the web. A unit of functionality in this framework is a web service, a software application that exposes a set of "typed" connections that can be accessed over the web using standard protocols. These units can then be composed into a composite web service. BPEL (Business Process Execution Language) is a high-level distributed programming language for creating composite web services.

Although a BPEL program invokes services distributed over several servers, the orchestration of these services is typically under centralized control. Because performance is a major concern in enterprise applications, it is important to remove the inefficiencies introduced by the centralized control. In a distributed, or decentralized orchestration, the BPEL program is partitioned into independent sub-programs that interact with each other without any centralized control. Decentralization can increase parallelism and reduce the amount of network traffic required for an application.

This paper presents a technique to partition a composite web service written as a single BPEL program into an equivalent set of decentralized parts. It gives a new code partitioning algorithm to partition a BPEL program represented as a program dependence graph, in a way that minimizes communication costs. The paper explains why standard program partitioning approaches for multiprocessor execution do not work well in our setting. Departing from the traditional optimization target of reducing completion time, the paper uses throughput as its primary performance metric, which is more appropriate for the intended workload. It also gives a cost model to estimate throughput of a given code partition in order to select the most efficient one. Experimental results show that our algorithm can increase the throughput of example composite services substantially, easily doubling it under high system load.

## ***Resolving Feature Convolution in Middleware Systems***

Charles Zhang, *University of Toronto*

Hans-Arno Jacobsen, *University of Toronto*

Middleware provides simplicity and uniformity for the development of distributed applications. However, the modularity of the architecture of middleware is starting to disintegrate and to become complicated due to the interaction of too many orthogonal concerns imposed from a wide range of application requirements. This is not due to bad design but rather due to the limitations of the conventional architectural decomposition methodologies. We introduce the principles of horizontal decomposition which addresses this problem with a mixed-paradigm middleware architecture. This architecture uses conventional decomposition methods to implement the core functionalities of middleware and employs aspect orientation to address orthogonal properties. Our evaluation of the horizontal decomposition principles focuses on refactoring major middleware functionalities into aspects in order to modularize and isolate them from the core architecture. New versions of middleware can be created through combining the minimum core and the flexible selection of middleware aspects such as IDL data types, the asynchronous invocation, the dynamic

# Technical Papers

messaging style, and additional character encoding schemes. As a result, the primary functionality of middleware is supported with a much simpler architecture and enhanced performance. Moreover, customization and configuration of the middleware for a wide-range of requirements becomes possible.

**Wednesday, 10:30-12:00,  
Meeting Rooms 1-3  
Java Technologies**

**Session Chair: David Bacon, IBM T.J.  
Watson Research**

## ***Transparent Proxies for Java Futures***

Polyvios Pratikakis, *University of Maryland, College Park*

Jaime Spacco, *University of Maryland, College Park*

Michael Hicks, *University of Maryland, College Park*

A proxy object is a surrogate or placeholder that controls access to another target object. Proxies can be used to support distributed programming, lazy or parallel evaluation, access control, and other simple forms of behavioral reflection. However, wrapper proxies (like futures or suspensions for yet-to-be-computed results) can require significant code changes to be used in statically-typed languages, while proxies more generally can inadvertently violate assumptions of transparency, resulting in subtle bugs.

To solve these problems, we have designed and implemented a simple framework for proxy programming, which employs a static analysis based on qualifier inference, but with additional novelties. Code for using wrapper proxies is automatically introduced via a classfile-to-classfile transformation, and potential violations of transparency are signaled to the programmer. We have formalized our analysis and proven it sound. Our framework has a variety of applications, including support for asynchronous method calls returning futures. Experimental results demonstrate the benefits of our framework: programmers are relieved of managing and/or checking proxy usage, analysis times are reasonably fast, and overheads introduced by added dynamic checks are negligible, and performance improvements can be significant. For example, changing two lines in a simple RMI-based peer-to-peer application and then using our framework resulted in a large performance gain.

## ***An Efficient Parallel Heap Compaction Algorithm***

Diab Abuaiadh, *IBM Haifa Research Lab*

Yoav Ossia, *IBM Haifa Research Lab*

Erez Petrank, *Technion, Israel Institute of Technology*

Uri Silbershtein, *IBM Haifa Research Lab*

We propose a heap compaction algorithm appropriate for modern computing environments. Our algorithm is targeted at SMP platforms. It demonstrates high scalability when running in parallel but is also extremely efficient when running single-threaded on a uniprocessor. Instead of using the standard forwarding pointer mechanism for updating pointers to moved objects, the algorithm saves information for a pack of objects. It then does a small computation to process this information and determine each object's new location.

In addition, using a smart parallel moving strategy, the algorithm achieves (almost) perfect compaction in the lower addresses of the heap, whereas previous algorithms achieved parallelism by compacting within several predetermined segments.

Next, we investigate a method that trades compaction quality for a further reduction in time and space overhead.

Finally, we propose a modern version of the two-finger compaction algorithm. This algorithm fails, thus, re-validating traditional wisdom asserting that retaining the order of live objects significantly improves the quality of the compaction.

The parallel compaction algorithm was implemented on the IBM production Java Virtual Machine. We provide measurements demonstrating high efficiency and scalability. Subsequently, this algorithm has been incorporated into the IBM production JVM.

## ***Finding Your Cronies: Static Analysis for Dynamic Object Colocation***

Samuel Guyer, *The University of Texas at Austin*

Kathryn McKinley, *The University of Texas at Austin*

This paper introduces cooperative object colocation, an optimization to reduce copying costs in generational and other incremental garbage collectors by allocating connected objects together in the same space. Previous work indicates that connected objects die together, but generational copying collectors allocate new objects in a nursery regardless of the location of an older object that points to it. Cooperative colocation consists of (1) a flow insensitive static compiler analysis that finds potential colocation allocation sites; (2) a modified allocator which takes a colocation object as a parameter; and (3) a runtime test that allocates a new object together with its colocator. At runtime in a generational setting, the allocator places the object in the old space if the colocator resides in the old space, and otherwise allocates the new object in the

# Technical Papers

nursery. Unlike pretenuring, colocation makes precise per object allocation decisions, and does not require lifetime analysis or allocation site homogeneity. Experimental results for SPEC Java Benchmarks using MMTK, a Java Memory Management Toolkit in JikesRVM, show colocation can reduce nursery collection work, improving the performance of two generational collectors by up to a factor of 2, and total performance by up to 10%

## Wednesday, 15:30-17:00, Ballroom A-B Performance

### Session Chair: Jan Vitek

#### **Vertical Profiling: Understanding the Behavior of Object-Oriented Applications**

Matthias Hauswirth, *University of Colorado at Boulder*  
Peter Sweeney, *IBM Thomas J. Watson Research Center*  
Amer Diwan, *University of Colorado at Boulder*  
Michael Hind, *IBM Thomas J. Watson Research Center*

Object-oriented programming languages provide a rich set of features that provide significant software engineering benefits. The increased productivity provided by these features comes at a justifiable cost in a more sophisticated runtime system whose responsibility is to implement these features efficiently. However, the virtualization introduced by this sophistication provides a significant challenge to understanding complete system performance, not found in a traditionally compiled languages, such as C or C++. Thus, understanding system performance of such a system requires profiling that spans all levels of the execution stack, such as the hardware, operating system, virtual machine, and application.

In this work, we suggest an approach, called vertical profiling, that enables this level of understanding. We illustrate the efficacy of this approach by providing deep understandings of performance problems of Java applications run on an VM with vertical profiling support. By incorporating vertical profiling into a programming environment the programmer will be able to understand how their program interacts with the underlying abstraction levels, such as application server, VM, operating system, and hardware.

#### **Method-Level Phase Behavior in Java Workloads**

Andy Georges, *Ghent University*  
Dries Buytaert, *Ghent University*  
Lieven Eeckhout, *Ghent University*  
Koen De Bosschere, *Ghent University*

Java workloads are becoming more and more prominent on various computing devices. Understanding the

behavior of a Java workload which includes the interaction between the application and the virtual machine, is thus of primary importance during performance analysis and optimization. Moreover, as contemporary software projects are increasing in complexity, automatic performance analysis techniques are indispensable. This paper proposes an off-line method-level phase analysis approach for Java workloads that consists of three steps. In the first step, the execution time is computed for each method invocation. Using an off-line tool, we subsequently analyze the dynamic call graph (that is annotated with the method invocations' execution times) to identify method-level phases. Finally, we measure performance characteristics for each of the selected phases. This is done using hardware performance monitors. As such, our approach allows for linking microprocessor-level information at the individual methods in the Java application's source code. This is extremely interesting information during performance analysis and optimization as programmers can use this information to optimize their code. We evaluate our approach in the Jikes RVM on an IA-32 platform using the SPECjvm98 and SPECjbb2000 benchmarks. This is done according to a number of important criteria: the overhead during profiling, the variability within and between the phases and its applicability in Java workload characterization (measuring performance characteristics of the various VM components) and application bottleneck identification.

#### **Instrumentation of Standard Libraries in Object-Oriented Languages: the Twin Class Hierarchy Approach**

Konstantin Shagin, *Technion—Israel Institute of Technology*  
Assaf Schuster, *Technion—Israel Institute of Technology*  
Michael Factor, *IBM Research Lab in Haifa*

Code instrumentation is widely used for a range of purposes including profiling, debugging, visualization, logging, and distributed computing. Due to their special status within the language infrastructure, the standard class libraries, also known as system classes, provided by most contemporary object-oriented languages, are difficult and sometimes impossible to instrument. If instrumented, the use of their rewritten versions within the instrumentation code is usually unavoidable. However, since it is equivalent to "instrumenting the instrumentation," it may lead to erroneous results. Consequently, most systems avoid rewriting system classes. We present a novel instrumentation strategy that alleviates the above problems by renaming the instrumented classes. Our approach allows system classes to be instrumented, both statically and dynamically. In fact, this is the first technique that enables dynamic instrumentation of Java system classes without modifying

# Technical Papers

any runtime components. We demonstrate our approach by implementing two instrumentation-based systems: a memory profiler and a distributed runtime for Java.

**Wednesday, 15:30-17:00,  
Meeting Rooms 1-3  
Advanced Design**

**Session Chair: Gail E. Harris, Instantiated Software**

## ***Recovering Binary Class Relationships: Putting Icing on the UML Cake***

Yann-Gaël Guéhéneuc, *Université de Montréal*  
Hervé Albin-Amiot, *Ecole des Mines de Nantes*

A discontinuity exists between object-oriented modeling and programming languages. This discontinuity arises from ambiguous concepts in modeling languages and a lack of corresponding concepts in programming languages. It is particularly acute for binary class relationships—association, aggregation, and composition. It hinders the traceability between software implementation and design, thus hampering software analysis. We propose consensual definitions of the binary class relationships with four minimal properties—exclusivity, invocation site, lifetime, multiplicity. We describe algorithms to detect automatically the properties in source code and apply these on several frameworks. Thus, we bridge the gap between implementation and design for the binary class relationships, easing software analysis.

## ***Refactoring Class Hierarchies with KABA***

Mirko Streckenbach, *Universität Passau*  
Gregor Snelting, *Universität Passau*

KABA is an innovative system for refactoring Java class hierarchies. It uses the Snelting/Tip algorithm in order to determine a behavior-preserving refactoring which is optimal with respect to a given set of client programs. KABA can be based on dynamic as well as static program analysis. The static variant will preserve program behavior for all possible input values; the dynamic version guarantees preservation of behavior for all runs in a given test suite. KABA offers automatic refactoring as well as manual refactoring using a dedicated editor.

In this contribution, we recapitulate the Snelting/Tip algorithm, present the new dynamic version, and explain new extensions which allow to handle full Java. We then present five case studies which discuss the KABA refactoring proposals for programs such as javac and antlr. KABA proved that javac does not need refactoring,

but generated semantics-based refactoring proposals for antlr.

## ***Mirrors: Design Principles for Meta-level Facilities of Object-Oriented Programming Languages***

Gilad Bracha, *Sun Microsystems*  
David Ungar, *Sun Microsystems*

We identify three design principles for reflection and metaprogramming facilities in object oriented programming languages. Encapsulation: meta-level facilities must encapsulate their implementation. Stratification: meta-level facilities must be separated from base-level functionality. Ontological correspondence: the ontology of meta-level facilities should correspond to the ontology of the language they manipulate.

Traditional/mainstream reflective architectures do not follow these precepts. In contrast, reflective APIs built around the concept of mirrors are characterized by adherence to these three principles. Consequently, mirror-based architectures have significant advantages with respect to distribution, deployment and general purpose metaprogramming.

**Thursday, 10:30-12:00, Ballroom A-B  
Languages**

**Session Chair: Doug Lea, SUNY Oswego**

## ***Practical Predicate Dispatch***

Todd Millstein, *University of California, Los Angeles*

Predicate dispatch is an object-oriented (OO) language mechanism for declaratively determining the method implementation to be invoked upon a message send. With predicate dispatch, each method implementation includes a predicate guard specifying the conditions under which the method should be invoked, and logical implication of predicates determines the method overriding relation. Predicate dispatch naturally unifies and generalizes disparate forms of dynamic dispatch, including traditional OO dispatch, multimethod dispatch, and functional-style pattern matching. Despite the promise of predicate dispatch, prior languages supporting predicate dispatch have had several deficiencies that limit its utility in practice.

We introduce JPred, a practical, backward-compatible extension to Java supporting predicate dispatch. While prior languages with predicate dispatch have been extensions to toy or non-mainstream languages, we show how predicate dispatch can be naturally added to a traditional OO language. While prior languages with predicate dispatch have required the whole program to be

# Technical Papers

available for typechecking and compilation, JPred retains Java's encapsulation properties and modular typechecking and compilation strategies. While prior languages with predicate dispatch have devised special-purpose algorithms for reasoning about predicates, JPred employs general-purpose, off-the-shelf decision procedures. As a result, JPred's type system is more flexible, allowing several useful programming idioms that are spuriously rejected by those other languages. After describing the JPred language and type system, we present a case study illustrating the utility of JPred in a real-world application, including its use in the detection of several bugs.

## **Concrete Syntax for Objects: Domain-Specific Language Embedding and Assimilation without Restrictions**

Martin Bravenboer, *Universiteit Utrecht*  
Eelco Visser, *Universiteit Utrecht*

Application programmer's interfaces give access to domain knowledge encapsulated in class libraries without providing the appropriate notation for expressing domain composition. Since object-oriented languages are designed for extensibility and reuse, the language constructs are often sufficient for expressing domain abstractions at the semantic level. However, they do not provide the right abstractions at the syntactic level. In this paper we describe MetaBorg, a method for providing concrete syntax for domain abstractions to application programmers. The method consists of embedding domain-specific languages in a general purpose host language and assimilating the embedded domain code into the surrounding host code. Instead of extending the implementation of the host language, the assimilation phase implements domain abstractions in terms of existing APIs leaving the host language undisturbed.

Indeed, MetaBorg can be considered a method for promoting APIs to the language level. The method is supported by proven and available technology, i.e., the syntax definition formalism SDF and the program transformation language and toolset Stratego/XT. We illustrate the method with applications in three domains: code generation, XML generation, and user-interface construction.

## **Object-Oriented Units of Measurement**

Eric Allen, *Sun Microsystems*  
David Chase, *Sun Microsystems*  
Victor Luchangco, *Sun Microsystems*  
Jan-Willem Maessen, *Sun Microsystems*  
Guy Steele, *Sun Microsystems*

Programs that manipulate physical quantities typically represent these quantities as raw numbers corresponding

to the quantities' measurement in a particular unit (e.g., a length represented as a number of meters). This approach eliminates the possibility of catching errors resulting from adding or comparing quantities expressed in different units (as in the Mars Climate Orbiter), and does not support dimensional analysis, a powerful form of static checking on expressions of physical quantities. We show how to formulate dimensions and units as classes in a (nominally typed) object-oriented language through the use of statically typed metaclasses. Our formulation allows both parametric and inheritance polymorphism with respect to both dimension and unit types. It also allows for integration of encapsulated measurement systems, dynamic conversion factors, and declarations of scales (including nonlinear scales) with defined zeros, and nonconstant exponents on dimension types. Finally, we show how to encapsulate most of the "magic machinery" that handles the algebraic nature of dimensional units in a single metaclass that allows us to treat select static types as generators of a free abelian group.

## **Thursday, 10:30-12:00, Meeting Rooms 1-3 Verification and Validation**

**Session Chair: Mira Mezini**

## **Pluggable Verification Modules: An Extensible Protection Mechanism for the JVM**

Philip Fong, *University of Regina*

Through the design and implementation of a JVM that supports Pluggable Verification Modules (PVMs), the idea of an extensible protection mechanism is entertained. Link-time bytecode verification becomes a pluggable service that can be readily replaced, reconfigured, and augmented. Application-specific verification services can be safely introduced into the dynamic linking process of the JVM. This feature is enabled by the adoption of a previously proposed modular verification architecture, Proof Linking, which decouples bytecode verification from the dynamic linking process, rendering the verifier a replaceable module. The PVM mechanism has been implemented in an open source JVM, the Aegis VM. To evaluate the software engineering and security engineering benefits of this extensible protection mechanism, an augmented type system JAC (Java Access Control) has been successfully implemented as a PVM.

## **Finding and Preventing Run-Time Error Handling Mistakes**

Westley Weimer, *UC Berkeley*  
George Necula, *UC Berkeley*

It is difficult to write programs that behave correctly in the presence of run-time errors. Existing programming

# Technical Papers

language features often provide poor support for executing clean-up code and for restoring invariants in such exceptional situations. We present a dataflow analysis for finding a certain class of error-handling mistakes: those that arise from a failure to release resources or to clean up properly along all paths. Many real-world programs violate such resource safety policies because of incorrect error handling. Our flow-sensitive analysis keeps track of outstanding obligations along program paths and does a precise modeling of control flow in the presence of exceptions. Using it, we have found over 800 error handling mistakes almost 4 million lines of Java code. The analysis is unsound and produces false positives, but a few simple filtering rules suffice to remove them in practice. The remaining mistakes were manually verified. These mistakes cause sockets, files and database handles to be leaked along some paths. We present a characterization of the most common causes of those errors and discuss the limitations of exception handling, finalizers and destructors in addressing them.

Based on those errors, we propose a programming language feature that keeps track of obligations at run time and ensures that they are discharged. Finally, we present case studies to demonstrate that this feature is natural, efficient, and can improve reliability; for example, retrofitting a 34kLOC program with it resulted in a 0.5% code size decrease, a surprising 17% speed increase (from correctly deallocating resources in the presence of exceptions), and more consistent behavior.

## ***Chianti: A Tool for Change Impact Analysis of Java Programs***

Xiaoxia Ren, *Rutgers University*

Fenil Shah, *IBM Software Group*

Frank Tip, *IBM T.J. Watson Research Center*

Barbara Ryder, *Rutgers University*

Ophelia Chesley, *Rutgers University*

This paper reports on the design and implementation of Chianti, a change impact analysis tool for Java that is implemented in the context of the Eclipse environment. Chianti analyzes two versions of an application and decomposes their difference into a set of atomic changes. Change impact is then reported in terms of affected (regression or unit) tests whose execution behavior may have been modified by the applied changes. For each affected test, Chianti also determines a set of affecting changes that were responsible for the test's modified behavior. This latter step of isolating the changes that induce the failure of one specific test from those changes that only affect other tests can be used as a debugging technique in situations where a test fails unexpectedly after a long editing session.

We evaluated Chianti on a year (2002) of CVS data from M. Ernst's Daikon system, and found that, on average,

51% of the unit tests are affected. Furthermore, each affected unit test, on average, is affected by only 3.88% of the atomic changes. These findings suggest that our change impact analysis is a promising technique for assisting developers with program understanding and debugging.

# Panels

**Chair: Kyle Brown, IBM**



OOPSLA panels have consistently been among the best-attended and well-received attractions at the conference. They offer an engaging, entertaining, and informative examination of a timely topic from a variety of viewpoints. They also give the OOPSLA community a way to let experts tackle controversial and divisive topics head-on in a fun, interactive way that can shed welcome light on the issues we all must deal with.

## **Tuesday, 13:30-15:00, Meeting Rooms 11-12 Model Driven Architecture: The Realities, A Year Later**

**Panel Chair: Granville Miller, Microsoft**

Scott Ambler, *Amblysoft*

Steve Cook, *Microsoft*

Karl Frank, *Borland*

Sridhar Iyengar, *IBM*

Jon Kern, *Compuware*

Stephen Mellor, *Mentor Graphics Corp.*

Model Driven Architecture (MDA) is a technology that has been in the process of evolution for many years. We looked at this technology last year in a panel that raised the roof. This time, we bring in the people who are shipping the technology.

## **Tuesday, 15:30-17:00, Meeting Rooms 11-12 The Role of the Customer in Software Development—The XP Customer: Fad or Fashion?**

**Panel Chair: Steven Fraser, Independent Consultant**

**Co-convenor: Angela Martin, Victoria University of Wellington**

Robert Biddle, *Carleton University*

David Hussman, *SGF Software*

Granville Miller, *Microsoft*

Mary Poppendieck, *Cutter Consortium Consultant*

Linda Rising, *Independent Consultant*

Mark Striebeck, *VA Software*

One of the core XP (eXtreme Programming) practices is that of the "on-site customer." This panel will discuss the role, its benefits, challenges and possible extensions to main-stream software development practices.

## **Wednesday, 13:30-15:00, Meeting Rooms 11-12 The View: The Ultimate IT Chat**

**Panel Chair: Nicolai Josuttis, Solutions in Time**

Jutta Eckstein, *Objects in Action*

Lise B. Hvatum, *Schlumberger*

Mary Lynn Manns, *University of North Carolina at Asheville*

Linda Rising, *Independent Consultant*

Rebecca Wirfs-Brock, *Wirfs-Brock Associates*

The world is changing rapidly and this is especially true of the IT business. In this panel, Nicolai Josuttis chats with five internationally recognized, extraordinary female representatives of our IT business about the most important, thought-provoking, and funniest headlines and trends of the past year. Lean back and enjoy being a fly on the wall while the following panelists chat in a pub-like atmosphere.

## **Wednesday, 15:30-17:00, Ballroom C Software Development: Math & Science or Arts & Crafts?**

**Panel Chair: Jim Haungs, IBM**

Martin Fowler, *ThoughtWorks*

Richard Gabriel, *Sun Microsystems*

Ralph Johnson, *University of Illinois at Urbana Champaign*

Steve McConnell, *Construx*

We've have been proposing formal mathematical methods of software development for nearly as long as we've been developing software. Agile methods propose "emergent design," and argue that BDUF (big design up front) is old-fashioned and limiting. Both viewpoints have their advantages and disadvantages. Is software development a science or an art? Are these viewpoints reconcilable?

# Panels

**Thursday, 10:30-12:00,  
Meeting Rooms 11-12  
Challenges in Outsourcing and Global  
Development: How Will Your Job Change?**

**Panel Chair: Steven Fraser, Independent  
Consultant**

Lougie Anderson, Sabrix  
Ron Crocker, Motorola  
Martin Fowler, ThoughtWorks  
Richard Gabriel, Sun Microsystems  
Ricardo Lopez, Qualcomm  
Dave Thomas, Bedarra

If you are a software professional, your job is changing. Outsourcing and global development affect many things in our work environment: what we build, how and when we communicate, and how we prepare ourselves for the future. This panel will focus on both the social and the technical aspects of outsourcing and globalization.

**Thursday, 13:30-15:00, Ballroom C  
The Great J2EE vs. Microsoft .NET Shootout**

**Panel Chair: Martin Fowler, ThoughtWorks**

Don Box, *Microsoft*  
John Crupi, *Sun Microsystems*  
Rob High, *IBM*  
Anders Hejlsberg, *Microsoft*  
Alan Knight, *Cincom*

J2EE and Microsoft .NET have emerged as two major frameworks for software development. Are they more similar than different or are the differences significant? How do they stack up against each other? This panel will offer a lively discussion from leaders of Microsoft and J2EE technologies.

# Practitioner Reports

**Chair: Gail E. Harris, Instantiated Software**



Practitioner reports are an opportunity to hear about others' experiences applying object-oriented technologies or related software development practices. Practitioner reports explore how concepts that sound good on paper (and at conferences!) work in the real world. They are a valuable means of sharing experiences, especially at the "bleeding edge" of technology. OOPSLA attendees can find out what it is like to adopt a new language, use a new method, develop a common framework, maintain a large application, or migrate to object technology. Expectations, beliefs, and hopes can be validated, or dashed, by the experiences reported from real-world projects.

**Tuesday, 10:30-12:00, Meeting Rooms 11-12  
Starting on the Right Foot**

**Session Chair: Linda Rising, Independent Consultant**

***The Software Architect: Essence, Intuition, and Guiding Principles***

Matt McBride, *Countrywide Financial Corporation*

Software architecture is a distinct and developing discipline in the software profession. Many practitioners have apparently entered the field with little effort; adding the word "architect" to a title is easy to do. However, beneath the surface appearance, distinct approaches and toolsets are required to succeed. Reports of IT overspending and project failures emphasize the fact that these skills must be leveraged and developed. The practical application of this growing body of knowledge will continue to play an important role in the maturing of our profession, and its ability to deliver effective solutions. The software architect possesses a unique perspective and mental framework that guides the development of software systems. Additionally, strong interpersonal skills are vital to the software architect's success. In this paper, I explore the unique approaches and characteristics of the successful software architect.

***Mock Roles, not Objects***

Steve Freeman, *ThoughtWorks*

Tim Mackinnon, *ThoughtWorks*

Nat Pryce, *ThoughtWorks*

Joe Walnes, *ThoughtWorks*

Mock Objects is an extension to Test-Driven Development that supports good Object-Oriented design by guiding the discovery of a coherent system of types within a code base. It turns out to be less interesting as a technique for isolating tests from third-party libraries than is widely thought. This paper describes the process of using Mock Objects with an extended example and

reports best and worst practices gained from experience of applying the process. It also introduces jMock, a Java framework that embodies our collective experience.

***Comparison of UML and Text Based Requirements Engineering***

Brian Berenbach, *Siemens Corporate Research, Inc.*

There appears to be a real dichotomy in the use of UML vs. text based Use Case development for requirements elicitation and documentation, that is, on those projects where use cases were work products. Not only are there different processes in place for text and graphical use case modeling, but also there are a variety of approaches and philosophies in each camp. This paper will discuss the prose vs. graphical approaches to requirements elicitation, observed advantages and disadvantages of each, and suggest best practices for improving and/or creating effective processes for requirements elicitation.

**Wednesday, 10:30-12:00,  
Meeting Rooms 11-12  
Staying the Course with Methodology and Process**

**Session Chair: Steven Fraser, Independent Consultant**

***PIP: A Product Planning Strategy for the Whole Family (Or... How We Became the Brady Bunch)***

Joseph Blotner, *Sabrix, Inc.*

As a small start-up company with a heavyweight client list, Sabrix found itself at a critical crossroads in its growth in the Spring of 2003. We were consuming all product development resources satisfying commitments made to existing customers, which seriously limited our ability to react to new requirements and improve our competitive edge in the market. In the face of this resource strain, we needed to add a series of smaller plug-in type products to our existing offering. Of course, this

# Practitioner Reports

environment required 100% focus on day-to-day tasks and tactical issues, leaving no time or energy for looking ahead.

Compounding these problems were the same issues that every company faces. A lack of trust between functional groups, with Product Development on one side and Sales and Support on the other; and, only very few people in the company knowing what work is being done in Product Development. When stepping back from the situation, we realized that all of the problems could be grouped into three general categories – prioritization, collaboration and visibility.

To improve each of these areas, we developed the Product Input and Planning (PIP) process. PIP is a cross-functional, cross-product way of building a product roadmap by prioritizing work, balancing competitiveness and customer-driven requirements, and improving product-related communication in all directions within the company. The PIP Team is responsible for prioritizing and building an 18-month roadmap, focusing on the future direction of the company, while Product Management and Product Development focus on delivering the current releases.

## **eXtreme ISO?!?**

Aki Namioka, *Cisco Systems*  
Cary Bran, *Cisco Systems*

This Report comes out of our experiences with a web-based application development project that decided to adopt XP within an ISO-compliant organization. This proposal starts with some background material on our corporate ISO-compliant process, followed by a discussion of the XP-friendly practices that the development team adopted. Finally, we discuss how we integrated ISO-compliant practices into our development environment, and we conclude with a summary of the results of the ISO audit.

## **Building a Digital Archive**

Daniel Antion, *American Nuclear Insurers*

This presentation will discuss the success American Nuclear Insurers had when building a document management system. The success is largely due to the existence of a well developed object library, and the attendant ability to reuse previously defined business logic. A premise of the presentation is that we often overlook the benefit of building upon earlier work, in search of the next wave of technology. Taking advantage of reuse in this case, allowed the developers at ANI to deliver on the promise of object technology.

The resulting document management system is easier to use than commercial systems that were available, and was

developed at a significantly lower cost. The advantage of tying documents to the business objects they represent (or are derived from) enables users of the system to build document indexes automatically and to retrieve current business data along with static document data when accessing the archive.

The presentation will focus on the goals of the project, the rationale for building it in-house, a description of the features and a discussion of the class structure as it pertains to the digital archive.

**Wednesday, 15:30-17:00,  
Meeting Rooms 11-12  
Working with Language Features**

**Session Chair: Rebecca Wirfs-Brock, Wirfs-Brock Associates**

## **Synchronization-Free Concurrency: Comparing the RTSJ to C++**

Daniel Dvorak, *Jet Propulsion Laboratory, California Institute of Technology*  
William Reinholtz, *Jet Propulsion Laboratory, California Institute of Technology*

Synchronized methods in Java make it easy to produce a correct, multithread-safe design for a “single writer / multiple readers” problem. However, in the domain of real-time embedded systems, synchronized methods can introduce unacceptable time delays and even priority inversions, so designers seek non-synchronized solutions based on atomic operations. Such a solution is surprisingly complicated in the Real-Time Specification for Java (RTSJ), yet much simpler in C++. This paper examines the reasons for that difference and suggests that, in the domain of hard real-time systems, C++ holds several advantages over the RTSJ.

## **Traits: Experience with a Language Feature**

Emerson Murphy-Hill, *The Evergreen State College*  
Andrew Black, *OGI School of Science & Engineering, Oregon Health and Science University*

This paper reports our experiences using traits, collections of pure methods designed to promote reuse and understandability in object-oriented programs. Traits had previously been used to refactor the Smalltalk collection hierarchy, but only by the creators of traits themselves. This experience report represents the first independent test of these language features. Murphy-Hill implemented a substantial multi-class data structure called ropes that makes significant use of traits. We found that traits

# Practitioner Reports

improved understandability and reduced the number of methods that needed to be written by 46%.

## Thursday, 8:30-10:00, Meeting Rooms 11-12 Building Distributed Systems

**Session Chair: Lougie Anderson, Sabrix, Inc.**

### ***Web Services-Oriented Architecture in Production in the Finance Industry***

Olaf Zimmermann, *IBM Corporation*

Effective and affordable business process integration is key in the finance industry. A large German joint-use centre, supplying services to 236 individual savings banks, enhanced the integration capabilities of its core banking system (500+ complex functions) through aggressive use of Web services.

The advanced requirements, e.g., heterogeneous client environment, sub-second response times, 300% traffic growth, and interface complexity did challenge today's Web services implementations. To achieve true interoperability between MS Office XP/.NET and the Java world, and to design a SOAP envelope compression scheme were two of the most important issues that had to be solved. The current release 2 of this solution is one of the first production references for the Web Services Interoperability (WSI) Basic Profile, which sped up development and testing effort significantly.

This report discusses the rationale behind our decision for Web services, and gives an architectural overview of the integration approach. Furthermore, it features the lessons learned and best practices identified during the design, implementation and rollout of this enterprise-scale solution.

### ***Migrating to Simpler Distributed Applications***

Joachim Kainz, *Wells Fargo Bank*

In 1994 Wells Fargo Bank was the first large financial services company to invest heavily in distributed object-oriented applications for high-volume, mission-critical applications using version 1 of OMG's Common Object Request Broker Architecture. Wells Fargo continued to improve upon its distributed applications technology leadership by launching a Model Driven Architecture[MDA] initiative in 1999.

Between 2001 and 2003 the original CORBA-based solution was completely replaced with new Web-Services-based distributed applications. This report contrasts the past and the present approach to such applications.

The final section of this report utilizes lessons learned from using CORBA & Web-Services for Intranet applications and proposes the "application transparent distribution" as a potential paradigm shift which may lead to radically faster and more efficient distributed application development.

### ***Validating Structural Properties of Nested Objects***

Darrell Reimer, *IBM*

Edith Schonberg, *IBM*

Kavitha Srinivas, *IBM*

Harini Srinivasan, *IBM*

Julian Dolby, *IBM*

Aaron Kershenbaum, *IBM*

Frameworks are widely used to facilitate software reuse and accelerate development time. However, there are currently no systematic mechanisms to enforce the explicit and implicit rules of these frameworks. This paper focuses on a class of framework rules that place restrictions on the properties of data structures in framework applications. We present a mechanism to enforce these rules by the use of a generic "bad store template" which can be customized for different rule instances. We demonstrate the use of this template to validate specific bad store rules within J2EE framework applications. Violations of these rules cause subtle defects which manifest themselves at runtime as data loss, data corruption, or race conditions. Our algorithm to detect "bad stores" is implemented in the Smart Analysis-Based Error Reduction (SABER) validation tool, where we pay special attention to facilitating problem understanding and remediation, by providing detailed problem explanations. We present experimental results on four commercially deployed e-commerce applications that show over 200 "bad stores".

# Onward!

**Chair: Geoff A. Cohen, Coherence Engine**



Onward! contains technical and philosophical papers describing new paradigms or metaphors in computing, new thinking about objects, new framings of computational problems or systems, and new technologies. Papers in Onward! aren't aimed at advancing the state of the art—they're aimed, instead, at altering or redefining the art by proposing a leap forward—or sideways—for computing.

**Tuesday, 10:30-12:00, Ballroom C**  
**Critiquing the Present**

**Session Chair: Geoff Cohen, Coherence Engine**

**Notes on Notes on Postmodern Programming**  
James Noble, *Victoria University of Wellington*  
Robert Biddle, *Carleton University*

These notes have the status of letters written to ourselves: we wrote them down because, without doing so, we found ourselves making up new arguments over and over again. So began the abstract of our earlier paper "Notes on Postmodern Programming". We now revisit the issue of postmodern programming, and attempt to address some of the questions raised by our exposition. To illustrate the nature of postmodernism we do not do this directly, but instead present a series of snapshots, parodies, and imagined conversations that we hope will help. What do you think of the abstract so far? Self-reference and a irreverent approach are part of this topic, so it's important to chill out and let things flow. We claim that computer science and software design grew up amid the unquestioned landscape of modernism, and that too often we cling to the otherwise ungrounded values, even as modernism itself is ever more compromised.

**Looking For Love (in All the Wrong Places)**  
David West, *New Mexico Highlands University and University of New Mexico*

Onward! seeks future directions. The Feyerabend Project seeks to redefine computing. Thoroughly disillusioned with traditional and contemporary practices, a significant number of us are looking for a software development profession that we can love. Humans with a history of repeating unsuccessful relationships are advised to look inward in order to discover the roots of the problem that leads them to making the same mistake over and over again. The same advice applies to the profession of software development—contemplative self examination

will help us discover the true sources of our problems. This paper introduces five factors that arise from reflective examination of computing and software development.

**Tuesday, 13:30-15:00, Ballroom C**  
**Onward! Panel: The Last New Language Feature**

Chair: William Cook, *University of Texas, Austin*  
Martin Rinard, *MIT*  
Tim Sheard, *Oregon Graduate Institute*  
Axel Schmolitsky, *Universität Hamburg*  
Cristina Videira Lopes, *University of California, Irvine*

Every new language and every new version of an old language offers new features. Some are syntactic sugar for doing something old, some provide genuinely new functionality. But almost none change the basic way we program. What new language features--as some argue objects did more than two decades ago--could make programming different?

**Tuesday, 15:30-17:00, Ballroom C**  
**Large Scale Reality Construction**

**Session Chair: Richard P. Gabriel, Sun Microsystems**

**Methodology Work is Ontology Work**  
Brian Marick, *Testing Foundations*

I argue that a successful switch from one methodology to another requires a switch from one ontology to another. Large-scale adoption of a new methodology means "infecting" people with new ideas about what sorts of things there are in the (software development) world and how those things hang together. The paper ends with some suggestions to methodology creators about how to design methodologies that encourage the needed "gestalt switch".

## **Protocols for Processes: Programming in the Large for Open Systems**

Munindar Singh, *North Carolina State University*  
Amit Chopra, *North Carolina State University*  
Nirmit Desai, *North Carolina State University*  
Ashok Mallya, *North Carolina State University*

The modeling and enactment of business processes is being recognized as key to modern information management. The expansion of Web services has increased the attention given to processes, because processes are how services are composed and put to good use. However, current approaches are inadequate for flexibly modeling and enacting processes. These approaches take a logically centralized view of processes, treating a process as an implementation of a composed service. They provide low-level scripting languages to specify how a service may be implemented, rather than what interactions are expected from it. Consequently, existing approaches fail to adequately accommodate the essential properties of the business partners of a process (the partners would be realized via services)—their autonomy (freedom of action), heterogeneity (freedom of design), and dynamism (freedom of configuration)

Flexibly represented protocols can provide a more natural basis for specifying processes. Protocols specify what rather than how; thus they naturally maximize the autonomy, heterogeneity, and dynamism of the interacting parties. We are developing an approach for modeling and enacting business processes based on protocols. This paper describes some elements of (1) a conceptual model of processes that will incorporate abstractions based on protocols and roles; (2) the semantics or mathematical foundations underlying the conceptual model and mapping global views of processes to the local actions of the parties involved; (3) methodologies involving rule-based reasoning to specify processes in terms of compositions of protocols

**Wednesday, 10:30-12:00, Ballroom C**  
**Changing the Experience of Coding**

**Session Chair: Cristina Videira Lopes,**  
**University of California, Irvine**

### **Example Centric Programming**

Jonathan Edwards, *MIT*

Programmers tend to understand programs by thinking of concrete examples. Example Centric Programming seeks to add IDE support for examples throughout the process of programming. Instead of programmers interpreting examples in their head, the examples are written down and the IDE interprets them automatically. Advanced UI

techniques are used to present the results closely integrated with the code. Traditionally distinct programming tools (the editor, Read-Eval-Print-Loop, debugger, and test runner) are unified into a single tool that might be called an example-enlightened editor. This is expected to benefit a wide spectrum of programming activities, for both novice and experienced programmers. Some novel methods for testing and development are suggested. A prototype has been implemented for Java in Eclipse.

### **Finding Bugs is Easy**

David Hovemeyer, *University of Maryland*  
William Pugh, *University of Maryland*

Many techniques have been developed over the years to automatically find bugs in software. Often, these techniques rely on formal methods and sophisticated program analysis. While these techniques are valuable, they can be difficult to apply, and they aren't always effective in finding real bugs.

Bug patterns are code idioms that are often errors. We have implemented automatic detectors for a variety of bug patterns found in Java programs. In this paper, we describe how we have used bug pattern detectors to find real bugs in several real-world Java applications and libraries. We have found that the effort required to implement a bug pattern detector tends to be low, and that even extremely simple detectors find bugs in real applications.

From our experience applying bug pattern detectors to real programs, we have drawn several interesting conclusions. First, we have found that even well tested code written by experts contains a surprising number of obvious bugs. Second, Java (and similar languages) have many language features and APIs which are prone to misuse. Finally, that simple automatic techniques can be effective at countering the impact of both ordinary mistakes and misunderstood language features.

**Wednesday, 13:30-15:00, Ballroom C**  
**Onward! Panel: Transforming the Lifecycle**

Jonathan Edwards, *MIT*

Richard P. Gabriel, *Sun Microsystems*

David West, *New Mexico Highlands University and New Mexico University*

Alistair Cockburn, *Humans and Technology*

Since the invention of the compiler, the software life cycle has generally been edit, compile, debug, ship. That's a recipe for creating software that is static, buggy, and unresponsive to changing circumstances and user

# Onward!

needs. Can we move beyond this model? Can we move testing earlier and later? Can software evolve?

## Thursday, 8:30-10:00, Ballroom C Rethinking Languages

**Session Chair: William Cook, University of Texas, Austin**

### **Granule-Oriented Programming**

Yinliang Zhao, *Xi'an Jiaotong University*

A program will become obsolete or lower effectiveness in solving domain problems due to many reasons. One main reason is because the program becomes unfitting to its context, which is defined as a sum of functionalities of all what support the program solving the domain problems, for instance, a runtime environment, meta-strategies in the domain, etc. This unfitting phenomenon exists in many complex systems, causing them premature end of their life cycle or a decrease in performance or accuracy in solving problems. In existing programming systems we pay little attention to unfitness of a program to its context, in language expressivity. Granule-oriented programming is an evolvment metaphor in which programs are ground into code ingredients in order to localize unfitting parts of a program as explicitly as possible, and then compound them into granular output code in which the code granulation space is formed. Code granulation space is an expression of a program in multiple-abstraction framework. The goal of building code granulation space for a program is to localize unfitness in a well-formed and multi-layered framework. In this paper, we propose and briefly describe the notion of granule-oriented programming.

### **Languages of the Future**

Tim Sheard, *OGI School of Science & Engineering, Oregon Health & Science University*

This paper explores a new point in the design space of formal reasoning systems—part programming language, part logical framework. The system is built on a programming language where the user expresses equality constraints between types and the type checker then enforces these constraints. This simple extension to the type system allows the programmer to describe properties of his program in the types of witness objects which can be thought of as concrete evidence that the program has the property desired. These techniques and two other rich typing mechanisms, rank-N polymorphism and extensible kinds, create a powerful new programming idiom for writing programs whose types enforce semantic properties.

A language with these features is both a practical programming language and a logic. This marriage between two previously separate entities increases the probability that users will apply formal methods to their programming designs. This kind of synthesis creates the foundations for the languages of the future.

## Thursday, 10:30-12:00, Ballroom C Onward! Breakthrough Ideas

**Session Chair: Geoff Cohen, Coherence Engine**

We present twenty short ideas for changing the way we program or use computers, how we view computation, and how we balance order and chaos.

## Thursday, 13:30-15:00, Ballroom A-B Onward! Keynote—Exocomputing in the Year 2304: A Survey of Confirmed Alien Information Technologies

As more alien civilizations have been encountered in recent decades, a variety of exotic information technology strategies have come to light. It has often been difficult to analyze these technologies, as alien cognitive and social factors must be taken into account, and these are in themselves challenging to interpret. It is now becoming possible to present an overview of a variety of alien information technologies and to glean insights into how they might inform the future of human IT as well as what might be expected from future alien encounters.

*Jaron Lanier, Interstellar Computer Science Institute: Jaron Lanier joined the Interstellar Computer Science Institute, based in Berkeley, CA, Earth, in 2003 as a Senior Research Scientist. He specializes in Virtual Reality, General Bio-information Theory, and Exocomputing.*

# Tutorials

Chair: Steve Metsker, CapTech Ventures



Take advantage of OOPSLA tutorials!

If you think you're too inexperienced—or too experienced!—to benefit from OOPSLA tutorials, please think again! OOPSLA gathers the world's finest educators on OO topics, covering the full breadth of classic and cutting-edge topics. Our presenters not only have world-class expertise, they're also successful presenters who know how to share their hard-won knowledge with you.

OOPSLA will provide 56 tutorials to choose from. It's a pity that the maximum that you can take is seven! Please take some time and carefully peruse the opportunities. The time you spend selecting your tutorials will certainly be repaid.

## Sunday Morning Tutorials

**Sunday, 8:30-12:00, Meeting Room 2**

**1 Model-Driven Software Development: Introduction & Best Practices**

Markus Voelter, *Independent Consultant*  
Jorn Bettin, *SoftMetaWare*

**Sunday, 8:30-12:00, Meeting Room 3**

**2 Scrum and Agile Project Management**

Dr. Dan Rawsthorne, *Net Objectives*

**Sunday, 8:30-12:00, Meeting Room 14**

**3 Java Generics**

Angelika Langer, *Independent Trainer / Mentor / Consultant*

**Sunday, 8:30-12:00, Meeting Room 10**

**4 Distributed .NET**

Michael Stal, *Siemens AG*

**Sunday, 8:30-12:00, Meeting Room 9**

**5 Usage-Centered Design in Agile Development**

Jeff Patton, *Abstractics LLC*

**Sunday, 8:30-12:00, Meeting Room 16**

**6 Guided Inspection of UML Models**

John McGregor, *Clemson University*

**Sunday, 8:30-12:00, Meeting Room 8**

**7 Large-Scale Software Architecture: A Practical Guide Using UML**

Jeff Garland, *CrystalClear Software, Inc*

**Sunday, 8:30-12:00, Meeting Room 13**

**8 C++ Templates—From Containers to Expression Templates**

Nicolai Josuttis, *Solutions in Time*

**Sunday, 8:30-12:00, Meeting Room 17**

**9 Essential Object-Oriented Analysis and Design**

Jill Aden, *EDS*  
Joseph Brennan, *EDS*

**Sunday, 8:30-12:00,**

**Governor General Suite D**

**GP1 Adaptive Object-Model Architecture: Dynamically Adapting to Changing Requirements**

Joseph Yoder, *The Refactory*

## Sunday Afternoon Tutorials

**Sunday, 13:30-17:00, Meeting Room 2**

**10 Skills for the Agile Designer**

Rebecca Wirfs-Brock, *Wirfs-Brock Associates*

**Sunday, 13:30-17:00, Meeting Room 3**

**11 Dungeons and Patterns!**

Steve Metsker, *CapTech Ventures*  
William Wake, *Independent Consultant*

**Sunday, 13:30-17:00, Meeting Room 9**

**12 Java Reflection**

Ira Forman, *IBM*  
Nate Forman, *Ticom Geomatics*

# Tutorials

**Sunday, 13:30-17:00, Meeting Room 8**

**13 Designing with Patterns**

John Vlissides, *IBM T.J. Watson Research*

**Sunday, 13:30-17:00,**

**Governor General Suite D**

**14 Enterprise Aspect-Oriented Programming with AspectJ**

Ron Bodkin, *New Aspects of Software*  
Nicholas Lesiecki, *VMS*

**Sunday, 13:30-17:00, Meeting Room 16**

**16 Patterns for Effective Management of Resources in Software Systems**

Prashant Jain, *IBM India Research Laboratory*  
Michael Kircher, *Siemens AG*

**Sunday, 13:30-17:00, Meeting Room 13**

**17 Patterns for Writing Effective Use Cases**

Steve Adolph, *WSA Consulting Inc*  
Paul Bramble, *AT&T Labs*

**Sunday, 13:30-17:00, Meeting Room 17**

**GP2 Multi-Stage Programming in MetaOCaml**

Walid Taha, *Rice University*  
Walid Taha, *Rice University*

**Sunday, 13:30-17:00, Meeting Room 10**

**GP3 Generative Software Development**

Krzysztof Czarnecki, *University of Waterloo*  
Jack Greenfield, *Microsoft Corporation*

## Sunday Full-Day Tutorials

**Sunday, 8:30-17:00, Meeting Room 12**

**18 Beyond the Gang of Four**

Kevlin Henney, *Curbralan Limited*

**Sunday, 8:30-17:00, Metropolis**

**19 Bots & Pieces: A Hands-On Investigation of Generative Pattern Languages using Mindstorm Robots**

Alan O'Callaghan, *De Montfort University*  
Maria Kavanagh, *De Montfort University*

**Sunday, 8:30-17:00, Meeting Room 7**

**20 Extreme Programming for the Enterprise**

Joshua Kerievsky, *Industrial Logic, Inc.*  
Ken Scott-Hlebek, *Industrial Logic, Inc.*

## Monday Morning Tutorials

**Monday, 8:30-12:00, Boardroom**

**21 Feature Oriented Programming and Product-Lines**

Don Batory, *University of Texas at Austin*

**Monday, 8:30-12:00, Meeting Room 12**

**22 Enterprise Integration Patterns: Introduction to Messaging**

Bobby Woolf, *IBM Corp*

**Monday, 8:30-12:00, Meeting Room 10**

**23 MDA and Other Model-Oriented Technologies**

Krzysztof Czarnecki, *University of Waterloo*  
David S. Frankel, *David Frankel Consulting*  
Petter Graff, *Inferdata Corporation*

**Monday, 8:30-12:00, Meeting Room 9**

**24 Aspect-Oriented Programming with AspectJ**

Erik Hilsdale, *Palo Alto Research Center*  
Mik Kersten, *University of British Columbia*

**Monday, 8:30-12:00, Meeting Room 1**

**25 Notes on the Forgotten Craft of Software Architecture**

Douglas Schmidt, *Vanderbilt University*

**Monday, 8:30-12:00, Meeting Room 7**

**26 Extend, Publish, Enable: Understanding and Writing Eclipse Plug-ins**

Erich Gamma, *IBM*  
Kai-Uwe Maetzel, *IBM*

**Monday, 8:30-12:00, Meeting Room 13**

**27 Implementing Lean Software Development: Practical Approaches for Applying Lean Principles to Software Development**

Mary Poppendieck, *Poppendieck.LLC*  
Tom Poppendieck, *Poppendieck.LLC*

# Tutorials

Monday, 8:30-12:00, Meeting Room 15

## **28** *Crash Ten Projects a Day—A Software Project Simulation*

Jens Coldewey, *Coldewey Consulting*

Monday, 8:30-12:00, Meeting Room 16

## **29** *The C# Programming Language*

Eric Meier, *McIntire School of Commerce at UVA*

Monday, 8:30-12:00, Meeting Room 17

## **30** *Test-Driven Development Workout!*

William Wake, *Independent Consultant*  
Steve Metsker, *CapTech Ventures*

## Monday Afternoon Tutorials

Monday, 13:30-17:00, Meeting Room 13

## **31** *Enterprise Integration Patterns—Designing Successful Integration Solutions*

Gregor Hohpe, *ThoughtWorks, Inc.*

Monday, 13:30-17:00, Meeting Room 9

## **32** *Hands-on Programming with AspectJ*

Erik Hilsdale, *Palo Alto Research Center*

Monday, 13:30-17:00, Meeting Room 17

## **33** *SWT: The Eclipse Standard Widget Toolkit*

Carolyn MacLeod, *IBM OTI Lab, Ottawa*  
Grant Gayed, *IBM OTI Lab, Ottawa*

Monday, 13:30-17:00, Meeting Room 16

## **34** *Agile Requirements: Tailoring the Functional Requirements Specification Process to Improve Agility*

Jennitta Andrea, *ClearStream Consulting*  
Gerard Meszaros, *ClearStream Consulting*

Monday, 13:30-17:00, Meeting Room 12

## **35** *Pattern-Oriented Software Architecture: Patterns for Concurrent and Distributed Objects*

Douglas Schmidt, *Vanderbilt University*

Monday, 13:30-17:00, Meeting Room 15

## **37** *Principles of Object-Oriented Modeling and Simulation with Modelica*

Peter Fritzson, *Linköping University, Sweden*  
Peter Bunus, *Linköping University, Sweden*

Monday, 13:30-17:00, Meeting Room 7

## **38** *Class Loading in Java: Building Dynamic Systems Without Pain*

Justin Gehrtland, *Relevance*

Monday, 13:30-17:00, Meeting Room 10

## **39** *From Models to Code with the Eclipse Modeling Framework*

Ed Merks, *IBM*  
Frank Budinsky, *IBM*

## Monday Full-Day Tutorials

Monday, 8:30-17:00, Meeting Room 2

## **40** *Using Domain Specific Languages, Patterns, Frameworks and Tools to Assemble Applications*

Jack Greenfield, *Microsoft Corporation*  
Keith Short, *Microsoft Corporation*  
Steve Cook, *Microsoft Corporation*  
Stuart Kent, *Microsoft Corporation*

Monday, 8:30-17:00, Meeting Room 8

## **41** *Better Software—No Matter What*

Scott Meyers, *Independent Consultant*

Monday, 8:30-17:00, Meeting Room 14

## **GP4** *Program Transformation Systems: Theory and Practice for Software Generation, Maintenance and Reengineering*

Ira Baxter, *Semantic Designs, Inc.*  
Hongjun Zheng, *Semantic Designs, Inc.*

## Tuesday Afternoon Tutorials

Tuesday, 13:30-17:00, Meeting Room 8

## **42** *Domain-Driven Design*

Eric Evans, *Domain Language, Inc.*  
Ralph Johnson, *University of Illinois*

Tuesday, 13:30-17:00, Meeting Room 7

## **43** *Architectural Patterns for Enabling Application Security*

Joseph Yoder, *The Refactory, Inc.*

# Tutorials

**Tuesday, 13:30-17:00, Meeting Room 9**

## **44 Using Java to Create Wireless Applications**

David Hemphill, *Gearworks, Inc.*

**Tuesday, 13:30-17:00, Meeting Room 10**

## **45 Organizational Patterns: Beyond Agility to Effectiveness**

Neil Harrison, *Avaya Labs*

James Coplien, *North Central College, Illinois*

**Tuesday, 13:30-17:00, Meeting Room 13**

## **46 Effective Interface Design: Seven Recommendations for Improving the Design of Interfaces in Code**

Kevlin Henney, *Curbralan Limited*

## Wednesday Afternoon Tutorials

**Wednesday, 13:30-17:00, Meeting Room 13**

## **47 Healing the Architecture**

Klaus Marquardt, *Draeger Medical AG*

**Wednesday, 13:30-17:00, Meeting Room 9**

## **48 Teaching Java: An Eventful Approach**

Kim Bruce, *Williams College*

**Wednesday, 13:30-17:00, Meeting Room 10**

## **49 Testing Component-Based Software**

John McGregor, *Clemson University*

**Wednesday, 13:30-17:00, Meeting Room 8**

## **50 The Elements of Software Design**

Bjorn Freeman-Benson, Ph.D., *Center for Urban Simulation and Policy, University of Washington*

David Socha, Ph.D., *Center for Urban Simulation and Policy, University of Washington*

## Thursday Afternoon Tutorials

**Thursday, 13:30-17:00, Meeting Room 7**

## **51 User-Centered Design with Use Case and Agile Methods**

William Hudson, *Syntagm Ltd.*

**Thursday, 13:30-17:00, Meeting Room 9**

## **52 Efficient Implementation of Object-Oriented Programming Languages**

Craig Chambers, *University of Washington*

**Thursday, 13:30-17:00, Meeting Room 10**

## **53 Service-Oriented Computing**

Munindar Singh, *North Carolina State University*

Michael Huhns, *University of South Carolina*

**Thursday, 13:30-17:00, Meeting Room 8**

## **54 Object-Oriented Reengineering: Patterns & Techniques**

Oscar Nierstrasz, *University of Bern*

Stéphane Ducasse, *University of Bern*

# Workshops

**Chair: Helen Sharp, The Open University, UK**



OOPSLA workshops are intensive collaborative sessions where groups of object technologists meet to surface, discuss, and attempt to solve challenging problems facing their field. The topics covered by this year's workshops are diverse, and we also have some joint workshops with our co-located conference GPCE, which OOPSLA attendees are welcome to attend.

Workshops are either full or half-day events, and most of them take place on Sunday (24th) or Monday (25th). However we also have one full day and one half day workshop on Thursday (28th). If you'd like to attend one of the workshops, and haven't already done so, then contact the relevant organizers as they are responsible for managing attendance.

Above all, workshops are lively interactive events which rely on the active participation of attendees. Please help to make them a success.

**Sunday 8:30-12:00,  
Governor General Suite A**  
***Third "Killer Examples" for Design Patterns and Objects First Workshop***  
Carl Alphonse, *University at Buffalo, State University of New York*  
Stephen Wong, *Rice University*  
Dung "Zung" Nguyen, *Rice University*  
Phil Ventura, *State University of West Georgia*

**Sunday 8:30-17:00, Meeting Room 5**  
***Patterns for Retrospectives***  
Linda Rising, *Independent Consultant*  
Mary Lynn Manns, *University of North Carolina at Asheville*

**Sunday 8:30-17:00,  
Governor General Suite B**  
***The Customer Role in Agile Projects***  
Jens Coldewey, *Coldewey Consulting*  
Mary Poppendieck, *Poppendieck LLC*  
Klaus Marquardt, *Draeger Medical AG*

**Sunday 8:30-17:00,  
Governor General Suite C**  
***The 4th OOPSLA Workshop on Domain-Specific Modeling***  
Juha-Pekka Tolvanen, *MetaCase*  
Jonathan Sprinkle, *University of California, Berkeley*  
Matti Rossi, *Helsinki School of Economics*

**Sunday 8:30-17:00, Pavilion C**  
***Ontologies as Software Engineering Artifacts***  
Joerg Pechau, *CoreMedia AG*  
Petra Becker-Pechau, *University of Hamburg*  
Ralf Klischewski, *University of Hamburg*  
Martin Lippert, *it-wps Workplace Solutions Ltd.*

**Sunday 8:30-17:00, Gazebo I**  
***Third International Workshop on Agent-Oriented Methodologies***  
Paolo Bresciani, *ITC-irst*  
John Debenham, *University of Technology, Sydney*  
Paolo Giorgini, *University of Trento*  
Cesar Gonzalez-Perez, *University of Technology, Sydney*  
Brian Henderson-Sellers, *University of Technology, Sydney*  
Graham Low, *University of New South Wales*

**Sunday 8:30-17:00, Pavilion A**  
***Eclipse Technology eXchange (ETX)***  
Michael Burke, *IBM T.J. Watson Research Center*  
Bjorn Freeman-Benson, *University of Washington*  
Margaret Storey, *University of Victoria*

**Sunday 8:30-17:00, Gazebo II**  
***Legacy Transformation: Capturing Business Knowledge from Legacy Systems***  
Jean Bézivin, *ATLAS Group (INRIA & LINA), University of Nantes, France*  
Sergio de Cesare, *Brunel University, UK*  
Grant Holland, *Sun Microsystems, Inc.*  
Mark Lycett, *Brunel University, UK*  
Chris Partridge, *Brunel University, UK*

**Sunday 8:30-17:00, Pavilion B**  
***Extreme Construction***  
Joseph Bergin, *Pace University*  
Fred Grossman, *Pace University*

# Workshops

**Sunday 8:30-17:00, Boardroom**

**OOPSLA/GPCE: Managing Variabilities Consistently in Design and Code**

Christa Schwanninger, *Siemens AG*  
Danilo Beuche, *pure-systems GmbH*  
Krzysztof Czarnecki, *University of Waterloo*  
Mira Mezini, *Darmstadt University of Technology*  
Markus Voelter, *ingenieurbüro für softwaretechnologie*

**Sunday 8:30-17:00, Meeting Room 4**

**Best Practices and Methodologies in Service-Oriented Architectures: Paving the Way to Web-services Success**

Amir Zeid, *The American University in Cairo*  
Ali Arsanjani, *IBM Corporation*  
Brian Henderson-Sellers, *University of Technology, Sydney*  
Kerrie Holley, *IBM Corporation*

**Sunday 8:30-17:00, Meeting Room 6**

**GPCE: 6th GPCE Young Researchers Workshop 2004**

Karine Arnout, *ETH Zurich- Douglas*  
R. Dechow, *Oregon State University*  
Andreas Pleuss, *University of Munich*  
Matthew J. Rutherford, *University of Colorado at Boulder*  
Denis Shestakov, *Turku Center for Computer Science*  
Daniel von Dincklage, *University of Colorado at Boulder*

**Sunday 8:30-17:00, Meeting Room 15**

**GPCE: Software Transformation Systems Workshop**

Magne Haveraaen, *University of Bergen*  
Jim Cordy, *Queen's University*  
Jan Heering, *CWI*  
Ganesh Sittampalam, *Oxford University*

**Sunday 13:30-17:00,**

**Governor General Suite A**

**Gumption Traps: Debugging Team Motivation**

Ivan Moore, *ThoughtWorks*  
Rachel Davies, *Agile Experience*

**Monday 8:30-17:00, Pavilion B**

**OOPSLA/GPCE: Best Practices for Model-Driven Software Development**

Jorn Bettin, *SoftMetaWare, New Zealand*  
Ghica van Emde Boas, *Independent Consultant, The Netherlands*  
Aditya Agrawal, *Vanderbilt University, USA*  
Markus Völter, *Independent Consultant, Germany*  
Jean Bézivin, *University of Nantes, France*

**Monday 8:30-17:00,**

**Governor General Suite C**

**Second Workshop on Method Engineering for Object-Oriented and Component-Based Development**

Magdy Serour, *University of Technology, Sydney, Australia*  
Brian Henderson-Sellers, *University of Technology, Sydney, Australia*  
Cesar González-Pérez, *University of Technology, Sydney, Australia*  
Pavel Hruby, *Microsoft Business Solutions, Copenhagen, Denmark*  
Don Firesmith, *The Software Engineering Institute (SEI), USA*  
Dilip Patel, *School of Computing, Information Systems and Mathematics, South Bank University, UK*  
Dan Rawsthorne, *Net Objectives, USA*  
Bernhard Rumpe, *Munich University of Technology, Germany*  
Hadar Ziv, *eBuilt Inc., USA*

**Monday 8:30-17:00, Gazebo II**

**Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design**

Ana Moreira, *Universidade Nova de Lisboa*  
Awais Rashid, *University of Lancaster*  
Elisa Baniassad, *Trinity College*  
Bedir Tekinerdogan, *University of Twente*  
Paul Clements, *Carnegie Mellon University*  
Joao Araujo, *Universidade Nova de Lisboa*

**Monday 8:30-17:00, Meeting Room 5**

**Component and Middleware Performance**

Petr Tuma, *Charles University, Czech Republic*  
Paul Brebner, *University College London, United Kingdom*  
Emmanuel Cecchet, *INRIA Rhone-Alpes, France*

**Monday 8:30-17:00,**

**Governor General Suite B**

**Extravagaria II: Art Assisting Science**

Richard Gabriel, *Sun Microsystems, Inc.*  
Janet Holmes, *Boise State University*  
Kirstie Bellman, *The Aerospace Corporation*

**Monday 8:30-17:00,**

**Governor General Suite A**

**Challenges in Outsourcing and Global Development: How will your job change?**

Dennis Mancl, *Lucent Technologies*  
William Opdyke, *North Central College*  
Steven Fraser, *Independent Consultant*

# Workshops

## Monday 8:30-17:00, Meeting Room 3

### **Design Patterns Revisited**

Brian Foote, *University of Illinois at Urbana-Champaign*  
James Noble, *Victoria University of Wellington, NZ*  
Kyle Brown, *IBM*  
Dirk Riehle, *Stanford University*

## Monday 8:30-17:00, Meeting Room 4

### **C++, Boost, and the Future of C++ Libraries**

Jeff Garland, *CrystalClear Software, Inc*  
Nicolai Josuttis, *Solutions In Time*  
Kevlin Henney, *Curbralan Limited*  
Jeremy Siek, *Indiana University*

## Monday 8:30-17:00, Pavilion C

### **Objects in Large Distributed Applications III (OLDA-III)**

Peter Dickman, *University of Glasgow*  
Karen Renaud, *University of Glasgow*  
Huw Evans, *Kelvin Institute*

## Monday 8:30-17:00, Revival of Dynamic Languages, Gazebo I

Gilad Bracha, *Sun Microsystems*  
Wolfgang De Meuter, *Vrije Universiteit Brussel*  
Stéphane Ducasse, *University of Bern*  
Oscar Nierstrasz, *University of Bern*  
Roel Wuyts, *Université Libre de Bruxelles*

## Monday 8:30-17:00, Pavilion A

### **Building Software for Pervasive Computing**

Cristina Lopes, *University of California, Irvine*  
Steffen Schaefer, *IBM*  
Siobhán Clarke, *Trinity College Dublin*  
Anind Dey, *University of California, Berkeley*  
Tzilla Elrad, *Illinois Institute of Technology*  
Robert Filman, *NASA Ames Research Center*  
Jens Jahnke, *University of Victoria*  
Tony Willis, *T-systems International*

## Monday 8:30-17:00, Governor General Suite D

### **GPCE: First MetaOCaml Workshop**

Walid Taha, *Rice University*

## Thursday 8:30-17:00, Meeting Room 14

### **3rd International Workshop on Language Runtimes: "Impact of Next Generation Processor Architectures On Virtual Machine Technologies"**

Yahya Mirza, *Aurora Borealis Software LLC*  
Aart Bik, *Intel Corporation*  
George Bosworth, *Microsoft Corporation*  
Tarek El-Ghazawi, *George Washington University*  
Mootaz Elnozahy, *IBM Austin Research Labs*  
Alexander Garthwaite, *Sun Microsystems Laboratories*  
Vinod Grover, *Microsoft Corporation*  
Richard Lethin, *Reservoir Labs, Inc*  
Timothy Mattson, *Intel Corporation*  
Mark Mendell, *IBM Canada*

## Thursday 13:30-17:00, Meeting Room 13

### **The Java Platform: Tiger and Beyond**

Josh Bloch, *Sun Microsystems*  
Gilad Bracha, *Sun Microsystems*  
Doug Lea, *SUNY Oswego*  
William Pugh, *University of Maryland*  
David Bacon, *IBM Research*  
Onno Kluyt, *Java Community Process*

**Chair: Rob van den Berg, Siemens VDO Automotive**



What would OOPSLA be without DesignFest! A long time OOPSLA favorite, it attracts a large number of OOPSLA's participants and provides them with an opportunity to learn more about object oriented design by doing design. DesignFest is about sharpening your skills by rolling up your sleeves and working on a real problem with others in the field. You can expect to interact with novices to experts. This year's DesignFest extends its reach with XPfest. At XPfest, participants will also learn more about Extreme Programming.

DesignFest is a free event (for conference registrants) created to give OOPSLA attendees the opportunity to learn more about design by doing it. Participants are organized into small groups to solve a particular problem.

Each team will have a moderator and a recorder. The moderator is responsible for making sure the team makes progress, without leading them in any particular direction. The recorder is responsible for recording the design the group produces and what they have learned.

The problem descriptions will be distributed at the start of each DesignFest session, along with a timetable, process hints, and a list of deliverables. Each team should produce a poster that summarizes their work. The posters will be presented at the end of the conference for all conference attendees to view.

Even if you haven't already registered for DesignFest, we are able to accommodate new participants at the last minute. If you are interested in participating, speak to one of the DesignFest organizers at the beginning of a DesignFest session. Note that all DesignFest sessions are separate sessions, last half a day or a full day. You are invited to take a tour of the previous DesignFests at <http://designfest.acm.org>.

**Sunday, 8:30-17:00, Meeting Room 1  
DesignFest® - XPfest**

**Sunday, 13:30-17:00, Meeting Room 1  
DesignFest®**

**Tuesday, 10:30-17:00,  
Governor General Suite A, B, C, D  
DesignFest® - XPfest**

**Tuesday, 13:30-17:00,  
Governor General Suites A, B, C, D  
DesignFest®**

The following problems have been selected for this year:

### ***The Registration and Assignment of Optional Sessions Preferences***

How to ease registration of DesignFest participants. This problem has been defined about every year a DesignFest was held at OOPSLA. The real origin of the problem definition is subject to mythology. People that have the chance to participate in a symposium can have some optional course assigned. To get these assignments rounded up, a registrar needs to know the preferences of

the candidates and the available time slots they have. This is basically the DesignFest registration problem. It has a somewhat wider application however.

### ***Interactive Television Applications***

Digital television allows interactive content to accompany standard broadcasts. The development of bespoke interactive content is expensive. You are to design a system that will allow the non-technical producers of television programs to build interactive content from a set of high-level building blocks.

### ***Extreme Programming Project***

This project involves using the Extreme Programming (XP) methodology to design and build a small working presentation tool in one day. The organizers will provide ongoing instruction in the XP practices as appropriate, and will also act as the customer for the software project. The day will proceed as a series of iterations. The project will involve building a small presentation tool in Java, similar to Microsoft PowerPoint.

### ***Logistic System for Storage and Distribution***

In a milk factory there are basically four parts: intake of fresh milk, production of various milk products, cooled storage and distribution. This problem deals with the

design of a system for the storage and distribution facilities in a milk factory.

## ***Case Management***

Case Management is a business function common to government health benefit programs, and insurance and financial organizations. This problem involves developing an object oriented framework for use in developing specific case management systems to support case workers in these industries. These systems provide access to all necessary information for case workers to process work or respond to customer service needs. They may also provide workflow and business rules processing to support case workers.

# Educators' Symposium

**Chair: Eugene Wallingford, University of Northern Iowa**



Join us at OOPSLA's 13th forum for object-oriented educators and trainers! This one-day symposium offers a venue for educators and trainers to share their experiences teaching object-oriented technology and to explore ideas that can help us understand and teach OO technology better.

This year's symposium offers new ideas for addressing the problems faced by OO educators. We will have presentations of teaching ideas, presentations of nifty assignments for students, demonstrations of novel OO teaching tools, a poster session—and plenty of active sessions where attendees work together to put these ideas into practice.

This year, we are honored and privileged to have as our keynote speaker the 2003 Turing Award winner, Dr. Alan Kay. In his talk, "The Computer Revolution Hasn't Happened Yet!," Dr. Kay will suggest rather different approaches to teaching computer science.

**Monday, 8:30-10:00, Ballroom C**

## Welcome and Introductions – Educators' Symposium Chair

### Keynote Address: The Computer Revolution Hasn't Happened Yet!

Alan Kay, *Viewpoints Research Institute*

There are many compelling reasons to believe that the real computer revolution—the one that mirrors the vast changes in thinking that came more than a century after the technology of the printing press—hasn't happened yet. If so, then we should be teaching and training young computer scientists very differently than is done now. Formal education tends to focus on the "last great thing, several generations removed" and to teach it as received truth. In a field that has the majority of its journey in its future, there is almost no real received truth, and the students need to be taught in ways that will allow them to be major participants in the very different future destiny of our field. We need them to move the field, not to occupy it. This talk will try to suggest rather different approaches to teaching the computer and systems sciences.

**Monday, 10:00-10:30 (Repeated at 15:00), Ballroom C**

## Posters

### ***Systems Design Thinking Using Patterns: A Report from rOOts 2004***

Maria Kavanagh and Alan O'Callaghan, *De Montfort University*, [kavanagh@dmu.ac.uk](mailto:kavanagh@dmu.ac.uk)

### ***On the Relative Advantages of Teaching Web Services in J2EE vs. .NET***

Sandeep Kachru and Edward Gehringer  
*Blackbaud, Inc., and North Carolina State University*  
[sandeep.kachru@blackbaud.com](mailto:sandeep.kachru@blackbaud.com)

## Software Demonstrations

### ***Programming with Alice***

Richard Zaccane and Stephen Cooper, *Bucknell University and Saint Joseph's University*,  
[zaccane@bucknell.edu](mailto:zaccane@bucknell.edu)

### ***A Paradigm for Teaching Modeling Environment Design***

Jonathan Sprinkle and James Davis, *University of California, Berkeley, and Vanderbilt University*  
[sprinkle@eecs.berkeley.edu](mailto:sprinkle@eecs.berkeley.edu)

### ***greenfoot: Combining Object Visualisation with Interaction***

Michael Kölling and Poul Henriksen, *University of Southern Denmark*, [mik@mip.sdu.dk](mailto:mik@mip.sdu.dk)

**Monday, 10:30-12:00, Ballroom C**

## New Ideas for Old Problems

### ***"Objects First, Interfaces Next" or Interfaces Before Inheritance***

Axel Schmolitzky, *University of Hamburg*,  
[schmolitzky@acm.org](mailto:schmolitzky@acm.org)

Objects first is a pedagogy that tries to introduce the core concepts of object-oriented programming—classes, objects, and methods—as early as possible in a programming course, even before variables, types,

# Educators' Symposium

assignments and control structures are explicitly introduced. The concept of a named interface is typically introduced at a much later stage, usually in connection with inheritance, polymorphism, and abstract classes. In this paper we point out that interfaces as a language mechanism can be introduced much earlier, even before inheritance. This way the concept of an explicit class interface can be decoupled from the more complicated issues of inheritance and subtype-polymorphism.

## **Ancestor Worship in CS1: On the Primacy of Arrays**

*Phil Ventura*, State University of West Georgia  
[pventura@westga.edu](mailto:pventura@westga.edu)

*Christopher Egert*, University at Buffalo, SUNY,  
[egert@cse.buffalo.edu](mailto:egert@cse.buffalo.edu)

*Adrienne Decker*, University at Buffalo, SUNY,  
[adrienne@cse.buffalo.edu](mailto:adrienne@cse.buffalo.edu)

History has given us the array as the fundamental data structure to present to students within the CS1 curriculum. However, with the recent growth in popularity of object-oriented languages for CS1 (C++, Java, C#), and with that, the acceptance of the Objects-First or Objects-Early approach to teaching CS1, it becomes imperative that we re-evaluate our long-held beliefs about what is appropriate to teach. It is our position that the first data structure that students are exposed to should not be arrays, but rather some other form of collection. We will give some examples of how to use `java.util.HashMap` and some of the other Java Collections classes in substitution of arrays. We also present data concerning the academic performance of students using arrays versus those using Java Collections for CS1 lab exercises.

## **greenfoot: Combining Object Visualisation with Interaction**

*Michael Kölling*, University of Southern Denmark,  
[mik@mip.sdu.dk](mailto:mik@mip.sdu.dk)

*Poul Henriksen*, University of Southern Denmark,  
[polle@mip.sdu.dk](mailto:polle@mip.sdu.dk)

The introduction of programming education with object-oriented languages slowly migrates down the curriculum and is now often introduced at the high school level. This migration requires teaching tools that are adequate for the intended target audience. In this paper, we present a new tool, named greenfoot, for teaching object-oriented programming aimed at students at or below college level, with special emphasis of supporting school age learners. Greenfoot was designed by analysing and combining the most beneficial aspects of several existing tools. It aims at combining the simplicity and visual appeal of microworlds with much of the flexibility and interaction of BlueJ. To achieve its goals of providing a suitable

learners' environment, greenfoot provides a meta-framework that allows easy creation of different, significantly varied microworlds.

**Monday, 13:30-15:00, Ballroom C**

## **Nifty Assignments**

### **Opening Paper: Examples that Can Do Harm in Learning Programming**

*Katherine Malan*, University of South Africa  
[malankm@unisa.ac.za](mailto:malankm@unisa.ac.za)

*Ken Halland*, University of South Africa,  
[hallakj@unisa.ac.za](mailto:hallakj@unisa.ac.za)

Examples form an integral part of learning to program. In this paper we argue that the role of examples should go beyond merely illustrating concepts or principles and should be programmers. We identify four common pitfalls to avoid when designing examples for teaching programming. We show how examples that are too abstract or too complex can be harmful in explaining new concepts to students. We also show how some examples used to illustrate new concepts can undermine previously taught concepts by not applying these concepts consistently. Finally, we show how some examples can do harm by undermining the very concept they are introducing. The aim of this paper is to encourage educators to think critically about examples before using them.

### **Abstract Factories and the Shape Calculator**

*Eric Cheng, Dung Nguyen, Mathias Ricken, and Stephen Wong*, Rice University, [dxnguyen@cs.rice.edu](mailto:dxnguyen@cs.rice.edu)

The assignment presents an orchestrated step-by-step process to create a GUI application that dynamically loads an arbitrary shape, displays it on the screen and computes its area. The shape can have an arbitrary set of configuration properties (e.g., width, height, color, etc.). This is very difficult to achieve using conditional statements but is straightforward using the abstract factory design pattern. This assignment serves as an introduction to GUI programming in Java, where the GUI programming, which besides being a valuable skill unto itself, is used to emphasize and illustrate compelling usage of encapsulation, inheritance and polymorphism. It is targeted at CS1 students in an objects-first curriculum.

# Educators' Symposium

## **Using the Game of Life to Introduce Design Patterns to Freshmen**

Michael Wick, *University of Wisconsin, Eau Claire*,  
wickmr@uwec.edu

Understanding design patterns is an important aspect of modern software development. Frequently, design patterns are introduced late into an undergraduate curriculum, typically in a junior or senior design course. We believe that design patterns deserve a more ubiquitous role in the undergraduate curriculum. In particular, we have found that with slight modifications and simplifications we are able to effectively introduce specific design patterns in our first programming course. Further, the necessary modifications and/or simplifications do not interfere with the primary message of each design pattern but rather highlight it. This presentation demonstrates a nifty assignment in which students refactor a traditional implementation of the classic Game of Life using the Command and Visitor design patterns. We have found this assignment to be particularly enjoyable for the students and quite effective in demonstrating the separation of concerns that are central to each of these two design patterns.

## **From Concrete to Abstract, the Power of Generalization**

Chris Nevison, *Colgate University*, chris@cs.colgate.edu

In this assignment students are given a program that solves a maze, with a display that shows the steps toward the solution. The given program has three variations of an iterative search (implemented using the strategy pattern), depth-first search, breadth-first search, best-first search (the latter using a priority queue). The students are asked to disentangle the problem-specific aspects of this code from the search strategy so as to define an abstract problem solver that can be applied to any problem fitting the model of step-by-step searching of a solution space. This requires three steps: defining appropriate interfaces that specify the information about the problem needed by the abstract solver, defining the abstract solver to find a solution using these interfaces, and defining the maze problem so as to implement these interfaces. With the abstract solver created, students should also be able to implement other problems fitting this model, such as the word-ladder game or a search in a graph for a Hamiltonian circuit, so that the abstract solver can be applied to them. The assignment demonstrates the power of generalization using abstract classes and interfaces as a bridge between concrete problems and the solution algorithm. This assignment is intended for a course in software engineering or an advanced programming course with emphasis on design. Students should already be familiar with object-oriented programming, inheritance, abstract classes and interfaces.

## **Marine Biology Simulation**

Eric Cheng, Dung Nguyen, Mathias Ricken, and Stephen Wong, *Rice University*, dxnguyen@cs.rice.edu

We redesigned the AP Marine Biology Simulation to enhance its robustness, security, flexibility and extensibility. This assignment uses it as a framework for a case study in the second semester of an objects-first curriculum. The students extend the system by adding new components such as new fish and new environments. The students learn the inner workings of the framework by rewriting critical portions of it. JUnit tests are used to specify and enforce behavioral contracts of the various components in the system. The assignment serves as a compelling example of how to use design patterns to capture appropriate abstractions and demonstrates the power derived from them. Test-driven development via unit testing provides immediate feedback regarding the correctness of their work while maintaining the freedom to implement code in different ways.

**Monday, 15:00-15:30 (Repeat of 10:00),  
Ballroom C**

## **Posters**

### **Systems Design Thinking Using Patterns: A Report from rOOts 2004**

Maria Kavanagh and Alan O'Callaghan, *De Montfort University*, kavanagh@dmu.ac.uk

### **On the Relative Advantages of Teaching Web Services in J2EE vs. .NET**

Sandeep Kachru and Edward Gehringer, *Blackbaud, Inc., and North Carolina State University*  
sandeep.kachru@blackbaud.com

## **Software Demonstrations**

### **Programming with Alice**

Richard Zaccone and Stephen Cooper, *Bucknell University and Saint Joseph's University*,  
zaccone@bucknell.edu

### **A Paradigm for Teaching Modeling Environment Design**

Jonathan Sprinkle and James Davis, *University of California, Berkeley, and Vanderbilt University*  
sprinkle@eecs.berkeley.edu

# Educators' Symposium

## **greenfoot: Combining Object Visualisation with Interaction**

Michael Kölling and Poul Henriksen, *University of Southern Denmark*, [mik@mip.sdu.dk](mailto:mik@mip.sdu.dk)

**Monday, 15:30-17:00, Ballroom C**

### **Techniques for Teaching**

#### ***If I had a model, I'd model in the mornin'***

Kurt Fenstermacher, *University of Arizona*, [kurtf@eller.arizona.edu](mailto:kurtf@eller.arizona.edu)

Despite the importance of modeling in producing high-quality software, modeling often receives scant attention in academic curricula. The recent (sometimes heated) discussion of the Object Management Group's Model-Driven Architecture (MDA) has created an opportunity and offered the motivation for making modeling a more central part of the study of software designs. This paper discusses the rationale for modeling, how modeling is currently taught in one graduate program in Management Information Systems and some experiences in the teaching of modeling to masters' level graduate students.

#### ***Event-Driven Programming Facilitates Learning Standard Programming Concepts***

Kim B. Bruce, *Williams College*, [kim@cs.williams.edu](mailto:kim@cs.williams.edu)  
Andrea Danyluk, *Williams College*, [andrea@cs.williams.edu](mailto:andrea@cs.williams.edu)  
Thomas Murtagh, *Williams College*, [tom@cs.williams.edu](mailto:tom@cs.williams.edu)

We have designed a CS1 course that integrates event-driven programming from the very start. We argue that event-driven programming is simple enough for CS1 when introduced with the aid of a library that we have developed. In this paper we argue that early use of event-driven programming makes many of the standard topics of CS 1 much easier for students to learn by breaking them into smaller, more understandable concepts.

#### ***Activity Session: Framegames!***

Steve Metsker, *CapTech Ventures*, [steve.metsker@acm.org](mailto:steve.metsker@acm.org)  
William Wake, *Independent Consultant*, [william.wake@acm.org](mailto:william.wake@acm.org)

To help your students be motivated, active, and alert in class, it helps to include interactive simulations and games in your teaching. It takes a lot of preparation, though, to create new games for your classes.

Framegames offer an approach to including interactive content without wearing yourself out. A framegame is a

shell into which you can load a lesson-of-the-day to quickly create an enjoyable and instructive game. In this hands-on session, we'll introduce two frames and four games, and give you the chance to test out the games. You'll walk away with a pair of frames that you can apply immediately to spice up your curriculum and energize your students without exhausting yourself!

**Monday, 17:00-17:30, Ballroom C**

### **Closing**

#### ***Quick Activity: Take-Home Quiz***

Symposium attendees take to the open mike to share what they have learned on the day—and what they intend to do with what they've learned when they return home.

#### ***Closing Remarks***

Eugene Wallingford, *The University of Northern Iowa*, [wallingf@cs.uni.edu](mailto:wallingf@cs.uni.edu)

# Doctoral Symposium

**Chair: Laurie Williams, North Carolina State University**



The goal of the OOPSLA 2004 Doctoral Symposium is to provide useful guidance for the completion of the dissertation research and the initiation of a research career. The Symposium will provide an interactive forum for doctoral students in one of two phases in their doctoral progress.

**Apprentices:** Students who are just beginning their research, are not ready to actually make a research proposal, but are interested in learning about structuring research and getting some research ideas.

**Proposers:** Students who have progressed far enough in their research to have a structured proposal, but will not be defending their dissertation in the next 12 months. This 12-month stipulation is set in place because we would like for the students to have sufficient time to incorporate the advice and suggestions discussed in the symposium.

This year, the mentors are Laurie Williams (North Carolina State University), Bjorn Freeman-Benson (University of Washington), Jim Coplien (Vrije Universiteit Brussel, University of Manchester), Philippe Kruchten (University of British Columbia), James Noble (Victoria University of Wellington), and Mario Wolczko (Sun Microsystems, Inc.). To provide further opportunity for discussion and feedback, doctoral symposium presenters have a poster on display during the conference and a two-page short paper published in the Conference Companion..

Due to the mentoring nature of the event, it is only open to those selected for participation.

## Monday, 8:30-17:00, Metropolis

### ***A Framework for Removing Redundant Context Management Services in Enterprise JavaBeans Application Servers***

Mircea Trofin, *Dublin City University*

We propose a framework for removing redundant context management services in contextual composition frameworks, with focus on Enterprise JavaBeans. It is expected that by applying our framework, performance can be improved without the loss of modularity.

### ***A Language-Independent Approach to Software Maintenance Using Grammar Adapters***

Suman Roychoudhury, *University of Alabama at Birmingham*

A long-standing goal of software engineering is to construct software that is easily modified and extended. Recent advances in software design techniques, such as aspect-oriented software development and refactoring, have offered new approaches to address challenges of software evolution. Several tools and language extensions have been developed by others to enable these techniques in a few popular programming languages. However, software exists in a variety of languages. An unfortunate consequence of legacy system adaptation is that new software engineering tools are often developed from scratch without preserving and reusing the knowledge

gained from the previous construction of the tool in a different language and platform context. To address this problem, this paper summarizes two core research ideas. First, the concept of extending several software reengineering techniques in disparate programming languages is explored. A core focus of this objective is the abstraction of transformation functions to enable design maintenance in legacy based systems. The second research objective extends the first goal to understand the fundamental science for constructing a generic platform using grammar adapters to enable language-independent software maintenance.

### ***Efficient Data Race and Deadlock Prevention in Concurrent Object-Oriented Programs***

Piotr Nienaltowski, *ETH Zurich*

The main goal of this PhD thesis is to propose and implement a methodology for the construction of programs based on the SCOOP model and reasoning about their correctness and liveness properties. In particular, the set of correctness rules that guarantee the absence of data races will be refined and formalized; an augmented type system will be proposed to enforce these rules at compile time. Furthermore, an efficient methodology for deadlock prevention, avoidance, detection, and resolution will be developed. A working implementation of SCOOP will be provided. It will take into consideration the proposed mechanisms and serve as a basis for further refinements of the model.

# Doctoral Symposium

## **Generic Ownership: Practical Ownership Control in Programming Languages**

Alex Potanin, *Victoria University of Wellington*

This research abstract outlines the work we plan to do as part of my PhD. In particular, we propose to devise a practical way of integrating ownership control into existing programming languages in a way that will help with adoption of ownership in the general programming community.

Modern programming languages provide little support for object encapsulation and ownership. Escaped aliases to private objects can compromise both security and reliability of code in reference-abundant languages such as Java. Object ownership is a widely accepted approach to controlling aliasing in programming languages. Proposals for adding ownership to programming languages do not directly support type genericity. We propose Generic Ownership - a unified approach to providing generics and ownership. By including support for default ownership, Generic Ownership imposes no more syntactic or runtime overheads than traditional generic types. We have implemented Generic Ownership in the context of the Ownership Generic Java (OGJ) programming language, an extension to Generic Java, and we ground the formal side of this work within the Featherweight Generic Java framework. We hope that our work will help bring full support for object encapsulation to the mainstream programming world.

## **Interactive Visualization of Object-Oriented Programs**

Paul Gestwicki, *University at Buffalo*

This research is motivated by the need to obtain better comprehension of the execution of object-oriented programs. We describe a novel approach to runtime visualization of object-oriented programs. We focus specifically on Java since it has been widely adopted by industry and academia. Our visualization methodology features:

- visualizations of execution state using a custom notation that clarifies the important fact that objects are environments of execution;
- visualizations of program history using dynamically-generated, interactive sequence diagrams;
- forward and reverse execution;
- support for visual queries regarding object, variable, and method behavior during program execution, the results of which are integrated into the object and sequence diagrams;

- advanced drawing capabilities driven by a combination of source code and runtime analysis; and
- the use of standard Java compilers and virtual machines.

Our approach is realized in a prototypical software tool, JIVE: Java Interactive Visualization Environment. JIVE is useful as a pedagogic tool and as a visual debugger, and its notations provide a visual operational semantics for Java.

## **Modeling Dynamics of Agile Software Development**

Lan Cao, *Georgia State University*

Agile software development challenges the traditional way of software development and project management. Strong interest among researchers and practitioners has generated substantial literature and debate over the agile approach and has even prompted traditional, plan based approaches to consider ways to make them agile. However, current research on the applicability and effectiveness of agile methods is fragmented and limited to selected aspects such as pair programming. The research seeks to address this void in current literature by developing an integrative simulation model using system dynamics based on both quantitative and qualitative data collected from real projects.

The primary objective of my dissertation is to develop an integrative view of agile software development to enhance our understanding and make predictions about the agile process. By modeling the dynamics of agile software development process, the applicability and effectiveness of agile methods will be investigated, and the impact of agile practices on project performance in terms of quality, schedule, cost, customer satisfaction will be examined.

My system dynamics simulation model of agile software development integrates the essential practices in agile development such as agile planning, short iterations, customer involvement, refactoring, and pair programming. These practices are modeled in four sub-models: Agile Planning and Control, Refactoring and Quality of Design, Change Management, and Customer Behavior. The model describes the behavior generated by the interaction of agile practices and their impact on critical project variables including project scope, schedule, and cost.

# Doctoral Symposium

## **Modular Generics**

Jeremy Siek, *Open Systems Laboratory, Indiana University*

The purpose of this research is to design a programming language, named G, for generic programming that improves on C++ and Haskell with regards to modularity. In previous work we compared several languages with regards to their support for generic programming, implementing a generic graph library in each of the languages. We then identified language features and properties that were important for implementing the library. Haskell and C++ stood out as providing the best support for generic programming. However, even these two languages have flaws. C++ provides neither separate type checking nor separate compilation for function templates. Even worse, there are loop-holes in the name lookup rules that defeat the modularity provided by namespaces. In Haskell, there are also modularity problems: situations can arise where a user may not be able to use two independently developed libraries due to clashing instance declarations. Thus, with the design of G we aim to build on the strengths of C++ and Haskell and improve the modularity of generic libraries.

## **Refining Designs along Middleware-Specific Concern-Dimensions at Different MDA-Levels of Abstraction**

Raul Silaghi, *Swiss Federal Institute of Technology in Lausanne, Switzerland*

For the MDA approach to software development to become a reality for distributed enterprise systems, MDA needs to provide both modeling support for middleware-specific concerns and tool support for helping developers refine their designs along such concern-dimensions at different MDA-levels of abstraction. In order to address these issues, the MDA-compliant Enterprise Fondue method proposes a hierarchy of UML profiles as a means for addressing middleware-specific concerns at different MDA-levels of abstraction, along with model transformations to incrementally refine existing design models according to the proposed profiles. Tool support is provided through the Parallax framework, which enables developers to modularize crosscutting concerns into aspect-promoting Eclipse plug-ins.

# Posters

Chair: Torsten Layda, SWX Swiss Exchange



OOPSLA 2004 Posters cover the same interest areas as the Technical Papers, Practitioner Reports, and the Onward! program. The poster session is an informal and highly interactive environment that gives OOPSLA attendees the opportunity to engage with one another in discussions about relevant, ongoing work and critical issues in key areas. The session also gives conference attendees the chance to learn about work in areas with which they want to become familiar and about preliminary research results. Researchers and practitioners can describe their work-in-progress and elaborate on work presented in other conference forums. All OOPSLA participants can obtain rapid, low-cost introductions to interesting work and technologies in object-oriented software engineering. They can also provide input and feedback directly to the authors.

The Posters program begins with a special session at the Welcome Reception on Monday evening. All posters will be on display and the authors will be present to meet with attendees and discuss their work. Posters also enable interactions to occur throughout the conference. After the Welcome Reception, the posters will be displayed in the OOPSLA Courtyard, where OOPSLA participants can view them at their convenience. Poster authors will be available as time permits. The goal is to encourage small groups of individuals interested in a technical area to gather and interact.

## Poster Sessions:

**Monday, 17:30-19:30, Welcome Reception in the Vancouver Convention Centre**

**Tuesday, 10:00-17:00, Courtyard**

**Wednesday 10:00-17:00, Courtyard**

**Thursday 10:00-14:00, Courtyard**

## Posters:

### **<CTRL> + <ALT> + <TOOL PARADIGM SHIFT>**

Russ Freeman, *Ergnosis Ltd.*  
Roly Perera, *Ergnosis Ltd.*  
Phil Webb, *Ergnosis Ltd.*

Explore a fresh tool platform which defines a revolutionary new environment for software development. Program editing is considered first and foremost as a direct manipulation of a rich semantic model whose persistent representation as plaintext source code is somewhere between conveniently incidental and almost entirely irrelevant.

### **A Core Calculus of Mixins and Incomplete Objects**

Lorenzo Bettini, *Dipartimento di Sistemi e Informatica, Università di Firenze, Italy*  
Viviana Bono, *Dipartimento di Informatica, Università di Torino, Italy*  
Silvia Likavec, *Dipartimento di Informatica, Università di Torino, Italy*

We design a calculus that combines class-based features with object-based ones, aiming at unifying the "best of both worlds." In a mixin-based approach, mixins are seen as incomplete classes from which incomplete objects can be instantiated. Incomplete objects can be completed in an object-based fashion.

### **A Creative Approach for Teaching Framework Development**

Amir Zeid, *The American University in Cairo*

Framework development is one of the most challenging software development tasks. Teaching framework development is even more challenging. In this paper, we propose a creative process to teach framework development. We propose using peer review in the process. We present the process, examples and findings out of our experience.

### **A Search System for Java Programs by using Extracted JavaBeans Components**

Hironori Washizaki, *National Institute of Informatics*  
Yoshiaki Fukazawa, *Waseda University*

We propose a new component-extraction-based program search system. Our system analyses existing Java programs, acquires relationships among classes, and extracts JavaBeans components composed of classes. Extracted components can be searched by keywords via a web browser such that the user can decide whether the query result component matches his/her requirements.

# Posters

## **A UML Profile for Service Oriented Architectures**

Rafik Salama, *American University in Cairo (AUC)*

Amir Zeid, *American University in Cairo (AUC)*

Service Oriented Computing is the new emerging paradigm for Distributed computing and e-business processing that is changing the way software applications are designed, architected, delivered and consumed. Engineering and modeling service-oriented architectures need extensions to existing modeling techniques and methodologies so we propose a UML profile for service-oriented architectures.

## **AJEER: An AspectJ-enabled Eclipse Runtime**

Martin Lippert, *it-wps GmbH & University of Hamburg*

The poster presents the combination of the world of Eclipse plugins and AspectJ, allowing developers to use AspectJ to modularize cross-plugin concerns into separately compiled aspect plugins. The runtime is based on the OSGi-runtime of Eclipse and features load-time weaving as well as adding and removing aspect plugins at runtime for the complete AspectJ language.

## **AOP as a First Class Reflective Mechanism**

Sergei Kojarski, *Northeastern University*

David Lorenz, *Northeastern University*

AOP is often perceived as a second class reflective mechanism, whereas reflection in OOP is considered first class. However, viewing AOP as a first class language mechanism is conducive to developing a general AOP model, which can be a basis for a unified AOP theory.

## **Advancements in Multicode Optimization**

Ben Stephenson, *University of Western Ontario*

Wade Holst, *University of Western Ontario*

In previous work, we have shown that multicores can be used to improve the performance of Java applications. We extend that work by both implementing more multicores and considering multicores of greater length. This has resulted in significantly larger performance gains than those that have been achieved previously for some benchmarks.

## **An Approach and Tools to Automate Externalization of Application Logic**

Hoi Chan, *IBM T.J. Watson Research Center*

Trieu Chieu, *IBM T.J. Watson Research Center*

Externalization of application logic from application objects has become pervasive in its use. Effort to install

decision points and lack of source code in legacy applications limit the potential use of this methodology. This report describes an approach to automate the process of externalizing application logic without modifying source code.

## **An Aspect-Oriented Generative Approach**

Uirá Kulesza, *Pontifical University Catholic of Rio de Janeiro*

Alessandro Garcia, *Pontifical University Catholic of Rio de Janeiro*

Carlos Lucena, *Pontifical University Catholic of Rio de Janeiro*

In this paper, we describe an aspect-oriented generative approach for the development of multi-agent systems. The generative approach is composed of: (i) a domain-specific language (DSL) that allows to model crosscutting and non-crosscutting agent features; (ii) an aspect-oriented architecture that offers a set of alternative design constructions for each of the agent features; and (iii) a code generator that maps abstractions of the Agent-DSL to specific compositions of objects and aspects in agent architectures. The use of aspect-oriented techniques in the definition of our generative approach brought benefits to the modeling and generation of crosscutting features since early design stages.

## **An Invitation to the Dance of Progress**

Lawrence Carleton, *BAE Systems, Inc.*

We evaluate events singly as gain or loss from status quo. We value sure gain, flee sure loss, assume progress cumulative. But progress relies on opportunism, taking advantage of tools and circumstances emerging through experience. Patterns, practices are superseded. Sometimes we refine, sometimes we invent, sometimes we stop using them.

## **An Open Model Infrastructure for Automotive Software**

Gabriel Voegler, *DaimlerChrysler AG/Research and Technology/Software Architectures*

Hajo Eichler, *Fraunhofer/Institute for Open Communication Systems*

Thomas Flor, *DaimlerChrysler AG/Research and Technology/Software Architectures*

Matthias Kasprovicz, *University of Applied Sciences Furtwangen*

To accommodate the growing demand for complexity in the automotive industry, an approach for a cross-process information model is introduced: the infrastructure presented aims at storing and relating key development

# Posters

data from the various software engineering phases. The model can be harmonized with proprietary development methodologies by means of metamodeling and serves as the starting point for the traceability of requirements, global plausibility checks, and (semi-)automatic transformation of model elements. Based on the medini tool chain, the model infrastructure is open with respect to the models and tools used thanks to the consistent deployment of OMG standards and the benefit of metamodeling.

## **B-UML: An Environment for Teaching UML in Smart Classrooms**

Amir Zeid, *The American University in Cairo*

Teaching object oriented software engineering through collaborative CASE tools can make the process much easier. We present B-UML which is a tool that captures use cases and Class diagrams in a smart class room. The diagrams are then converted to real class diagrams and integrated in a CASE tool.

## **Best Practices and Methodologies in Service-Oriented Architectures: Paving the Way to Web-services Success**

Amir Zeid, *The American University in Cairo*

Service Oriented Computing (SOC) is the new emerging paradigm for distributed computing and e-business processing that is changing the way software applications are designed, architected, and consumed. The main goal of this poster is to identify and promote best practices to properly engineer web-services and SOA.

## **Best Practices for Model-Driven Software Development**

Jorn Bettin, *SoftMetaWare, New Zealand*

Ghica van Emde Boas, *Independent Consultant, The Netherlands*

Aditya Agrawal, *Vanderbilt University, USA*

Markus Völter, *Independent Consultant, Germany*

Jean Bézivin, *University of Nantes, France*

Model-Driven Software Development ([www.mdsd.info](http://www.mdsd.info)) is a paradigm with roots in software product line engineering. The proliferation of model-driven development tools has led to an increasing need for solid methodological support for model-driven approaches. What sets MDSD apart from classical software product line engineering is the emphasis on a highly agile development process.

## **Bottleneck Analysis in Java Applications using Hardware Performance Monitors**

Dries Buytaert, *Ghent University*

Andy Georges, *Ghent University*

Lieven Eeckhout, *Ghent University*

Koen De Bosschere, *Ghent University*

We present MonitorMethod, a tool to gain insight in the behavior of Java applications. MonitorMethod instruments the application and relates hardware performance monitors to the source code. The case study shows that linking microprocessor-level performance characteristics to the source code is helpful for identifying performance bottlenecks and their causes.

## **Building Software for Pervasive Computing**

Cristina Lopes, *University of California, Irvine*

Steffen Schaefer, *IBM*

Siobhán Clarke, *Trinity College Dublin*

Anind Dey, *University of California, Berkeley*

Tzilla Elrad, *Illinois Institute of Technology*

Robert Filman, *NASA Ames Research Center*

Pervasive, or Ubiquitous, Computing envisions a world of communicating small devices, sensors and actuators. OOP owes its popularity in part to the analogy between virtual and physical objects. Physical objects abound in Pervasive Computing, and object technology is playing a major part in pervasive computing, with objects living on even the smallest device. But OOP issues such as contextual programming requirements, dealing with multiple concerns, and the "mass" nature of ubiquitous devices are examples of where object technology needs extension, adaptation or replacement. This workshop will bring together practitioners involved in the development of pervasive computing solutions and researchers devising new programming models in this area. The goals are (1) to identify recurring architectures and patterns typically used to build such systems, (2) to raise issues and gaps in design and implementation; and (3) to discuss new ideas and changes to object technology to better support Pervasive Computing.

## **C++, Boost, and the Future of C++ Libraries**

Jeff Garland, *CrystalClear Software*

Nicolai Josuttis, *Solutions In Time*

Kevlin Henney, *Curbralan*

Jeremy Siek, *Indiana University*

The C++ standard library provides a set of classes that greatly simplifies software development. However, it is not all that is needed in day-to-day programming. In this workshop we will work on other libraries that might

# Posters

extend C++. The poster will reflect the results of the workshop.

## **Challenges in Outsourcing and Global Development: How Will Your Job Change?**

Dennis Mancl, *Lucent Technologies*  
William Opdyke, *North Central College, Naperville, IL*  
Steven Fraser, *Consultant*

If you are a software professional, your job is changing. Outsourcing and global development affect many things in our work environment: what we build, how and when we communicate, and how we prepare ourselves for the future.

## **CoSMIC: Addressing the Deployment and Configuration Crosscutting Concerns of Distributed Real-time and Embedded Systems**

Aniruddha Gokhale, *Vanderbilt University*  
Krishnakumar Balasubramanian, *Vanderbilt University*  
Tao Lu, *Vanderbilt University*  
Jaiganesh Balasubramanian, *Vanderbilt University*  
Arvind Krishna, *Vanderbilt University*  
Gan Deng, *Vanderbilt University*

The Object Management Group has adopted the Deployment and Configuration specification to address crosscutting challenges of deployment and configuration of distributed real-time and embedded component middleware-based systems. This paper describes a model-based generative programming tool suite called CoSMIC that resolves these accidental complexities and crosscutting concerns.

## **Component and Middleware Performance (Workshop Report)**

Petr Tuma, *Charles University, Czech Republic*  
Paul Brebner, *University College London, United Kingdom*  
Emmanuel Cecchet, *ObjectWeb, INRIA Rhone-Alpes, France*

The poster presents results of the OOPSLA 2004 Workshop on Component and Middleware Performance. The workshop follows its predecessor in getting together people who are involved in performance evaluation, to identify the most significant obstacles encountered in the current practice and propose approaches to tackle them.

## **Cona -- Aspects for Contracts and Contracts for Aspects**

Therapon Skotiniotis, *North Eastern University*  
David Lorenz, *North Eastern University*

We present an aspect-oriented Design by Contract (DBC) tool for Java named Cona. We also extend the use of DBC and assertions to AOP. Aspects are used in the implementation of contracts, and contracts are used for enforcing assertions on aspects.

## **Dependable Distributed Systems - A Research Project Overview**

Alexander Szep, *Vienna University of Technology, Institute of Computer Technology*  
Robert Smeikal, *Vienna University of Technology, Institute of Computer Technology*  
Martin Jandl, *Vienna University of Technology, Institute of Computer Technology*  
Karl Goeschka, *Vienna University of Technology, Institute of Information Systems*

Complex distributed systems require dependability encapsulated in coherent system parts. We achieve fault-tolerance through replication and optimize dependability by trading availability against constraint consistency in the case of site failure or network partition. A real-life telecommunication application proves the concept by using synchronous and asynchronous replication techniques.

## **Deriving Refactorings For AspectJ**

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

We present aspect-oriented programming laws that are useful for deriving refactorings for AspectJ. The laws help developers to verify if the transformations they define preserve behavior. We derived several AspectJ refactorings. Our laws are also useful for restructuring applications with the aim of using aspects to modularize crosscutting concerns.

## **Design Pattern Integrated Tool**

Somsak Phattarasukol, *California State Polytechnic University, Pomona USA*  
Daisy Sang, *California State Polytechnic University, Pomona USA*

The use of design patterns has been limited due to the lack of an integrated supporting tool. Several were built but rarely used in practice because of technical

# Posters

restrictions. We developed an integrated tool that supports language and platform independence, and cross-platform interoperability with a validation features for quality assurance.

## **Early Aspects Workshop - Ideas and Results**

Elisa Baniassad, *Trinity College Dublin*  
Ana Moreira, *Universidade Nova de Lisboa*  
Awaish Rashid, *Lancaster University, UK*  
Bedir Tekinerdogan, *University of Twente*  
Paul Clements, *Carnegie Mellon University*  
João Araújo, *Universidade Nova de Lisboa*

"Early aspects" is an emerging research area of Aspect-Oriented Software Development, focusing on managing aspects during requirements engineering and architecture design. This poster reflects the ideas and results of the Early Aspects workshop, which focused on providing a forum for the discussion of approaches, for management of early-aspects.

## **Eclipse Technology eXchange (ETX) Workshop**

Michael Burke, *IBM T.J. Watson Research Center*  
Bjorn Freeman-Benson, *University of Washington*  
Margaret Storey, *University of Victoria*

The Eclipse platform (<http://www.eclipse.org>) is designed for building integrated development environments (IDEs) for object-oriented application development. We will feature presentations and group discussions concerning the use of the Eclipse open source as a code base for teaching and research. The presenters will report on their projects and discuss potential new uses of Eclipse.

## **Extravagaria: Art Assisting Science**

Richard Gabriel, *Sun Microsystems*  
Janet Holmes, *Boise State University*  
Kirstie Bellman, *The Aerospace Corporation*

Artists and scientists share the most important human goal: the search for truth. Their methods and approaches differ, but there is no a priori reason to grant primacy to either. In software, we not only seek truth and reality, we create it. Certainly there is a role for art here.

## **Extreme Construction**

Joseph Bergin, *Pace University*  
Fred Grossman, *Pace University*

Extreme Construction is a training methodology for Extreme Programming (XP), utilizing art materials. A team of 10 "constructors" creates an artifact under direction of a "customer". It is accessible to programmers,

educators, managers, and customers. Most practices of XP are covered. It has been used in education and industrial training.

## **GMT - an Open Source MDA Tool Initiative, status and progress**

Ghica van Emde Boas, *Bronstee.com Software & Services*  
Jorn Bettin, *SoftMetaWare*  
Frederic Jouault, *University of Nantes, France*  
Mark Kofman, *Royal Institute of Technology at Stockholm, Sweden*

Generative Model Transformer - an Open Source MDA tool initiative. Hosted at: [www.eclipse.org/gmt](http://www.eclipse.org/gmt). Report on the progress and status of the GMT project.

## **Gumption Traps**

Rachel Davies, *Agile Alliance*  
Ivan Moore, *ThoughtWorks*

Motivation is possibly the most important factor for the success of any software project. The book "Zen and the Art of Motorcycle Maintenance" introduces the idea of "Gumption Traps"; things that sap motivation, such as not having the correct tools. This poster is the output from the "Gumption Traps" workshop at OOPSLA 2004. A workshop that introduces participants to techniques from Systems Thinking for the identification and understanding of Gumption Traps, which will lead to suggestions for how to avoid or combat them. Once Gumption Traps can be identified and removed, improvements in motivation can lead to improvements in the well-being of team members and improvements in productivity.

## **Improving Quality in Conceptual Modeling**

Tauqeer Hussain, *LUMS*  
Shafay Shamail, *LUMS*  
Mian Awais, *LUMS*

This poster describes a new approach to improve the semantic and pragmatic quality of a conceptual model through schema transformations. The quality is improved by introducing rules based on a new semantic quality parameter: extent of normalization, and two new concepts of Single-Valued Relationship Attributes and Multi-Valued Relationship Attributes.

# Posters

## **J2EE for the Public Administration: a Success Story**

Maurizio Lancia, *Italian National Research Council*  
Paola Garzenini, *Italian National Research Council*  
Alessio Marchetti, *Italian National Research Council*  
Roberto Puccinelli, *Italian National Research Council*

Our poster shows how J2EE technologies, well-known design patterns and standard methodologies have successfully been applied in building a complete, robust and well-documented accounting application. Our experience demonstrates that those three factors allow for easy maintenance and reuse of components.

## **JIVE: Java Interactive Visualization Environment**

Paul Gestwicki, *University at Buffalo*  
Bharat Jayaraman, *University at Buffalo*

JIVE represents a novel approach to runtime visualization and analysis of Java programs. It facilitates program understanding and interactive debugging, featuring: multiple, customizable views of object structure; representation of execution history via sequence diagrams; interactive queries on runtime behavior; forward and reverse interactive execution. JIVE uses standard JVM and compilers.

## **Legacy Transformation: Capturing Business Knowledge from Legacy Systems**

Jean Bézivin, *Group (INRIA & LINA), University of Nantes*

Sergio de Cesare, *Brunel University*

Grant Holland, *Sun Microsystems*

Mark Lycett, *Brunel University*

Chris Partridge, *Brunel University*

Legacy transformation is a significant problem that modern organizations currently need to cope with. Legacy systems are source of significant corporate knowledge (relating to concepts such as objects, rules, events, processes and services) which in many cases is not properly documented. The source code and data may represent the only source from which business models can be reverse engineered. Current transformation strategies are limited in number and mainly address the technological aspects. The primary concern of current approaches is migrating functionality and data from an obsolete technology platform to a new system developed with state-of-the art technology. Minor or no emphasis is given to the reverse engineering of business knowledge from the legacy systems and its representation in technology-agnostic business models. There is a growing need in the software development community for methods and techniques that address two complementary

problems. Firstly, the reverse engineering of corporate/business knowledge from legacy systems in a traceable form. Secondly, being able to represent business requirements in models that can effectively trace to software models. From the perspective of Model Driven Architecture (MDA) solving this second problem would imply being able to represent computational independent models (CIM) and map these to platform independent models (PIM).

## **Managing Variabilities consistently in Design and Code**

Christa Schwanninger, *Siemens AG*

Danilo Beuche, *pure-systems GmbH*

Krzysztof Czarnecki, *University of Waterloo*

Mira Mezini, *Darmstadt University of Technology*

Markus Voelter, *ingenieurbüro für softwaretechnologie*

The poster will cover the result of the workshop "Managing Variabilities consistently in Design and Code". With the introduction of software product lines into the practice, variants and variability add a new dimension of complexity to the software development process. The goal is to identify efficient ways for dealing with highly variable software systems on design and code level by evaluating existing approaches and new ideas from the research community and industrial practice.

## **Meta: Augmenting and Unifying Object-Oriented (and other) Languages**

Wade Holst, *University of Western Ontario*

Meta is an ambitious research project whose overall purpose is to increase the utility and expressive power of a wide range of existing languages. Meta provides augmented versions of existing languages and guarantee support for aspects, components, language interoperability, visualization, reflection, various inheritance models, and many other extensions.

## **Mobile Musical Agents - The Andante Project**

Leo Ueda, *University of São Paulo*

Fabio Kon, *University of São Paulo*

We are investigating the use of mobile agent technology for the creation of music within a distributed computing environment. We believe this technology has the potential to foster new ways of making music. This poster presents Andante, an open-source infrastructure for building distributed musical applications based on mobile agents.

# Posters

## **Mobile-D: An Agile Approach for Mobile Application Development**

Pekka Abrahamsson, *VTT Technical Research Centre of Finland*

Antti Hanhineva, *VTT Technical Research Centre of Finland*

Hanna Hulkko, *VTT Technical Research Centre of Finland*

Tuomas Ihme, *VTT Technical Research Centre of Finland*

Juho Jaalinoja, *VTT Technical Research Centre of Finland*

Mikko Korkala, *University of Oulu*

Juha Koskela, *VTT Technical Research Centre of Finland*

Pekka Kyllonen, *VTT Technical Research Centre of Finland*

Outi Salo, *VTT Technical Research Centre of Finland*

Development of mobile applications is difficult due to mobile-development specific demands and technical constraints. To overcome these challenges, we have developed an agile development approach called the Mobile-D. The Mobile-D approach is briefly outlined and the experiences gained from four case studies are presented.

## **Model Synchronization as a Problem of Maximizing Model Dependencies**

Igor Ivkovic, *University of Waterloo*

Kostas Kontogiannis, *Technical University of Crete*

During the course of its evolution, software is modified through models at different levels of abstraction. To enable systematic development and maintenance, related models need to be kept synchronized. In this poster, we present our view of model synchronization as a problem of maximizing model dependencies.

## **Modeling and Implementing Software Architecture with Acme and ArchJava**

Jonathan Aldrich, *Carnegie Mellon University*

David Garlan, *Carnegie Mellon University*

Bradley Schmerl, *Carnegie Mellon University*

Tony Tseng, *Carnegie Mellon University*

Software architecture describes the high-level organization of a software system, and is essential both for reasoning about system properties and for implementing and evolving code. This poster describes two architecture-related tools: AcmeStudio for architectural modeling and analysis, and ArchJava for ensuring consistency between architecture and implementation.

## **Modeling Event Driven Applications With a Specification Language (MEDASL)**

Murali Kaundinya, *Client Services, Sun Microsystems, Inc.*

Syed Ali, *Client Services, Sun Microsystems, Inc.*

RFID technologies and Sensor networks necessitate the retrofitting of existing applications and development of new applications rapidly. We need efficient techniques to develop bug-free applications quickly. DASL, a practical realization of the MDA vision, helps modelling the RFID application domain correctly and deploying such applications on different target platforms.

## **OOLACA: An Object Oriented Library for Abstract and Computational Algebra**

Virginia Niculescu, *Babes-Bolyai University, Faculty of Mathematics and Computer Science*

Grigoreta Sofia Moldovan, *Babes-Bolyai University, Faculty of Mathematics and Computer Science*

We develop a library that allows working with concrete and abstract algebraic structures. The advantages are given mainly by the creational design patterns, reflection, dynamic loading, and storage independence. We may work with abstract algebraic structures, define new algebraic structures, and operate with them in abstract and concrete ways.

## **Objects in Large Distributed Applications (OLDA-III) workshop poster**

Peter Dickman, *University of Glasgow*

Karen Renaud, *University of Glasgow*

Huw Evans, *Kelvin Institute*

Objects in Large Distributed Applications (OLDA-III) workshop overview, integrating development & deployment technologies across all system levels. Graphically links problems and possible solutions across: distributed systems infrastructure, middleware, language support, software architecture & engineering methodologies, ubiquitous computing and application demands. Incorporates historical perspectives and annotations drawn from previous OLDA meetings.

## **On the Relative Advantages of Teaching Web Services in J2EE vs. .NET**

Edward Gehringer, *North Carolina State University*

Sandeep Kachru, *Blackbaud, Inc.*

J2EE and .NET, the two leading Web-services platforms, provide comparable development environments through rich and extensive APIs and powerful tools. The choice between them will often depend on local factors, such as

# Posters

languages and OSs used in teaching. We provide a flowchart to help educators make an informed decision.

## **Ontologies as Software Engineering Artifacts - Workshop at OOPSLA 2004**

Joerg Pechau, *CoreMedia AG*

Petra Becker-Pechau, *University of Hamburg*

Ralf Klischewski, *German University in Cairo*

Martin Lippert, *it-wps Workplace Solutions Ltd.*

The poster summarizes results of the workshop "Ontologies as Software Engineering Artifacts" for OO-systems. It reflects (1) submissions & examples, (2) identified current trends and fields of interest, (3) discussion summary (especially application), and impact on software engineering, adequacy and feasibility of the approaches, (4) information for launching an ongoing discussion and forming a community.

## **Patterns in Retrospectives**

Linda Rising, *Independent Consultant*

Mary Lynn Manns, *University of North Carolina at Asheville*

Retrospectives are important, not only for object technology, but for any human activity. Our workshop will continue documenting patterns for retrospectives and proposing an initial pattern language structure. The poster will present the skeleton outline of the fledgling pattern language and the patterns that will have been workshopped.

## **Predicting Performance of EJB-based Systems Using Layered Queuing Networks**

Alexandre Oufimtsev, *Performance Engineering Lab/Dept. of Computer Science/University College Dublin*

Liam Murphy, *Performance Engineering Lab/Dept. of Computer Science/University College Dublin*

Performance is one of the crucial factors for large-scale component-oriented development. We propose a method for predicting the performance of EJB-based systems based on Layered Queuing Networks. We argue that only a detailed model of the EJB infrastructure along with the application model provides accurate performance results.

## **Pseudo-Classes: Very Simple and Lightweight MockObject-like Classes for Unit-Testing**

Geoff Sobering, *Berbee*

Levi Cook, *Isthmus Group*

Steve Anderson, *Berbee*

A simple alternative to MockObjects is presented. Given an interface, a Pseudo-Class is created implementing all methods such that they immediately fail. A test-specific implementation can then be extended from the Pseudo, over-riding only the methods required by the test. This mechanism is adequate for most tests.

## **QuA: Building with Reusable QoS-Aware Components**

Frank Eliassen, *Simula Research Laboratory*

Richard Staehli, *Simula Research Laboratory*

Gordon Blair, *Lancaster University*

Jan Oyvind Aagedal, *SINTEF*

The QuA project team at Simula Research Laboratory has prototyped an open source component middleware platform to support QoS-sensitive applications. We are investigating the design of reusable QoS management components and tools to enable rapid construction of complex distributed systems.

## **RAIL: Code Instrumentation for .NET**

Bruno Cabral, *CISUC - University of Coimbra*

Paulo Marques, *CISUC - University of Coimbra*

Luís Silva, *CISUC - University of Coimbra*

The Runtime Assembly Instrumentation Library (RAIL) is one of the first frameworks to implement code instrumentation in the .NET platform. RAIL gives the programmer an object-oriented vision of the code of an application, allowing assemblies, modules, classes, references and even intermediate code (IL) to be easily manipulated.

## **RUBiS Revisited: Why Benchmarking J2EE Applications is Hard**

Jaime Spacco, *University of Maryland*

Bill Pugh, *University of Maryland*

We have replicated the experiments of Cecchet et al. detailed in "Performance and Scalability of EJB Applications" at OOPSLA '02. We report on our experiences configuring, deploying and tuning Enterprise software, and provide evidence that many of the conclusions of the original work are misleading or cannot be generalized.

# Posters

## **Reflective Composition: the Declarative Composition of Roles to Unify Objects, Roles, & Aspects**

Simon Holland, *The Open University, Milton Keynes, United Kingdom*

As bases for object-orientation, both class-based and prototype-based organization have limitations. We argue that Roles have significant benefits as a foundation for organizing objects, and that these benefits can be realised most flexibly using logic meta-programming. This makes it possible to reduce redundancy and subsume aspects.

## **Results of the Second Workshop on Method Engineering for Object-Oriented and Component-Based Development**

Cesar Gonzalez-Perez, *University of Technology, Sydney*  
Donald Firesmith, *Software Engineering Institute*  
Brian Henderson-Sellers, *University of Technology, Sydney*  
Pavel Hruby, *Microsoft*  
Dilip Patel, *South Bank University*  
Dan Rawsthorne, *Net Objectives*  
Bernhard Rumpe, *Munich University of Technology*  
Magdy Serour, *University of Technology, Sydney*  
Hadar Ziv, *eBuilt, Inc.*

This poster presents the outcomes and ideas gathered during the Second Workshop on Method Engineering for Object-Oriented and Component-Based Development. The main theme of the workshop is "putting the fragments together", i.e. how to construct a customized methodology from a collection of method fragments.

## **Results of the Third International Workshop on Agent-Oriented Methodologies**

Cesar Gonzalez-Perez, *University of Technology, Sydney*  
Paolo Bresciani, *ITC-Irst*  
John Debenham, *University of Technology, Sydney*  
Brian Henderson-Sellers, *University of Technology, Sydney*  
Graham Low, *University of NSW*

This poster presents the outcomes and ideas gathered during the Third International Workshop on Agent-Oriented Methodologies. The main theme is the relationships between object-orientation and agent-orientation from a methodological perspective, i.e. what these relationships are and how AO can be enriched from OO knowledge and experience.

## **SODA: A Stability-Oriented Domain Analysis Method**

Haitham Hamza, *University of Nebraska-Lincoln*

In work to date, domain analysis methods do not explicitly emphasize stability. In this poster, we present the SODA (Stability-Oriented Domain Analysis) method, a domain analysis method that embraces stability as a driver for domain analysis. We give an overview and example of the method.

## **The 4th Workshop on Domain-Specific Modeling**

Juha-Pekka Tolvanen, *MetaCase*  
Jonathan Sprinkle, *University of California, Berkeley*  
Matti Rossi, *Helsinki School of Economics*

Domain-Specific Modeling (DSM) raises the level of abstraction beyond programming by specifying the solution directly using domain concepts. This poster introduces DSM by showing examples from various fields of model-based software development. It presents the results and research themes of the 4th OOPSLA workshop on Domain-Specific Modeling.

## **The Bots 'n Pieces Pattern Language Experiment**

Alan O'Callaghan, *De Montfort University*  
Maria Kavanagh, *De Montfort University*

The Bots 'n Pieces Pattern Language experiment investigates the feasibility of Alexanderian Pattern Languages for software systems. Patterns have been mined from workshops where teams use agile techniques to create Lego Mindstorms robots. The resulting Language can then be tested to see if 'whole systems' can be generated from it.

## **The Customer Role in Agile Projects**

Jens Coldewey, *Coldewey Consulting*  
Klaus Marquardt, *Dräger Medical*  
Mary Poppendieck, *Poppendieck.LLC*

One of the values of agile methodologies is to satisfy the customer through frequent delivery of working, useful software. The claim is that agile projects yield better customer satisfaction than traditional processes. However, the customer role in agile methodologies is often perceived to be nebulous. There are hints, such as the regular Sprint Meeting in Scrum or the Customer on Site in XP, but more questions are raised than answered: Are customer and client the same? Are there any assumptions about the corporate culture of the client? Are there any restrictions on the legal construction of the client-vendor-relationship? Is the customer one person or a team? And so on. On Sunday we will present a workshop aimed at

# Posters

exploring the customer role further in a highly interactive setting. The results of that workshop will be presented in poster format.

## **The "Killer Examples" for Design Patterns and Objects First workshop**

Carl Alphonse, *University at Buffalo, State University of New York*  
Dung Nguyen, *Rice University*  
Stephen Wong, *Rice University*  
Phil Ventura, *State University of West Georgia*

A killer app "makes a sustaining market for a promising but under-utilized technology." [The Jargon File] A killer example provides clear and compelling motivation for some concept. This poster reports results from the third annual OOPSLA "Killer Examples" workshop exploring killer examples for teaching design patterns and objects-first.

## **Towards a Framework for the General Intensional Programming Compiler in the GIPSY**

Joey Paquet, *Concordia University*  
Peter Grogono, *Concordia University*  
Aihua Wu, *Concordia University*

This paper presents a framework to develop the General Intensional Programming Compiler in the development of a programming system. We adopt a technique with automatic generation of framework hot spots from user-defined specifications of the syntax and semantics of novel members of the Lucid family of intensional programming languages.

## **Visualisation for Learning OOP, Using Aspects and Eclipse**

Rilla Khaled, *Victoria University of Wellington*  
James Noble, *Victoria University of Wellington*  
Leo Ferres, *Carleton University*  
Judy Brown, *Carleton University*  
Robert Biddle, *Carleton University*

This poster outlines our project to help beginners learn to program by showing object visualisations driven by aspect-oriented programming, and presented as part of the Eclipse development platform. The aspect-oriented programming is part of the infrastructure used to drive the visualisations that help students learn object-oriented programming.

## **WSAgent: an Agent Based in Web Services to Promote Interoperability Between Heterogeneous Systems in Health Domain**

Leticia Rheinheimer, *Interdisciplinary Postgraduation Program on Applied Computing – UNISINOS*  
Júnior Martins, *Interdisciplinary Postgraduation Program on Applied Computing – UNISINOS*  
Sérgio Crespo C. S. Pinto, *Interdisciplinary Postgraduation Program on Applied Computing – UNISINOS*

This paper describes a Software Agent called WSAgent, which combines technologies such as Web Services, Frameworks and Design Patterns in the construction of a bind to grant interoperability, reuse and flexibility between heterogeneous environments in health domain.

## **Workshop: Revival of Dynamic Languages (RDL 2004)**

Gilad Bracha, *Sun Microsystems*  
Wolfgang De Meuter, *Vrije Universiteit Brussel*  
Stéphane Ducasse, *University of Bern*  
Oscar Nierstrasz, *University of Bern*

In industry, static languages (such as Java, C++ and C#) are much more widely used than their dynamic counterparts (like CLOS, Python, Self or Smalltalk). So it appears as though dynamic language concepts were forgotten and lost the race. But this is not the case. Java and C#, the latest mainstream static languages, popularized to a certain extent dynamic language features such as garbage collection, portability and (limited forms of) reflection. In the near future, we expect this dynamicity to increase even further. E.g., it is getting clearer year after year that pervasive computing is becoming the rule and that concepts such as meta programming, reflection, mobility, dynamic reconfigurability and distribution are becoming increasingly popular. All of these features are the domain of dynamic languages, and hence it is only logical that more dynamic language concepts have to be taken up by static languages, or that dynamic languages can make a breakthrough. The goal of this workshop is to bring together researchers in (object-oriented) language design interested in dynamic language features and languages that do not fit into the static mainstream.

# Posters

## ***iXj: Interactive Source-to-Source Transformations for Java***

Marat Boshernitsan, *University of California, Berkeley*  
Susan Graham, *University of California, Berkeley*

Manual large-scale modification of source code can be tedious and error-prone. Integrating scriptable program transformations into development environments will assist developers with this task. We discuss various usability issues of bringing transformations to end-users and describe a developer-oriented interactive transformation tool that we are building.

# The ACM SIGPLAN Student Research Competition

Chair: Torsten Layda, SWX Swiss Exchange



After its remarkable success in 2002 and 2003, OOPSLA is again hosting the ACM SIGPLAN Student Research Competition. The first round of evaluations will be held jointly with the OOPSLA Poster Session. The Student Research Competition shares the posters' goal of having students meet and interact with researchers so that both sides have the opportunity to learn of ongoing, current research. The Student Research Competition has the additional goal of helping students gain experience with formal presentations and evaluations.

Submissions have been reviewed by a committee and students have been chosen to participate in the competition. These students will discuss their work with competition judges (and conference attendees) during the regular Poster Session. The highly interactive Poster Session is used to evaluate the research on its quality, uniqueness, significance of the work, and the clarity of the presentation and the informal discussion. The semi-finalists chosen from this round of competition will present their research during a conference session. The ACM Student Research Competition winners will be recognized prior to the keynote speaker on the last day of the conference.

**Monday, 17:30-19:30, Vancouver Convention Centre**  
**Student Research Presentations at the Welcome Reception**

## ***A Language-Independent Approach to Software Maintenance Using Grammar Adapters***

Suman Roychoudhury, *University of Alabama at Birmingham*

**Tuesday, 15:30-17:00, Meeting Rooms 1-3**  
**Student Research Finalist Presentations**

The research presented in this poster illustrates the concept of abstract transformation rules that enable design maintenance in legacy-based systems. A generic platform is described that uses grammar adapters to lay a common foundation for the construction of new software development tools.

**Thursday, 8:30-8:35, Ballroom A-B**  
**Student Research Award Presentation**

## ***A Framework for Detecting, Assessing and Visualizing Performance Antipatterns in Component Based Systems***

Trevor Parsons, *Dublin City University*

Component-based enterprise systems often suffer from performance issues as a result of poor system design. In this paper, we propose a framework to automatically detect, assess and visualize poor system design, from a performance perspective, by analyzing run-time data using data mining techniques.

## ***A Lightweight LTL Runtime Verification Tool for Java***

Eric Bodden, *RWTH Aachen University*

We propose a runtime verification tool for Java 1.5 that introduces certain metadata annotations in order to parameterize an AspectJ backend that is in combination capable of verifying formulas in LTL against the original Java application. The system preserves full Java 1.5 compliance of the source code.

## ***A Framework for Removing Redundant Context Management Services in Enterprise JavaBeans Application Servers***

Mircea Trofin, *Dublin City University*

We propose a framework for removing redundant context management services in contextual composition frameworks, with focus on Enterprise JavaBeans. It is expected that by applying our framework, performance can be improved without the loss of modularity.

## ***Automatic Identification of Common and Special Object-Oriented Unit Tests***

Tao Xie, *University of Washington at Seattle*

Common and special tests can be created to exercise some common and special behavior of the class under test, respectively. This research develops a novel approach for automatically identifying common and special unit tests from automatically generated tests for a class without requiring any specification.

# The ACM SIGPLAN Student Research Competition

## **Empirical Investigation of the Impact of Extreme Programming Practices on Software Projects**

Lucas Layman, *North Carolina State University*

Extreme Programming (XP) is an agile software development methodology composed of several practices that purportedly yield high quality and high customer satisfaction. However, there has been a lack of formal investigation of these claims. We conduct empirical, industrial case studies to evaluate XP. Results from two case studies are presented.

## **Enhancing Quality of Service for Distributed Object Computing Middleware**

Arvind Krishna, *Electrical Engineering and Computer Science Dept, Vanderbilt University*

This poster showcases research advances towards enhancing the QoS of middleware for Distributed Real-time and Embedded systems including: Higher level programming language abstractions that reduce development complexities, Software processes to identify configuration patterns that address system configuration challenges and Partial evaluation techniques that customize implementations according to operational contexts.

## **Identification of Reusable Components Within an Object-Oriented Software System Using Algebraic Graph Theory**

Spiros Xanthos, *University of Macedonia*

A novel method for identifying dense communities of classes (clusters) within an Object-Oriented system has been developed. Such communities might possibly imply relevance of functionality and thus be used as Reusable Components. To accomplish this we employ Algebraic Graph Theory (Spectral Graph Partitioning Techniques).

## **NGMF: A Generic Framework for Constructing Graph-Based Systems**

Hadi Salimi, *Nebras Informatics Co.*

Many branches of science rely on graphs for representing a wide variety of objects. It is therefore reasonable to have a framework for constructing this kind of applications. Nebras Graph Management Framework (NGMF) is a layered client-server framework for rapid development of these applications.

## **Refining Designs along Middleware-Specific Concern-Dimensions at Different MDA-Levels of Abstraction**

Raul Silaghi, *Swiss Federal Institute of Technology in Lausanne, Switzerland*

For the MDA approach to software development to become a reality for distributed enterprise systems, MDA needs to provide both modeling support for middleware-specific concerns and tool support for helping developers refine their designs along such concern-dimensions at different MDA-levels of abstraction. In order to address these issues, the MDA-compliant Enterprise Fondue method proposes a hierarchy of UML profiles as a means for addressing middleware-specific concerns at different MDA-levels of abstraction, along with model transformations to incrementally refine existing design models according to the proposed profiles. Tool support is provided through the Parallax framework, which enables developers to modularize crosscutting concerns into aspect-promoting Eclipse plug-ins.

## **Reverse Engineering of UML Specifications from Java Programs**

Martin Keschenau, *RWTH Aachen University*

An important issue in software engineering is the reverse engineering, that is the generation of specifications from programs. In this work, we will sketch an intuitive approach for constructing UML class diagrams from Java bytecode which still allows analysing features beyond the mere enumeration of fields and methods.

## **Supporting Software Evolution through Model-Driven Program Transformation**

Jing Zhang, *The University of Alabama at Birmingham*

This poster presents two research objectives for supporting software evolution in Model-Driven Software Development. First, Model-Driven Program Transformation technique is introduced to support legacy system evolution. The second objective involves the application of a program transformation engine to automate model interpreter evolution driven by meta-model schema changes.

# Demonstrations

Chair: Neil B Harrison, Avaya



OOPSLA demonstrations provide an opportunity for companies and universities to show their latest work to an experienced audience. This can be work in progress, commercial applications, proof of concepts, results of academic research, tools, systems or any topic that has interesting object-oriented aspects. Demonstrations are not focused on selling a product but to highlight, explain and present the technical aspects of it. Demonstrators may actively solicit feedback from the usually very technically savvy audience. In the past, this has made for some very interesting demonstration sessions.

**Tuesday 10:30-11:15, Courtyard**  
**Thursday 11:30-12:15, Courtyard**

## **1 Refactoring: to the Rubicon...and Beyond!**

Roly Perera, *Ergnosis Ltd.*  
Russ Freeman, *Ergnosis Ltd.*  
Phil Webb, *Ergnosis Ltd.*

Software development involves a continual process of refactoring to accommodate new features and consolidate existing functionality. Without this ongoing maintenance effort, entropy rapidly takes hold and delivering further features or bug-fixes becomes increasingly difficult.

Interest in software refactoring and tools for automating or assisting with this activity has been growing steadily over the last decade, thanks to the influential efforts of Roberts and Brant, Opdyke, Fowler and others. However the refactorings which have been discussed to date, such as Extract Method, are in desperate need of decomposition into more primitive, but more widely applicable refactorings, such as Push Code Into Method.

We will demonstrate this new approach to refactoring through a powerful new tool which supports these refactoring primitives directly. By understanding all refactorings as the introduction or elimination of degrees of freedom we can see how a large proportion of programming edits are in fact micro-refactorings, and gain an insight into how tools that support these micro-refactorings will have a dramatic impact on developer productivity.

**Tuesday 12:30-13:15, Courtyard**  
**Wednesday 11:30-12:15, Courtyard**

## **2 jMock. Supporting Responsibility-Based Design with Mock Objects**

Steve Freeman, *ThoughtWorks*  
Tim Mackinnon, *ThoughtWorks*  
Nat Pryce, *ThoughtWorks*  
Joe Walnes, *ThoughtWorks*

jMock is a dynamic Mock Object framework that supports responsibility-based design when programming test-first.

Test-Driven Development with Mock Objects is a technique for concentrating on the relationships between objects. It guides the programmer to discover the precise dependencies that an object has on its neighbours, and the interfaces that describe those dependencies. This is much more powerful than the usual use of Mock Objects which is to support development against third-party libraries. jMock allows the programmer great flexibility in describing interactions between objects. We have also taken care to develop an API that makes tests readable and self-explanatory; in effect, we have written a domain-specific language for testing that sits in Java and C#.

In this demonstration we will show the flow of the mock object development process. We will show how using mock objects to guide your design results in a more effective form of test driven development and more flexible code; how mock objects allow you to concentrate more on end-user requirements than on infrastructure; and how the objects in the resultant code are small and loosely coupled, with well-defined responsibilities. We will also provide a tour of the structure of the jMock libraries.

# Demonstrations

**Tuesday 12:30-13:15, Courtyard**

**Wednesday 15:30-16:15, Courtyard**

## **3 Modeling and Implementing Software Architecture with Acme and ArchJava**

Jonathan Aldrich, *Carnegie Mellon University*

Software architecture describes the high-level organization of a software system, and is essential both for reasoning about system properties and for implementing and evolving code. This demonstration will describe two architecture-related tools: AcmeStudio for architectural modeling and analysis, and ArchJava for ensuring consistency between architecture and implementation.

AcmeStudio is an architectural design tool based on Eclipse, supporting graphical and textual descriptions of software architecture as well as various forms of architectural analysis. We will show how to create an architecture in Acme and check it against the constraints of an architectural style such as pipe-and-filter.

ArchJava extends the Java language to include explicit architectural modeling constructs, and uses type system techniques to ensure that the implementation conforms to the architecture. We will show how AcmeStudio can generate skeleton ArchJava code, how developers can fill in this architecture with an implementation using an Eclipse-based ArchJava IDE, and how ArchJava's type system can help the developer to maintain consistency between code and architecture.

Additional information about ArchJava and AcmeStudio is available at <http://www.archjava.org/> and <http://www.cs.cmu.edu/~acme/AcmeStudio/>.

**Tuesday 15:30-16:15, Courtyard**

**Wednesday 12:30-13:15, Courtyard**

## **4 Advanced Refactorings in Eclipse**

Robert Fuhrer, *IBM T.J. Watson Research Center*  
Frank Tip, *IBM T.J. Watson Research Center*  
Adam Kiezun, *MIT*

We propose a demonstration of several advanced refactorings for Java implemented for the Eclipse IDE ([www.eclipse.org](http://www.eclipse.org)). These refactorings are semantics-preserving program transformations typical of the transformations object-oriented programmers perform manually to improve the structure of existing programs, and promote reuse, clarity, and extensibility.

This demonstration consists of three parts: (i) a review of Eclipse's implementations of several common refactorings, including Extract Method, Introduce Factory and Change Method Signature, described in Fowler's

seminal book on refactoring, (ii) a demonstration of generalization-related refactorings such as Extract Interface, which extracts an interface from a class to reduce dependencies on implementation particulars, and Generalize Type, for remediating over-specific declarations, and (iii) a demonstration of Genericize Container Uses, a refactoring in development for migrating uses of standard container classes (e.g. ArrayList or HashMap) to their generic counterparts in Java 1.5 (e.g., ArrayList<String>). This determines the element types that flow into containers, modifies declarations and allocations where possible, and removes casts rendered redundant.

The latter two categories require non-trivial static analysis, and present challenging software engineering usability issues, reflecting complications that programmers face in performing such transformations manually. Therefore, each demonstration will include a brief overview of the technical challenges that the refactoring presents, and highlights of the approach we took in addressing them. In particular, we briefly describe a new piece of Eclipse infrastructure designed to help implement various kinds of context-sensitive and context-insensitive type and pointer analyses for Java.

**Tuesday 10:30-11:15, Courtyard**

**Thursday 12:30-13:15, Courtyard**

## **5 JQuery: Finding Your Way Through Tangled Code**

Edward McCormick, *UBC*  
Kris de Volder, *UBC*

JQuery is a flexible, query-based source code browser developed as an Eclipse plug-in that allows the user to define browsers on-the-fly by formulating logic queries and running them against the source code. A typical IDE based exploration of an OOP system will often involve multiple searches through class hierarchies, field accesses, method calls, regular expression matches and more. Developers who must follow connections between these disconnected views may find great difficulty in combining the capabilities of each view and may as well suffer significant disorientation due to loss of context while switching. JQuery alleviates the disorientation associated with switching tools to complete a single task. To begin browsing the code, the user may either select from a set of predefined browsers or create a browser by entering a query. Because JQuery is built on a rich logic language called TyRuBa, a wide variety of context specific searches can be performed on any node. Users can also edit a configuration file to add their own queries to the JQuery menus.

# Demonstrations

Following a brief presentation of the ideas and motivation behind JQuery, why it is useful, and its relevance OOP, the audience will see a live demonstration of the tool.

The task will involve locating information from scattered places in an Object Oriented system in order to perform a refactorization. This task will demonstrate how JQuery can help maintain search context even within complex, multi-stepped searches, because a JQuery view can retain multiple browsers within one window.

**Tuesday 15:30-16:15, Courtyard**  
**Wednesday 11:30-12:15, Courtyard**

## **6 Modelling Event Driven (RFID) Applications With a Specification Language**

Murali Kaundinya, *Client Services, Sun Microsystems, Inc.*

Syed Ali, *Client Services, Sun Microsystems, Inc.*

Radio Frequency Identification technology provides the means to track any object, any time, anywhere. A major consequence of this technology is that the existing Information Technology systems, applications and processes have to be retrofitted to have RFID-awareness. Many new systems and applications have to be developed while the technology and standards are still emerging. These will be driven by dynamic business processes and therefore have to be agile, and easy to modify and use by a business user.

Model Driven Architecture from the Object Management Group is a laudable vision with lofty goals and a wobbly foundation. We describe its merits and challenges. We describe DASL from Sun Microsystems, Inc., a practical realization of the MDA vision. DASL helps you go directly from an application model to its deployment, bypassing traditional coding work. It provides for a rapid, intuitive modelling of enterprise applications and supports an automated deployment of its implementation to a variety of J2EE architectures. We discuss how we used DASL to model an RFID application, test for its correctness and proceed to deploy auto refreshing web applications and services to track event data about the type of objects, their whereabouts in terms of their location and time. We model the application with Java and UML and use RFID technology to sense the electronic product code tags and integrate them with back end systems.

The audience will see a live demo of how the application is modelled from scratch and deployed as a web application using DASL.

**Tuesday 16:30-17:15, Courtyard**  
**Wednesday 10:30-11:15, Courtyard**

## **7 Design Snippets: Partial Design Representations Extracted from Source Code**

Vibha Sazawal, *University of Washington*

David Notkin, *University of Washington*

Most software undergoes frequent refactoring, modification, and extension. Decisions made as a software system is implemented and modified affect whether the system becomes increasingly brittle (hard to change) over time. In this demonstration, we present the Design Snippets Tool, a new tool that supports Java programmers as they make decisions related to ease of change.

The Design Snippets Tool statically analyzes Java code to generate partial design representations ("design snippets"). Programmers can use design snippets to identify design problems and evaluate planned modifications. Design snippets are co-displayed with Java code for easy reference during refactoring and maintenance tasks.

Written in Java, the Design Snippets Tool is implemented as a plug-in to the Eclipse IDE. The tool generates four snippets from Java code and updates them automatically as the user edits. Each snippet is scoped to remove information irrelevant to the Java file currently in focus. Configuration options allow users to trade completeness for speed.

During the demonstration, we will present the snippets generated by the tool and their purpose. Usage scenarios and implementation details will also be described. If time permits, we may briefly discuss our experiences using the tool in "dog food"-style as the tool was being developed.

The Design Snippets Tool illustrates how static analysis can be used to improve the design of OO software evolving incrementally. Two features make the tool unique: 1) the focus on ease of change, and 2) the quick creation of partial design views that are co-located with code.

**Wednesday 10:30-11:15, Courtyard**  
**Thursday 12:30-13:15, Courtyard**

## **8 Meta-Programming for the Real World**

Matthew Fowler, *New Technology / Enterprise Ltd*

JeeWiz is a commercial tool for model-driven development of enterprise-level systems. It can describe and generate any technology, and has been used for system-level generation (multi-tier J2EE, .NET, SOA

# Demonstrations

systems) and model transformations (XMI to WSDL/XML Schema, and back).

The most difficult problem we have confronted is the complexity of mapping from rich, high-level logical models to the detailed artifacts required in real-life systems. JeeWiz provides reusable architecture and technology layers, local flexibility and cross-tier patterns, which makes JeeWiz meta-programming particularly complex; each major version of JeeWiz has been driven by the need to manage this complexity.

This process has resulted in a number of new elements - our "sub-atomic particles" of model-driven development - to allow us to partition the technology, domain and skill spaces into independently-changeable areas. These are then reassembled using a synergistic combination of meta-models, patterns, a framework for generation, and large-scale build techniques. The assembly process applies classic O-O techniques - inheritance, polymorphism, logical/physical management, encapsulation and patterns - to improve the effectiveness of JeeWiz meta-programming.

The demonstration will show:

- the process of specifying a model based on a meta-model
- the transformations to build a real system
- the process of meta-program maintenance, showing how the new elements are developed in isolation and then combined.

**Tuesday 12:30-13:15, Courtyard**

**Wednesday 10:30-11:15, Courtyard**

## **9 JRA: Offline Analysis of Runtime Behaviour**

Helena Aberg Ostlund, *BEA Systems Inc.*

The JRockit Java virtual machine has a built-in capability to produce recordings of the runtime behaviour of an application environment. It is a light-weight system with a very low performance cost, which can give valuable insights as to how the JVM and application interact in production. The JRockit Runtime Analyzer (JRA) is a tool written in Java for analyzing these recordings by graphic visualization.

The JRA system has many advantages over traditional profiling tools. It doesn't require installation of any new software to enable a recording and its low performance overhead makes it well suited for deployed production systems.

We will present the techniques behind the JRA system and demonstrate how to record a snapshot of a running application. We will view the captured data in the JRockit Runtime Analyzer tool. The audience will see that

detailed and highly useful information about the runtime behaviour of the application and virtual machine can be provided easily at a very low performance cost.

**Tuesday 16:30-17:15, Courtyard**

**Wednesday 12:30-13:15, Courtyard**

## **10 Modeling and Building Software Product Lines with Eclipse**

Olaf Spinczyk, *Friedrich-Alexander-Universität Erlangen*  
Danilo Beuche, *pure-systems GmbH*

Software product line development can reduce the overall software production costs, but imposes extra complexity on the development process. Dealing with the commonalities and variabilities of the product variants and with the flexible software architecture makes product line development a real challenge. Developers can only be successful with adequate tool support for product line development, which is still not the "state of the art".

The demonstration shows the integration of software product line modeling based on feature models into the popular integrated development environment Eclipse. We will demonstrate that a flexible product line architecture can be build by using the modeling capabilities provided by pure::variants, an eclipse plugin for software product line development.

The plugin covers all steps of product line development from requirements and variability analysis to product generation. Extended feature models are used for modeling of problem domain . Family models are used to represent the variable architecture of product line solution domains independent of the programming or modeling languages used for product line implementation.

Along some concrete examples from the domain of embedded software product lines we will make a round-trip through all steps of the development from domain analysis to product deployment and demonstrate how pure::variants facilitates the entire process. We will show how features are used to model problem domains, how product line architectures are connected with domain models and finally how to integrate concepts of model driven development into the process of product derivation.

# Demonstrations

**Tuesday 16:30-17:15, Courtyard**  
**Wednesday at 15:30-16:15, Courtyard**

## **11 Prism is Research in Software Modularization through Aspect Mining**

Charles Chuan Zhang, *University of Toronto*  
Hans-Arno Jacobsen, *University of Toronto*

A crucial premise of applying aspect oriented software development is the identification of crosscutting concerns or aspects. For very large software systems, finding aspects is often a great challenge due to their non-localized presence in code. In this software demonstration, we show Prism, a source code analysis tool aiming at giving users lucid views of non-modularized concerns in large code bases.

Prism implements a user-driven aspect mining approach by providing diversified ways of describing hidden concerns. These descriptions, called "fingerprints," can be simple lexical patterns or more complicated type patterns. For complex characteristics of aspects, Prism has the ability of specifying arbitrary code phrases which describe syntactical structures, type information and program flows about aspects. In addition, different types of descriptions can be composed together to form more complex fingerprints.

Implemented as an Eclipse plugin, Prism provides an integrated mining environment with the Eclipse IDE in which the user can easily create, modify, and reuse fingerprints. Prism computes matches of the fingerprints and allows direct navigation from these matches, called "footprints," to their actual location in the code. Prism is also designed as a cross-language aspect mining platform currently with full support for Java systems and partial support for C# systems.

This demo will first illustrate the afore-described functionalities of Prism. A case study is then presented to show an aspect mining methodology we have developed and the effectiveness of finding and analyzing aspects in large software systems.

Prism is readily available at Prism Eclipse plug-in for Download at <http://www.eecg.toronto.edu/~jacobsen/prism/> . Prism has been actively used in several research projects

**Tuesday 11:30-12:15, Courtyard**  
**Thursday 10:30-11:15, Courtyard**

## **12 Smell Detection for Eclipse**

Robert Fuhrer, *IBM T.J. Watson Research Center*  
Arnab Bhattacharyya, *MIT*

We demonstrate a new smell detection framework for Java, implemented for the Eclipse IDE ([www.eclipse.org](http://www.eclipse.org)). Smells are architectural, rather than functional, flaws in software that tend to reduce maintainability, extensibility, modularity, testability, or other software quality measures. Common smells include overly-long method bodies, message chains, parallel inheritance, use of switch statements to implement polymorphic behavior, data clumps, overly specific variable types, and duplicated code. Code duplication in particular is a well-known source of maintenance problems. Because smells do not represent functional flaws, they can be remediated by behavior-preserving transformations (refactorings).

We have implemented an extensible smell detection framework in Eclipse that provides a simple Eclipse extension point for defining smell detectors, along with several basic smell detectors, including an efficient code duplication detector. Several of our smell detectors feature Eclipse quick-fix refactorings to automatically remediate smells upon the user's request. This transforms the tool from a mere problem indicator into a tool that actively assists developers in improving their code. We believe that such a facility has significant potential in educational settings, in directing a student's attention to software engineering principles while coding.

Our demonstration consists of two parts. First, we'll demonstrate some of our smell detectors on realistic source code bases (including Eclipse itself), along with their corresponding quick-fix support. Second, we'll walk through the implementation of a very simple smell detector within our framework, along with a very simple remediation.

**Tuesday 10:30-11:15, Courtyard**  
**Wednesday 11:30-12:15, Courtyard**

## **13 AJEER: An AspectJ-enabled Eclipse Runtime**

Martin Lippert, *it-wps GmbH & University of Hamburg*

In this demonstration we will present the AspectJ-enabled Eclipse runtime (AJEER for short in the following). This enhanced runtime for Eclipse allows developers to use two modularization technologies together seamlessly: OSGi-based plugins using Eclipse and aspects using

# Demonstrations

AspectJ. This enables developers to modularize concerns that crosscut plugin boundaries without the need to recompile all plugins of the system. Aspects can be promoted to the system by just installing them via the standard Eclipse plugin mechanism. This makes it possible to build Eclipse RCP (rich client platform) applications while using AspectJ at the same time – combining all the features of both technologies without limitations.

The presented version of AJEER works with the new OSGi-based runtime of Eclipse 3.0. Because the latter allows you to add and remove plugins at runtime AJEER can offer this feature for aspect-promoting plugins as well.

The demonstration will provide an in-depth look at the implementation of the load-time weaving runtime and will show how the Eclipse runtime can be extended. To illustrate the possibilities of AJEER the demonstration will include a number of example applications ranging from a re-implemented version of Chris Laffras Eclipse Monitor (showed last year at OOPSLA in the static bytecode modification version) to an IBM research project that takes heavy advantage of the AspectJ-enabled runtime. Attendees will also see how the Eclipse PDE and the AspectJ extension AJDT can be used in combination to build aspect-promoting plugins. The implementation of AJEER is open source under the CPL 1.0 and can be downloaded at <http://www.martinlippert.com/>

**Wednesday 15:30-16:30, Courtyard**  
**Thursday 11:30-12:15, Courtyard**

## **14 Program Transformations for Re-Engineering C++ Components [OOPSLA/GPCE]**

Ira Baxter, *Semantic Designs*

Larry Akers, *Semantic Designs*

Michael Mehlich, *Semantic Designs*

Component-based software engineering enables applications to be assembled from component parts, provided they adhere to a component-style specific interface specification and protocol. Obviously, components available for one style are not available for another. Component styles evolve, too, which can obsolete components using a legacy style. This creates a demand for migrating components from one style to another, which can require complex changes to the component source code. For a large component library, doing this manually is likely prohibitive. An alternative is to apply automated program transformations to carry out the changes.

Using source-to-source transformations on real code requires a scalable, robust program transformation technology. Such technologies are difficult to justify for

single applications. DMS is a commercial program transformation system which has been used to transform many programming languages, including C++, C#, Java and ObjectPascal. It is parameterized by language and desired task, enabling its infrastructure costs to be amortized across many different software analysis or change applications.

This demonstration shows a concrete example of DMS program transformations being used to migrate legacy C++ components from a Boeing distributed avionics software system, using a Boeing proprietary component format, to a CORBA component style. The conversion requires nontrivial understanding and manipulation of the C++ source code. It will explain the component migration problem to be solved, show some of the transformations, and actually convert a component.

**Tuesday, 11:30-12:15, Courtyard**  
**Thursday at 12:30-13:15, Courtyard**

## **15 C-SAW and GenAWeave: A Two-Level Aspect Weaving Toolsuite [OOPSLA/GPCE]**

Jeff Gray, *University of Alabama at Birmingham*

Ira Baxter, *Semantic Designs*

Jing Zhang, *University of Alabama at Birmingham*

Suman Roychoudhury, *University of Alabama at Birmingham*

The C-SAW and GenAWeave tools support evolution of object-oriented 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 generated transformation rules, along with the initial version of the application source code, serve as input to the Design Maintenance System (DMS) from Semantic Designs. The generated rules drive the transformation process in order to produce a modified version of the source containing new concerns that have been woven across the application code base. 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

# Demonstrations

demonstration will illustrate the use of DMS 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.

**Wednesday 15:30-16:15, Courtyard**  
**Thursday 10:30-11:15, Courtyard**

## **16 The Concern Manipulation Environment [OOPSLA/GPCE]**

Peri Tarr, *IBM Thomas J. Watson Research Center*  
Matthew Chapman, *IBM Hursley Park*  
William Chung, *IBM Thomas J. Watson Research Center*  
Andy Clement, *IBM Hursley Park*

The Concern Manipulation Environment (CME) aims to provide a set of open, extensible components and a set of tools that promote aspect-oriented software development (AOSD) throughout the software lifecycle. It has two main goals:

1. To provide an open, integrated development environment (IDE) to enable software engineers to use AOSD techniques throughout the software lifecycle, and to allow them to use different AOSD approaches in an integrated manner.
2. To promote the rapid development of new tools supporting AOSD at any stage of the software lifecycle, and to serve as an integrating platform for such tools, facilitating development and experimentation with new AOSD approaches.

This demonstration will highlight a number of tools and components that are useful to software developers and to AOSD tool providers and researchers. Tools for software developers include ones that allow developers to identify, model and visualize concerns, aspects and relationships in their software, covering software artifacts of any type, including both code and non-code artifacts, and including latent concerns or aspects that were not separated in the artifacts; that enable flexible queries over software; and that compose/integrate aspects and other concerns.

For AOSD tool providers and researchers, the demonstration will describe some of the CME's support for integration of tools and approaches within the environment, highlighting the integration of Java, AspectJ and Ant artifacts within the CME, and how to use the CME's extensible components to create new AOSD tools or prototypes more rapidly.

**Tuesday 11:30-12:15, Courtyard**  
**Thursday 10:30-11:15, Courtyard**

## **17 Performance Explorer: Understanding Java Application Behavior**

Matthias Hauswirth, *Univeristy of Colorado at Boulder*  
Peter Sweeney, *IBM T.J. Watson Research Center*  
Amer Diwan, *Univeristy of Colorado at Boulder*

This demonstration will illustrate how the visualization tool, Performance Explorer (PE), can be used to understand the behavior of Java applications.

PE allows the visualization of data over time in various graphical and tabular ways. In addition, PE allows the interactive creation and modification of visualizations. Most properties of the visualization are defined using expressions in a language similar to Java expressions.

PE is particularly well suited for traces generated by Jikes RVM, an open source research virtual machine for Java.

**Tuesday 15:30-16:15, Courtyard**  
**Thursday 11:30-12:15, Courtyard**

## **18 ConstrainedJava**

Donald Gordon, *Victoria University of Wellington*  
James Noble, *Victoria University of Wellington*  
Robert Biddle, *Carleton University*

Encapsulation is a fundamental principle of object-oriented programming. Support for encapsulation in many languages is limited to specifying access restrictions for fields; this does not guard against unrestricted aliasing, where due to programmer error, references to internal parts of aggregate objects can leak into other parts of the program, breaking encapsulation. Several systems have been proposed to deal with this problem. ConstrainedJava is a programming language with a syntax and semantics similar to those of Java (based on the BeanShell Java source interpreter), extended to maintain runtime ownership relationships between individual objects. Constraints on message sends between objects can then be imposed based on these ownership relationships. This enforces the encapsulation topology intended by the programmer, ensuring that references which leak from aggregate objects cannot cause unforeseen dependencies, as message sends to these leaked references are disallowed. It also allows the programmer to explicitly define dependencies between objects, by providing notification of changes to objects and their children in the ownership tree.

The demonstration will include examples of message constraints, and the types of behaviour they inhibit. Also

# Demonstrations

shown will be a demonstration of a GUI which utilises the explicit dependencies to simplify relating object state to widget state, and a demonstration of the natural cloning operation enabled by utilising ownership information to decide whether references or objects themselves should be copied.

**Tuesday 12:30-13:15, Courtyard**  
**Wednesday 12:30-13:15, Courtyard**

## **19 AspectJ Tools - New and Noteworthy**

Mik Kersten, *University of British Columbia*  
Adrian Colyer, *IBM*

AspectJ is an aspect-oriented programming (AOP) extension to Java that can be used to modularize crosscutting concerns such as error-handling and security policies. AspectJ controls code-tangling and makes the underlying concerns more apparent, making programs easier to develop and maintain. Using only OOP techniques, code for these concerns is scattered and tangled throughout entire programs. While modern Java IDEs provide features for working with encapsulation and inheritance, the AspectJ tools make it easy to work with this crosscutting modularity.

The updated AspectJ development tools suite and Eclipse AJDT plug-in will be used to demonstrate new tool features including incremental building, bytecode weaving, crosscutting structure views, code assist, and debugger support. Some common crosscutting concerns of an existing object-oriented system will be mined and refactored into a modular aspect-oriented implementation. The tools will be demonstrated on an enterprise system, and new features for working with crosscutting in large systems will be shown. The integration of AspectJ into existing development processes will also be reviewed along with JBuilder, NetBeans, Emacs, and command-line tool support. Installation of the tools and project setup will demonstrate how easily AspectJ can be applied to existing Java systems. In addition, a preview of the next-generation AJDT plugin will show how advanced IDE features such as refactoring support will work in upcoming releases of AJDT.

**Tuesday 11:30-12:15, Courtyard**  
**Wednesday 11:30-12:15, Courtyard**

## **20 Molhado: Object-Oriented Structural Software Configuration Management**

Tien Nguyen, *University of Wisconsin - Milwaukee*

Many software configuration management (SCM) systems treat a software system as a set of files on a file

system. However, software engineers usually think and reason in terms of high-level abstractions, compositions, and the interrelations among them. Therefore, SCM systems whose concepts are closely tied to physical structure can become burdensome for developers because software development methods and SCM infrastructures require different mental models.

To bridge that gap, this demonstration presents an object-oriented approach to managing the evolution of software systems at the logical level. Keys to Molhado approach are its object-oriented structural system model and structure versioning framework in which logical objects and structures in a system are extended from a small set of system model concepts, allowing them to be uniformly versioned in a fine-grained manner and independent of the physical file structure. Changes to all objects and structures are captured and related to each other in a cohesive manner via the Molhado product versioning SCM system.

This research demonstration also presents applications of our object-oriented versioning approach for logical objects in different development frameworks such as UML-based object-oriented software development, architecture-based software development, and Web application development. A set of comparison tools was developed to show differences between two arbitrary versions of a system's hierarchical structure, of any logical object, and of any logical unit of an object in both structural and line-oriented fashions. Logical relationships among software objects are maintained and versioned separately from their contents.

**Wednesday 10:30-11:15, Courtyard**  
**Thursday at 10:30-11:15, Courtyard**

## **21 Xirc: Cross-Artifact Information Retrieval [GPCE]**

Michael Eichberg, *Darmstadt University of Technology*  
Thorsten Schaefer, *Darmstadt University of Technology*

In large scale software development projects, in particular in the field of Component-Based Software Engineering (CBSE), the kinds of a project's sources are diverse and related information is spread over the different artifacts. E.g., the transaction attributes (Required, Requires-New, etc.) of methods of an Enterprise Java Bean are defined in the deployment descriptor while the method bodies are defined in a Java class.

If we want to put these information into relation, e.g., to find all methods with a specific transaction attribute, we have to use multiple search engines and have to map the information manually. It is not possible to execute one query that returns the desired result.

# Demonstrations

To solve these problems we have developed Xirc, a tool and architecture that enables one to define queries over a uniform representation of all artifacts of a software project. Xirc maps all artifacts of a project to XML representations and stores the documents in a database. Then, XQuery, a functional query language for XML documents (databases), can be used to query the database. Hence, Xirc can be used as a sophisticated search engine, as a tool to check implementation restrictions, to find errors or as a basis for further tools for code generation and visualization.

The first part of the demo will be a short PowerPoint presentation introducing the Xirc architecture and its features. The second part will be a live demo of the Xirc Eclipse Plug-in. This part will show how to put information spread over (EJB-) deployment descriptors and code into relation to ease the development of a component, to check implementation restrictions or to find errors.

**Tuesday 10:30-11:15, Courtyard**  
**Thursday 11:30-12:15, Courtyard**

## **22 Towards Domain-Driven Development: the SmartTools Software Factory [GPCE]**

Didier Parigot, *INRIA Sophia-Antipolis*

Nowadays, software needs to be more open, flexible, and capable of evolving quickly to meet new user or technology requirements. It should be easy to adapt, even by none computer-specialists. To tackle these challenges for DSL (Domain-Specific language) tools, we have developed a software factory, named SmartTools.

One of the main ideas behind the design of SmartTools (and consequently behind the design of the generated DSL tools) is to model the business logic of each concern in a technology-free manner which can then be used to generate platform-specific code. The following four concerns have been taken into consideration:

- The language data definition;
- The semantics analysis frameworks that describe method signatures and traversal;
- The graphical representation of the data;
- And the component that links together all the tools of a DSL.

To offer open and adaptable tools, its design benefits from the following paradigms:

- Generative programming;
- Aspect-Oriented Software Development (AOSD);
- Component programming;

- Model-Driven Architecture (MDA);
- Standard technologies (XML) and patterns (such as the visitor design pattern).

SmartTools is heavily bootstrapped; that is, it internally uses its technology to develop its own models. Through the development of these models, our approach in integrating the mentioned paradigms and technologies has been intensively tested and refined. Since then, SmartTools has been used to produce tools for many diverse languages such as SVG, DTD, XML scheme, CSS, WSDL, and BPEL.

The SmartTools framework represents approximately 100 000 lines of Java source code before the generation stage and 1 000 000 lines after. This ratio shows the efficiency of this approach and validates this new development approach based on generative programming.

Before the demonstration, we will present the key features of SmartTools. Then during the demonstration, we will focus on how to define different tools for a DSL and show, for each concern, the benefits gained from using the mentioned paradigms and technologies. This demonstration is not only targeted at DSL tool developers, but also at people who want to understand how these new paradigms can be practically integrated into software.

**Tuesday 15:30-16:15, Courtyard**  
**Thursday 12:30-13:15, Courtyard**

## **23 Implementation of DSLs Using Staged Interpreters in MetaOCaml [GPCE]**

Kedar Swadi, *Rice University*

Domain-specific languages (DSLs) allow programmers to write applications faster than in general-purpose languages. Two approaches are commonly taken to implement DSLs. The first is to compile the DSLs into machine-level code, or low-level languages such as C. While this results in fast implementations, writing compilers requires expertise in compiler technology, and long implementation times. An alternative approach uses interpreters to implement DSLs. Though easier to write, extend and maintain than compilers, the overhead of interpretation is unacceptably large for many applications. Relevance to GPCE and uniqueness of approach: The shortcomings in either approaches significantly impede the widespread use of DSL-based generative programming in software development. The demonstration addresses this problem using multistage programming. It shows how staged interpreters for implementing DSLs avoid problems in both abovementioned approaches, and allow efficient machine-level realizations of DSL programs. Unlike other existing approaches, this approach guarantees that the generated

# Demonstrations

code is typesafe. Underlying implementation techniques and technologies used: This demonstration uses multistage programming in the MetaOCaml language to write two-stage interpreters which translate DSL programs into OCaml programs, which are finally compiled into efficient machine code and executed.

First a quick introduction to multistage programming is given using a simple example of the power function that computes  $x^n$ . The demonstration shows how three staging constructs in MetaOCaml are used to automatically generate typesafe specialised versions of this function from the generic function.

Second, the demonstration shows how to implement a staged interpreter for a small language and how to correctly measure performance of implementations. Finally, the demonstration visually shows performance gains from a staged interpreter for the LOGO graphics language. This is done by concurrently running the graphical outputs for the simple and staged interpreters on two windows placed side-by-side.

Research papers and information about downloading, installing, and using MetaOCaml are found at <http://www.metaocaml.org>.

**Tuesday 16:30-17:15, Courtyard**

**Wednesday 12:30-13:15, Courtyard**

## **24 MetaEdit+: Domain-Specific Modeling for Full Code Generation Demonstrated [GPCE]**

Juha-Pekka Tolvanen, *MetaCase*

Domain-Specific Modeling (DSM) raises the level of abstraction beyond programming by specifying the solution directly using domain concepts. In many cases, the final products can be generated from these high-level specifications. This automation is possible because both the language and generators need fit the requirements of only one company and domain.

This demonstration introduces DSM by showing real world cases from various fields of software development:

- enterprise application development into Symbian smartphones
- financial product definition into B2B J2EE web site
- voice menu development into 8-bit microcontroller
- MMS/SMS telecom service configuration

These cases illustrate how DSM, giving first class support for modeling, can prevent incorrect or unwanted designs at early stages of the development, how underlying

platform complexity is hidden, and how full code can be generated from the modeler's point of view.

The second part of the demonstration will show in an interactive manner both the design side and the use side of DSM languages and generators. On the DSM use side, we implement the OOPSLA/GPCE conference registration application into a mobile phone. This is done by modeling in MetaEdit+ tool. Then we shift to the DSM design side: Using MetaEdit+ tool for metamodeling, we extend the modeling language as well as the generator. Language extensions deal with adding domain constraints, rules and new concepts. Once the DSM is extended, this allows us to revert to modeling in order to finalize our sample conference registration application.



# Special Events

*a friend's airplane mechanic who swears blind that  
Disinterested Requester works really, really well.*

**Wednesday, 19:00-23:00, (See Info Booth for  
Transportation Information)**

## ***Special Event at the Vancouver Aquarium***

The Vancouver Aquarium is home to over 60,000 colorful aquatic creatures, exciting exhibitry and educational displays. Voted one of Canada's best-loved attractions, the Aquarium is an intriguing and unforgettable venue. The evening will include a light reception and entertainment. One ticket to this event is included in the full conference registration, and additional tickets are \$70.00 each.

**Thursday, 15:00-17:00, Convention Level  
Lobby**

## ***Ice Cream Social/OOPSLA 2005 Invitation***

The Ice Cream Social serves as the kick-off event of next year's OOPSLA where you will be able to pick up your OOPSLA 2005 poster and share your 2004 experiences.

# FlashBoF

**Chair: Rob van den Berg, Siemens VDO Automotive**



Birds-of-a-Feathers, or BoFs have been a long standing tradition at OOPSLA. Many new products and ideas have been born, nurtured and raised at BoFs (e.g. eclipse, Squeak). Given their success, we are going to raise the status of BoFs this year! Meet the FlashBoF.

FlashBoF is the marriage between a BoF session and an 'open space'. So what is an Open Space? An Open Space meeting is like a formalized coffee break: people getting together to discuss a particular subject. But maybe the following quote from <http://www.openspaceworld.org/wiki/wiki/wiki.cgi?>

describes it better.

“In Open Space meetings, events and organizations, participants create and manage their own agenda of parallel working sessions around a central theme of strategic importance, such as: What is the strategy, group, organization or community that all stake-holders can support and work together to create?”

The basic idea with an Open Space is that participants manage their own agenda. Someone who wants to discuss a topic announces it at the start of an Open Space session, and invites others to participate. As an organizer of an Open Space session, he or she then has the responsibility to write minutes or a summary of what was discussed. Another vital point for Open Space meeting is the participation: it is not considered rude to leave a session. Only people interested in the subject should be present, and if the subject is not interesting, leave!

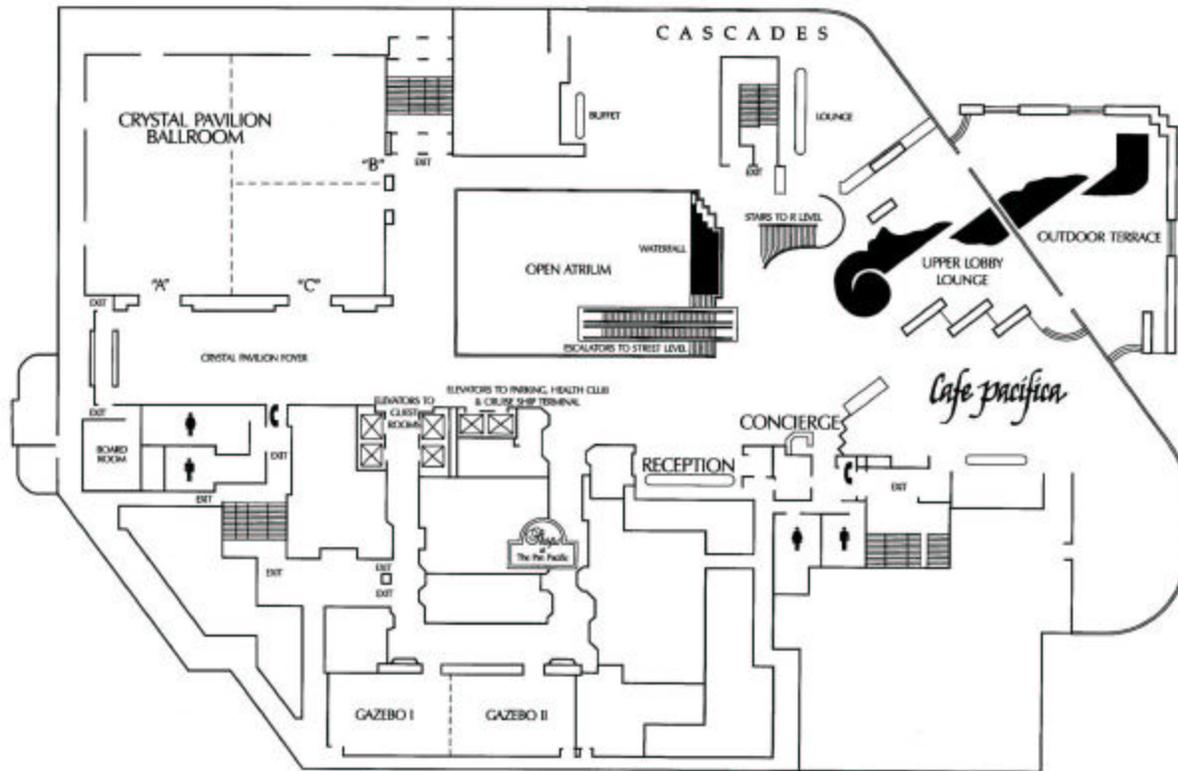
So what does this mean for the FlashBoF? Instead of using written announcements, Tuesday and Wednesday evenings of the conference FlashBoFs will be held. Each FlashBoF session will last about 90 minutes. People are invited to announce their FlashBoFs with time and place at the conference. A conference Wiki is made available for the FlashBoF organizers to display their FlashBoF results.

**Tuesday, 17:00-19:00,  
Meeting Rooms 7-16**

**Wednesday, 17:00-19:00,  
Meeting Rooms 7-16**

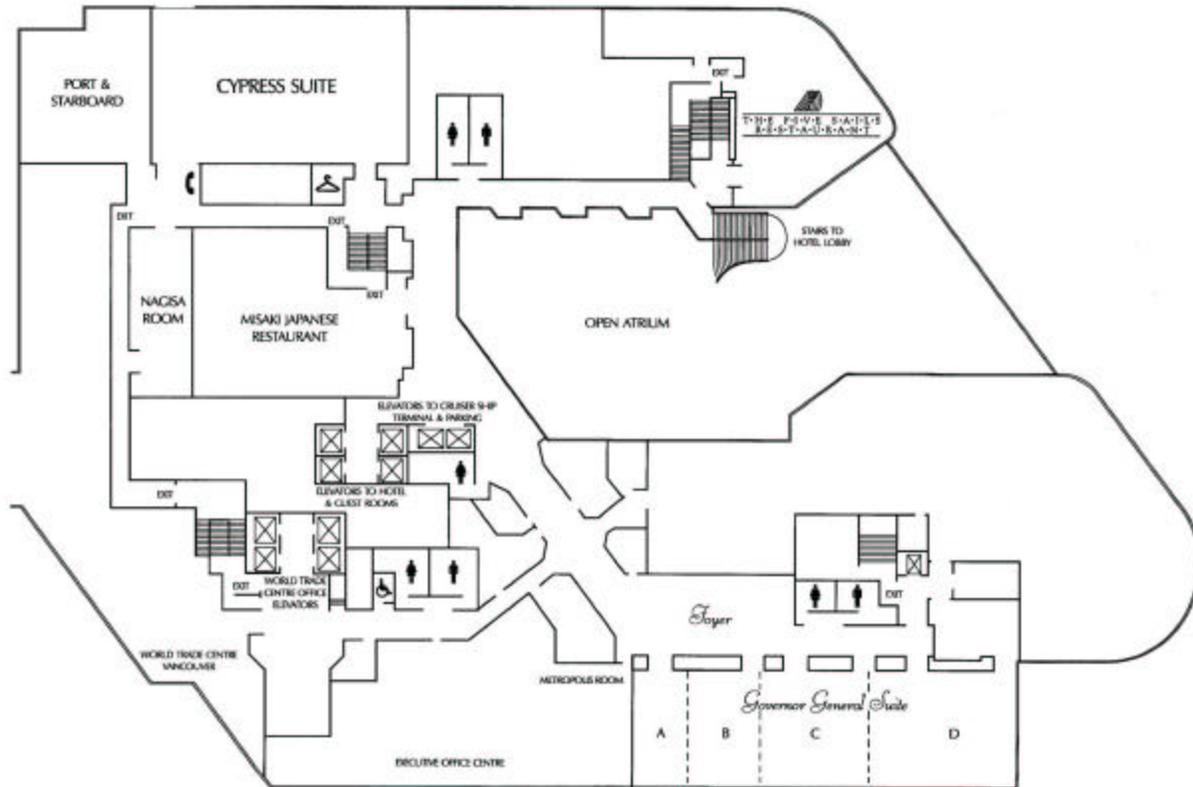
# Maps

## Pan Pacific Hotel Lobby-Level



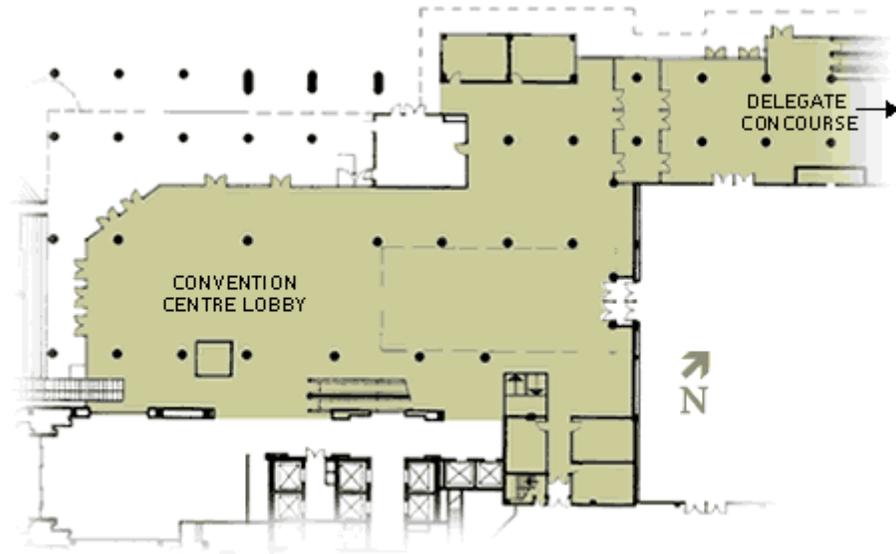
# Maps

## Pan Pacific Hotel Restaurant-Level

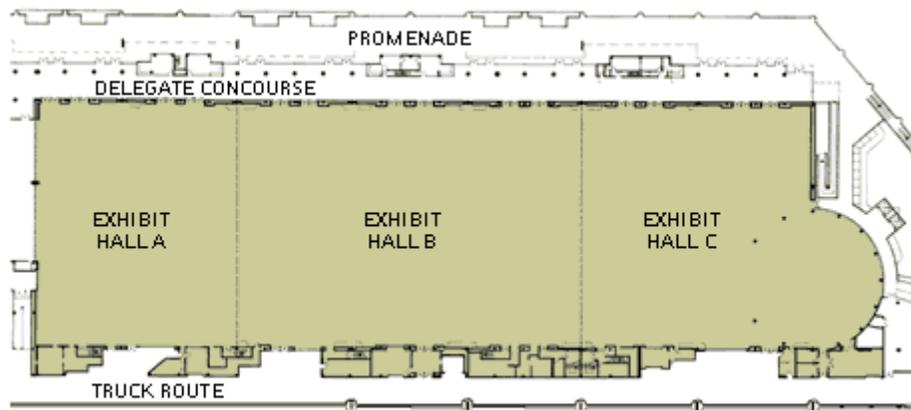


# Maps

## Vancouver Convention & Exhibition Centre Lobby and Concourse



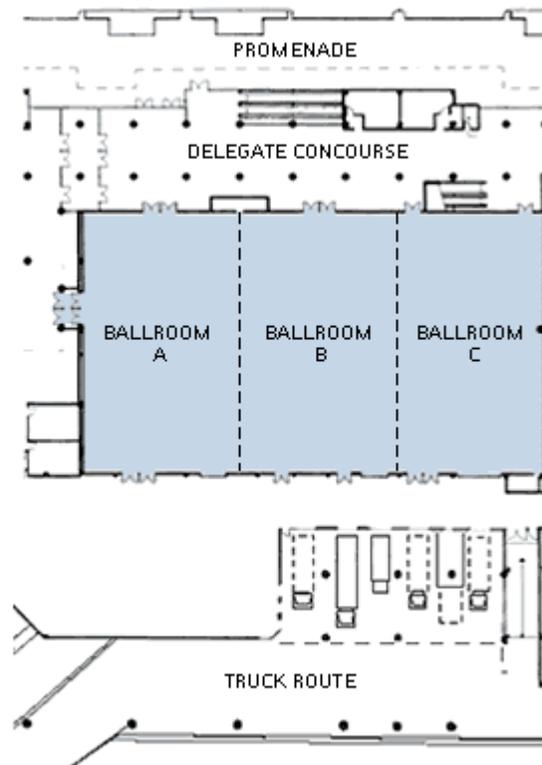
## Vancouver Convention & Exhibition Centre Exhibition Hall



The Courtyard Is Located In Exhibition Hall A

# Maps

## Vancouver Convention & Exhibition Centre Ballrooms



## Vancouver Convention & Exhibition Centre Meeting Rooms



## Transportation

### **Airport:**

Vancouver International Airport, located about 30 minutes from the hotels (<http://www.yvr.ca>)

### **Ground Transportation:**

- Public Transportation - taxis, limos, shuttles, car rentals, and driving directions to everything you need to know to get to and from Vancouver, the Lower Mainland, Victoria, and Whistler (<http://www.yvr.ca/guide/toandfrom/index.asp>)
- Via Rail - transcontinental (<http://www.viarail.ca>)
- Amtrak - to/from Seattle (<http://www.wsdot.wa.gov/amtrak>)
- Greyhound Bus line (<http://www.greyhound.ca>)
- Budget Rent-a-Car, 800-527-0700
- Avis Rent-a-Car, 800-230-4898
- Alamo Rent-a-Car, 800-462-5266
- National Car Rentals, 1-800-CAR-RENT
- Hertz Canada, 800-654-3131
- Thrifty Car Rental, 800-847-4389

## Things to do in Vancouver

Vancouver is the province of British Columbia's largest city and Canada's third largest city. It's bordered by water on three sides and is nestled alongside the Coast Mountain Range. Vancouver has one of the mildest climates in Canada. The Pacific hotel is located on Vancouver's downtown harbour front and shares the same complex as Vancouver's World Trade Centre and the Vancouver Convention & Exhibition Centre. Take a stroll through Vancouver's beautiful Stanley Park, the largest city park in Canada. Walk across the beautiful Lions Gate Bridge and stop midway for a bird's-eye view of Burrard Inlet and the North Shore mountains. Take a ride across Burrard Inlet on the SeaBus, and view the cargo ships up close. Queen Elizabeth Park is Canada's first civic arboretum- and botanical garden devoted to trees. Check out the world's thinnest office building on the corner of Pender and Carrall streets in Chinatown. The city boasts over 14,000 sidewalk wheelchair ramps and extensive public transit (bus, Skytrain, SeaBus) designed to accommodate persons with disabilities. Visit Gastown and hear the Gastown Steam Clock whistle every hour on the hour. It is the only one of its kind in the world. Granville Island is the place to spend a leisurely afternoon. Part farmers market, part artist studio, and part magnet for the creative and flamboyant - it's one of the busiest spots in Vancouver. Visit Whistler, a ski lover's paradise, just a 2.5-hour drive from Vancouver. And don't forget to go to the Vancouver Aquarium (an OOPSLA event location). Visit <http://www.tourismvancouver.com/docs/visit/attractions.html> for a complete listing of attractions, restaurants, entertainment and more.

# ACM / SIGPLAN / SIGSOFT



ACM, the Association for Computing Machinery, founded in 1947, is the world's first educational and scientific computing society. Today, its members – over 75,000 computing professionals and students worldwide – and the public turn to ACM for authoritative publications, pioneering conferences, and visionary leadership for the new millennium. ACM offers its members a vast array of IT information resources, including the ACM Portal, which includes over 750,000 pages of text, the Online Guide to Computing Literature (with 350,000 bibliographic citations extending far beyond ACM's own literature), the Online computing Review Service, customized personal services and more.

For a complete listing of ACM's Professional and Student memberships benefits and options, visit <http://www.acm.org/joinacm>.

**SIGPLAN** ACM SIGPLAN explores programming languages concepts and tools, focusing on design, implementation, and efficient use. Its members are programming language users, developers, implementers, theoreticians, researchers and educators. The monthly newsletter "ACM SIGPLAN Notices" publishes several conference proceedings issues, regular columns and technical correspondence. This SIG offers an additional newsletter (FORTRAN forum), on a subscription only basis. SIGPLAN sponsors four major annual conferences: the OOPSLA conference on object-oriented technology, the Conference on Programming Language Design and Implementation (PLDI), the major professional conference in the field, the Symposium on Principles of Programming Languages (POPL), and the International Conference on Functional Programming (POPL), held periodically in Europe.

## **Contact ACM to Join**

### **Voice:**

1-800-342-6626: US & Canada

1-212-626-0500 NY & Global

### **Fax:**

1-212-944-1318

### **Email:**

[acmhelp@acm.org](mailto:acmhelp@acm.org)

### **Web:**

<http://www.acm.org/>

<http://www.acm.org/join>

<http://www.acm.org/sigplan>

<http://www.acm.org/sigsoft>

# OOPSLA 2004 Committees

## Conference Chair

John Vlissides, *IBM Research*

## Program Chair

Doug Schmidt, *Vanderbilt*

## Communications Chair

Dwight Deugo, *Espirity, Inc.*

## Corporate Support Chair

Ralph Johnson, *University of Illinois*

## Demonstrations & Exhibits Chair

Neil Harrison, *Avaya Labs*

## DesignFest<sup>®</sup> Chair

Rob van den Berg, *Siemens VDO Automotive*

## Doctoral Symposium Chair

Laurie Williams, *NC State University*

## Educators' Symposium Chair

Eugene Wallingford, *University of Northern Iowa*

## Internet Technology Chair

Ken Bauer, *ITESM Campus Guadalajara*

## Onward! Chair

Geoff Cohen, *Coherence Engine*

## Panels Chair

Kyle Brown, *IBM*

## Posters Chair

Torsten Layda, *SWX Swiss Exchange*

## Practitioner Reports Chair

Gail E. Harris, *Instantiated Software*

## ACM Student Research Competition Chair

Torsten Layda, *SWX Swiss Exchange*

## Student Volunteer Chair

Don Roberts, *University of Evansville*

## Steering Committee/Trivia Chair

Linda Northrop, *SEI*

## Technology Chair

Max Rahder, *Rahder Consulting*

## Treasurer

Shail Arora, *Gradepoint, Inc.*

## Tutorials Chair

Steve Metsker, *CapTech Ventures*

## Workshop Chair

Helen Sharp, *The Open University, UK*

## OOPSLA 2005 Chair

Ralph Johnson, *University of Illinois*

## Liaison with ACM

Donna Baglio, *ACM*

## Operations

Nancy Wilson, *Meeting Strategies Worldwide*

## Registration

Carole Mann, *Registration Systems Lab*

## CyberChair Operation and Support

Richard van de Stadt, *Borbala Online Conference Services*

## Webmaster

Ken Bauer, *ITESM Campus Guadalajara*

## Program Committee

Gul Agha, *University of Illinois, Urbana-Champaign*

David Bacon, *IBM T.J. Watson Research*

Elisa Baniassad, *Trinity College*

Frank Buschmann, *Siemens*

Roy Campbell, *University of Illinois, Urbana-Champaign*

Fabio Kon, *University of São Paulo*

Doug Lea, *SUNY Oswego*

Jens Coldewey, *Coldewey Consulting*

Krzysztof Czarnecki, *University of Waterloo*

Richard P. Gabriel, *Sun Microsystems*

Gail E. Harris, *Instantiated Software*

John Hatcliff, *Kansas State University*

David Holmes, *DLTECH*

Arno Jacobsen, *University of Toronto*

Torsten Layda, *SWX Swiss Exchange*

Gary Leavens, *Iowa State University*

Joe Loyall, *BBN Technologies*

Satoshi Matsuoka, *Tokyo Institute of Technology*

John McGregor, *Clemson University*

Mira Mezini, *University of Siegen*

Martin Odersky, *EPFL*

Jens Palsberg, *UCLA*

Guy Steele, *Sun Microsystems*

Peri Tarr, *IBM T.J. Watson Research*

Steve Vinoski, *IONA Technologies*

Jan Vitek, *Purdue University*

Dave Ungar, *Sun Microsystems*

## ACM Student Research Competition Committee

Aditya Agrawal, *Vanderbilt University*

Marat Boshernitsan, *University of California, Berkeley*

Ada Diaconescu, *Dublin City University*

Torsten Layda, *SWX Swiss Exchange*

## Corporate Support Committee

Richard P. Gabriel, *Sun Microsystems*

Neil Harrison, *Avaya Labs*

Mamdouh Ibrahim, *IBM Global Services*

John Vlissides, *IBM Research*

## Demonstrations Committee

Martine M. Devos, *Avaya Labs Research*

# OOPSLA 2004 Committees

Neil Harrison, *Avaya Labs*  
Simon Helsen, *Interactive Objects GmbH*

## DesignFest<sup>®</sup> Committee

Rob van den Berg, *Siemens VDO Automotive*  
Matthew S. Davis, *The Pennsylvania State University*  
Raul Gnaga, *SES-ASTRA*  
Jim Heliotis, *Rochester Institute of Technology*  
Daan Hoogland, *Snow B.V.*  
Torsten Layda, *SWX Swiss Exchange*  
Peter Pascale, *Pearson VUE*  
John Rayburn, *Threewide Corporation*  
Wafa Aldiwan, *California Institute of Technology*

## Doctoral Symposium Committee

Laurie Williams, *North Carolina State University*  
Bjorn Freeman-Benson, *University of Washington*  
Jim Coplien, *Vrije Universiteit Brussel and University of Manchester*  
Philippe Kruchten, *University of British Columbia*  
James Noble, *Victoria University of Wellington*  
Mario Wolczko, *Sun Microsystems*

## Educators' Symposium Committee

Owen Astrachan, *Duke University*  
Joe Bergin, *Pace University*  
Robert Biddle, *Carleton University*  
Robert Duvall, *Duke University*  
Mary Lynn Manns, *UNC Asheville*  
Rick Mercer, *University of Arizona*  
Linda Rising, *Independent Consultant*  
Helen Sharp, *The Open University, UK*  
Eugene Wallingford, *University of Northern Iowa*

## Onward! Program Committee

Geoff A. Cohen, *Coherence Engine*  
William Cook, *University of Texas, Austin*  
Richard P. Gabriel, *Sun Microsystems*  
Cristina Lopes, *University of California, Irvine*  
Martin Rinard, *Massachusetts Institute of Technology*

## Poster Committee

Rob van den Berg, *Siemens VDO Automotive*  
Uri Dekel, *Carnegie Mellon University*  
Ada Diaconescu, *Dublin City University*  
Elena Gonzalez-Volynskaya, *Univision Online*  
Daan Hoogland, *Snow B.V.*  
Torsten Layda, *SWX Swiss Exchange*  
Martin Lippert, *it-wps GmbH*  
Wolfgang Radinger, *SPIRIT Business & Finance Solutions*

## Practitioner Reports Committee

Gail E. Harris, *Instantiated Software*  
Lougie Anderson, *Sabrix, Inc.*  
Rebecca Wirfs-Brock, *Wirfs-Brock Associates*  
Linda Rising, *Independent Consultant*  
Steven Fraser, *Independent Consultant*

## Tutorials Support Committee

Steve Metsker, *CapTech Ventures*  
Doug Lea, *SUNY Oswego*  
Kevlin Henney, *Independent Consultant*  
Peri Tarr, *IBM T.J. Watson Research*  
Rebecca Wirfs-Brock, *Wirfs-Brock Associates*

## Workshop Committee

Ali Arsanjani, *IBM Global Services*  
Martine M. Devos, *Avaya Labs Research*  
Lars Grunske, *Hasso-Plattner-Institute at the University of Potsdam, Germany*  
Laura Hill, *Sun Microsystems Laboratories*  
Torsten Layda, *SWX Swiss Exchange*  
John S. Nolan, *Sun Microsystems Laboratories, Europe*

# OOPSLA 2005



# OOPSLA 2005

**Conference Chair: Ralph Johnson, *University of Illinois***

On behalf of the entire OOPSLA 2005 Conference Committee, we invite you to contribute and actively participate the 20th OOPSLA, to be held in San Diego, California, October 16-20, 2005, at the Town and Country Resort & Convention Center. As participants in previous OOPSLAs will understand, OOPSLA is widely recognized as the premier gathering of practitioners, researchers, educators, and students sharing their thoughts and experiences with object technology.

OOPSLA offers many venues for participation, including technical papers, practitioner reports, expert panels, intriguing invited talks, demonstrations, exhibits, posters, participant-led workshops, conferences-within-conference (one specifically designed for educators and a second for doctoral students), informal “birds-of-a-feather” sessions, and plenty of social opportunities for networking (or just mingling).

Becoming an active participant in OOPSLA is much easier than you think – identify the events that match your interests and work, review the submission requirements from the OOPSLA 2005 web site, and submit your contribution using the OOPSLA on-line submission system. If you have ideas for further OOPSLA activities, such as one-day symposia or student activities, please email a proposal to the Conference Chair for consideration.

We strongly encourage you to become a contributor to OOPSLA 2005 and to take part in shaping the future of object technology (as well as OOPSLA). For more information about actively participating in OOPSLA 2005, and for submission deadlines and details, please visit the OOPSLA 2005 site or contact the OOPSLA 2005 Office.

## OOPSLA 2005 Important Dates

### **March 11, 2005**

Submission deadline for Technical Papers, Onward! Papers, Essays, Lightning Talks, Practitioner Reports, Educators’ Symposium Submissions

### **March 18, 2005**

Tutorials Proposals, Panels Proposals, Workshops Proposals, and DesignFest Proposals.

### **July 1, 2005**

Submission deadline for Posters, Demonstrations, Doctorial Symposium, and Student Volunteers.

## OOPSLA 2005 Contact Information

**Conference Chair: Ralph Johnson, *University of Illinois***

**Program Chair: Richard P. Gabriel, *Sun Microsystems, Inc.***

### ***OOPSLA 2005 Office***

6220 NE Glisan  
Portland, OR 97213 U.S.A.  
E-mail: [info@oopsla.org](mailto:info@oopsla.org)  
Voice: +1-503-252-5709  
Fax: +1-503-261-0964

# David N. Smith



David N. Smith, a founder of the OOPSLA conference, passed away on April 13th, 2004. Dave, a long time member of the technical staff at IBM's Thomas J. Watson Research Center, coordinated the famous "Armonk" meeting of August 1985, at which the idea of OOPSLA was born. He remained active in the OOPSLA community, serving as conference treasurer in 1991 and 1992. He authored two well-known books on object technology—IBM Smalltalk: The Language and Concepts of Object-Oriented Programming—and remained active in the Squeak list until his death. He will be missed.