

INVITATION

Conference Chair:

Linda M. Northrop,
Software Engineering Institute



Welcome to OOPSLA 2001! We are very glad to have you here at the waterfront in sunny Tampa for this the sixteenth edition of our conference that celebrates the world of object-oriented systems, languages, and applications. Over two hundred people (mostly volunteers) have devoted their time, energy, and ideas to organize this conference, with the objective of making OOPSLA 2001 an outstanding experience for you.

OOPSLA is well known as the premier conference for object technology. The wide variety of events and activities make it appealing to object experts and novices alike. Object technologists from all over the world have come together to present interactive workshops, an outstanding set of tutorials, thought-provoking invited talks on a broad range of topics, panels and debates designed to surface controversy and provoke audience participation, technical papers on the latest object research, practitioners' reports sharing lessons learned from development experiences, and two "Intriguing Technology" sessions that promise to be of interest to both researchers and practitioners. There are also a number of specialized activities – symposia for Ph.D. students to get feedback on their research ideas (the Doctoral Symposium) and for academic and industrial educators to share new ideas and best practices (the Educators' Symposium) as well as our ever-popular DesignFest and CodeFest. And you can organize your own Birds of a Feather session here at the conference.

In the OOPSLA Courtyard you will find commercial exhibits of the latest products, news, and publications, a new "geek alley" where you can pick up your favorite personal devices and software, posters and demonstrations that present late-breaking results (including summaries of some pre-conference events), Birds of a Feather sign-ups, job and message boards, and tables for impromptu demos or discussions. And don't miss the opportunity to network with your colleagues at the social events. You should also be sure to leave some personal time to explore the Tampa area and enjoy some of Florida's many recreational areas.

This is a busy and exciting conference. If this is your first OOPSLA join us Monday evening after the Welcome Reception for the Newcomer's Session. We will answer your questions and give you some tips for organizing your activities.

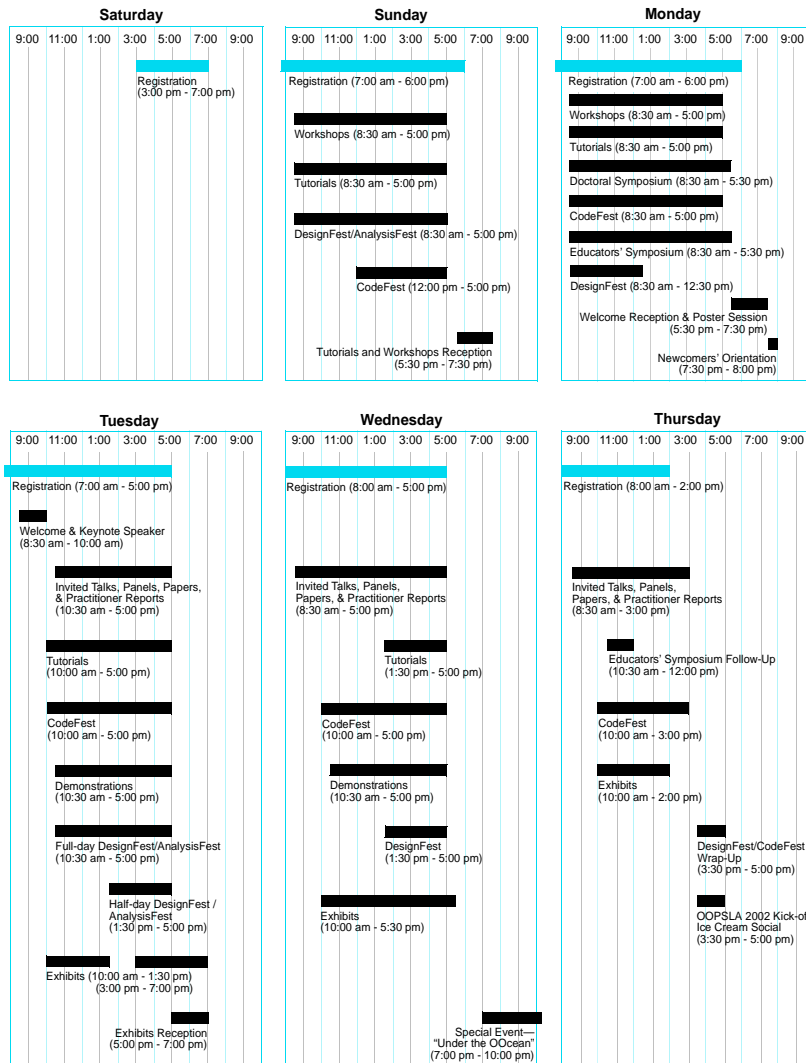
I sincerely thank the authors, presenters, exhibitors, the entire conference committee, program committee, and all the other volunteers who make OOPSLA possible. I also thank the OOPSLA Steering Committee, SIGPLAN, SIGSOFT, and ACM for sponsoring the conference and keeping us on track, while encouraging our innovations. Finally, I thank you, the conference attendees. You make OOPSLA a success. We are grateful for your continuing, enthusiastic support. Enjoy the conference.

INVITATION

Table of Contents

TECHNICAL PROGRAM	5
TUTORIALS	31
WORKSHOPS	111
DEMONSTRATIONS	151
POSTERS.....	163
EDUCATORS' SYMPOSIUM	177
DOCTORAL SYMPOSIUM.....	184
STUDENT VOLUNTEERS	186
BIRDS OF A FEATHER SESSIONS (BOFs)	186
DESIGNFEST	187
CAMP SMALLTALK	190
EXHIBITS	191
SPECIAL EVENTS.....	192
INFORMATION AND MAPS	193
TAMPA FACTS	198
ACM / SIGPLAN / SIGSOFT.....	200
OOPSLA 2001 COMMITTEES	201

OOPSLA 2001 AT A GLANCE



CANCELLATIONS:

Tutorial 5	Sunday
Tutorial 13	Sunday
Tutorial 23	Sunday
Tutorial 30	Monday
Tutorial 50	Tuesday
Tutorial 61	Wednesday
Tutorial 70	Wednesday
Workshop 8	Sunday
Workshop 26	Monday
Workshop 29	Monday

TECHNICAL PROGRAM

TECHNICAL PROGRAM

**Program Chair:**

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

Practitioner Reports Chair:

Jim Doble, *Tavve Software Company*

Panels Chair:

Ron Crocker, *Motorola, Inc.*

OOPSLA 2001 continues the tradition of quality and diversity in technical offerings. Here is a brief guide to the sessions and events comprising the technical program:

- Invited talks provide a forum for world-class speakers to present expert yet accessible accounts of important and timely topics.
- Technical paper presentations discuss new research in object-oriented languages, systems, and applications. The 27 technical papers were selected after a rigorous peer review of 145 submissions. Some of these papers present work that foreshadows the future of OO systems; others describe and analyze the underpinnings of common OO languages, design methods, and applications.
- Practitioner Reports relate how OO languages, systems, middleware, methods, and tools are used in practice. Some of these reports describe what it's like to adopt a new language, apply a new methodology, or use a new tool. Others distill the experience of one or more projects. The reports describe both benefits and drawbacks of approaches used in practice.
- Poster sessions present technical contributions in an informal setting. These sessions contain some of the newest research, often still in-progress. Authors are present to discuss their work, usually one-on-one.
- Demonstration sessions offer a chance to see software in action in a similarly informal (and commercial-free) context. In many cases, demos show the software discussed in technical sessions, tutorials, and other presentations.
- Panel sessions are the place where experts are encouraged to disagree publicly. The six panel sessions this year cover a spectrum of issues in software development, from understanding the impact of research to the future of object computing.
- DesignFest and CodeFest are our most hands-on events. Participants work in teams to design and implement solutions to assigned problems. The participants present their results and lessons learned at a wrap-up session at the end of the conference.
- This year we introduce two sessions in a new crossover track dubbed "Intriguing Technology." Designed to appeal to both researchers and practitioners, Intriguing Technology presentations describe best practices, introduce systems that make research ideas more widely accessible, discuss experiences that might spur further research—in short, any work that's "intriguing" to the OOPSLA audience.

In addition to these, as well as Tutorials, Workshops, Exhibits, and social events, there are two embedded symposia at OOPSLA:

- The Doctoral Symposium provides a forum for graduate students to share ideas and receive guidance.
- The Educators' Symposium includes presentations and discussions on teaching object-oriented technology in both academics and industry.

There's something here for everyone who works with object-oriented software. Whether you're a programmer or technical lead, student or faculty member, researcher or practitioner—you stand to benefit greatly from the state-of-the-art technologies and practices on display at OOPSLA 2001.

TECHNICAL PROGRAM AT A GLANCE

Tuesday, October 16, 2001

8:30 am - 10:00 am	Welcome and Introduction Conference Chair: Linda Northrop Program Chair: John Vlissides Keynote Address: Henry Petroski, "Success and Failure in Design"	Convention Center Ballroom B-C
10:30 am - 12:00 pm	Session A Panel: Somewhere Between Anarchy and Monarchy: Choosing the Appropriate Process for Your Next Project Session B Papers: Languages I Session C Practitioner Reports: Distributed Systems	Convention Center Ballroom B-C Convention Center Ballroom A Convention Center Ballroom D
1:30 pm - 3:00 pm	Session A Invited Speaker: Tom DeMarco, "Software: The New Realities" Session B Papers: Program Analysis Session C Papers: Performance	Convention Center Ballroom B-C Convention Center Ballroom A Convention Center Ballroom D
3:30 pm - 5:00 pm	Session A Panel: How Do Requirements Relate to Objects? Session B Papers: Resource Management Session C Practitioner Reports: Software Reuse	Convention Center Ballroom B-C Convention Center Ballroom A Convention Center Ballroom D

Wednesday, October 17, 2001

8:30 am - 10:00 am	Session A Invited Talk: Don Box, "Pervasive XML: Infoset-Based Software Integration"	Convention Center Ballroom B-C
10:30 am - 12:00 pm	Session A Panel: The IMPACT Project: Determining the Impact of Software Engineering Research upon Practice Session B Papers: Dynamic Optimization Session C Intriguing Technology Papers: Expressive and Adaptive Systems	Convention Center Ballroom B-C Convention Center Ballroom A Convention Center Ballroom D
1:30 pm - 3:00 pm	Session A Invited Speaker: L. Peter Deutsch, "Software Transparency and Object Technologies" Session B Papers: Modules Session C Practitioner Reports: Applied Techniques	Convention Center Ballroom B-C Convention Center Ballroom A Convention Center Ballroom D
3:30 pm - 5:00 pm	Session A Panel: The OO Survivor Game Session B Papers: Languages II Session C Intriguing Technology Papers: TACO and SALSA	Convention Center Ballroom B-C Convention Center Ballroom A Convention Center Ballroom D

Thursday October 18, 2001

8:30 am - 10:00 am	Session A Invited Talk: Alistair Cockburn, “Harnessing Convection Currents of Information”	Convention Center Ballroom B-C
10:30 am - 12:00 pm	Session A Panel: The Future of Distributed Objects — .NET vs Java	Convention Center Ballroom B-C
	Session B Papers: Software Engineering	Convention Center Ballroom A
	Session C Balloon Debate: Educators’ Symposium Follow-Up: Empowering People: What Are the Most Important Criteria for Effectively Training People in OT?	Convention Center Ballroom D
1:30 pm - 3:00 pm	Session A Panel: Is “Software Engineering” the Wrong Metaphor? And Why Should We Care?	Convention Center Ballroom B-C
	Session B Papers: Garbage Collection	Convention Center Ballroom A
3:30 pm - 5:00 pm	OOPSLA 2002 Kick-off (Ice Cream Social) DesignFest/CodeFest Wrap-Up	Convention Center Ballroom D

8:30 am - 10:00 am

Convention Center — Ballroom B-C

Welcome and Introduction

Conference Chair:
Linda Northrop,
Software Engineering Institute

Program Chair:
John Vlissides,
IBM T.J. Watson Research Center

Keynote Address

Success and Failure in Design



Henry Petroski,
A.S. Vesic Professor of Civil Engineering and
Professor of History, Duke University

This illustrated lecture will present a historical perspective on the interrelationship between success and failure in design. Case studies will range from ancient Greek and Roman construction practices to modern American packaging and merchandizing, emphasizing the timelessness and cultural independence of design principles as applied to objects of all kinds.

Henry Petroski is the Aleksandar S. Vesic Professor of Civil Engineering and a professor of history at Duke University. Before moving to Duke in 1980, he was on the faculty of the University of Texas at Austin and on the staff of Argonne National Laboratory.

*Petroski has written on many aspects of engineering and technology, including design, success and failure, error and judgment, the history of bridges, and the use of case studies in education and practice. His books on these subjects, which are intended for professional engineers and laypersons alike, include *To Engineer Is Human*, which was adapted for a BBC-television documentary; *The Pencil*; *The Evolution of Useful Things*; *Design Paradigms*; *Engineers of Dreams*; *Invention by Design*; and *Remaking the World*. Among the languages into which his books have been translated are Chinese, Finnish, German, Hebrew, Korean, Japanese, and Spanish. His latest book, a history of books as artifacts and the structures that have housed them from ancient times to the computer age, is entitled *The Book on the Bookshelf*.*

*In addition to publishing the usual technical articles in the refereed journals of his field, Petroski has published numerous articles and essays in newspapers and magazines, including the *New York Times*, *Washington Post*, *Wall Street Journal*, and *Scientific American*. Since 1991 he has been writing the engineering column in the bimonthly magazine *American Scientist*, and since 2000 has been writing a column on the engineering profession for *ASCE Prism*.*

Henry Petroski has been a Guggenheim Fellow, a National Endowment for the Humanities Fellow, and a Fellow of the National Humanities Center. Among his other honors are the Ralph Coats Roe Medal from the American Society of Mechanical Engineers, the Civil Engineering History and Heritage Award from the American Society of Civil Engineers, honorary degrees from Clarkson University, Trinity College (Hartford, Conn.) and Valparaiso University, and distinguished engineering alumnus awards from Manhattan College and the University of Illinois at Urbana-Champaign. He is a Fellow of the American Society of Civil Engineers and the Institution of Engineers of Ireland and is a member of the National Academy of Engineering.

10:30 am - 12:00 noon

Convention Center — Ballroom B-C

Session A

Panel: Somewhere Between Anarchy and Monarchy: Choosing the Appropriate Process for Your Next Project

Moderator: John Daniels, *Syntropy, Ltd.*

The rise of the agile process movement has received a good deal of attention lately, but many questions remain regarding when such process models are appropriate. Determining which process model is the right one for your project is a critical factor in success. In this panel, the topic of when and why a given process model should be applied will be discussed. To maximize the value of the discussion, panelists will be asked to respond to specific questions based on real-world issues.

Panelists:

Ron Jeffries, *Object Mentor, Inc.*

Craig Larman, *Valtech*

Ivar Jacobson, *Rational Software Corp.*

Alistair Cockburn, *Humans and Technology*

Jim Highsmith, *Information Architects, Inc.*

Bob Marcus, *Rogue Wave*

Convention Center — Ballroom A

Session B

Papers: Languages I

Chair: Joseph Kiniry, *California Institute of Technology*

The papers in this session are excellent examples of theory successfully applied to OO problems. Findler and Felleisen address contract soundness with respect to annotated Java in its various incarnations. Their results will be relevant to anyone applying formal specification to Java using tools like iContract, JML, and Jass. Ancona, et al., solidify the type theoretic and algebraic foundations of Java's semantics, a highly pertinent area given the popularity of that language. Finally, Bachrach shows how to do macros "right" in Java—à la Dylan and Lisp, with several new twists.

Contract Soundness for Object-Oriented Languages

Robert Bruce Findler, *Rice University*

Matthias Felleisen, *Rice University and Northeastern University*

A Core Calculus for Java Exceptions

Davide Ancona, *DISI – Università di Genova*

Giovanni Lagorio, *DISI – Università di Genova*

Elena Zucca, *DISI – Università di Genova*

The Java Syntactic Extender

Jonathan Bachrach, *MIT AI Lab*

Keith Playford, *Functional Objects, Inc.*

Convention Center — Ballroom D

Session C

Practitioner Reports: Distributed Systems

Chair: Gerard Meszaros, *ClearStream Consulting*

Distributed architectures are both a blessing and a curse: With careful planning and design, they can help with reliability and scalability, but “getting it right” is no small challenge. In this session, two reports describe how distributed systems were successfully developed to provide scalable business applications, and the third author presents an approach for certifying performance of a distributed system.

SPCQuote: The Evolution of a Client Application to Distributed Service Processing

Karen Hope, *The St. Paul, Inc.*

John Finegan, *The St. Paul, Inc.*

The Business Foundation System (BFS) writes and manages small business commercial insurance policies for The St. Paul Fire & Marine insurance companies. The original system, deployed in 1996, was a client-based VSE Smalltalk application. By the time of its third release, it supported 120 internal users in three locations. Since its deployment the system has had over 30 subsequent releases to increase and improve its functionality, the most notable of these occurring with BFS’s SPCQuote release in April 2000. SPCQuote significantly changed both the system architecture and its development environment and satisfied many disparate key business objectives. Today the system supports over 7500 users in over 3200 locations. It manages over 120,000 policies with policy premiums totaling over \$500,000,000. This paper describes the evolution of the smart-client VSE Smalltalk application to an n-tier distributed service model supported by two Smalltalk dialects, Java servlets running within Websphere, and XML as a means of data abstraction.

Business Automation with Distributed Objects

Jason Rogers, *Ontario Teachers’ Pension Plan Board*

Dean Mackie, *Ontario Teachers’ Pension Plan Board*

Angus MacArthur, *Ontario Teachers’ Pension Plan Board*

A large financial institution was faced with the challenge of having to process twice the normal yearly workload without increasing their workforce. The challenge was met by reusing components of an existing object-oriented application in a distributed, fully automated configuration. That solution is discussed, along with subsequent system architecture evolution, resulting improvements to development practices, and the ripple effects of changing the culture of senior management that it was designed to serve.

Certifying Component Performance in Synchronous Distributed Client/Server Systems

Dong-Lih Denq, *AT&T-Network Services*

Irwin Dunietz, *AT&T-Network Services*

John Eddy, *AT&T-Network Services*

Willa Ehrlich, *AT&T-Network Services*

Don Gerth, *AT&T-Network Services*

Brian Larson, *AT&T-Network Services*

Geetha Sivaprasad, *AT&T-Network Services*

This paper describes an approach for certifying performance of a distributed system that utilizes synchronous interprocess communication. The system is composed of multiple managed objects implemented in object server processes (i.e., components) that communicate via CORBA middleware. Component performance is certified empirically through “pairwise” testing that emphasizes interactions between a component and its lower-level servers. The applicability of this approach is demonstrated for an AT&T provisioning system.

1:30 pm - 3:00 pm

Convention Center — Ballroom B-C
Session A**Invited Talk: Software: The New Realities****Tom DeMarco,**
Atlantic Systems Guild

We live in an age of acceleration. Whatever the formula was for business success a few years ago, it won't work today. Today there needs to be more and more work crammed into less and less time. There are fewer people doing more and doing it faster in less space with less support and with tighter tolerances and higher quality requirements than ever before. So we have spent the last decade becoming ever more efficient. Now enter the need for change. In the super-accelerated corporation, meaningful change of direction is almost impossible. The very improvements that we made to go faster and cheaper have undermined our capacity to make any other kind of change. An organization that can accelerate but not change direction is like a car that can speed up but not steer. In the short run it makes lots of progress in whatever direction it happened to be going. In the long run it's just another road wreck.

Tom DeMarco is a Principal of the Atlantic Systems Guild, a New York and London-based consulting practice. He is the author of seven books on software method and management, including the recent Random House book Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency. Past titles include The Deadline: A Novel About Project Management, as well as PEOPLEWARE: Productive Projects and Teams (with co-author T. Lister). In 1986 he was named winner of the J.D. Warnier Prize for "lifetime contribution to the information sciences." He won the 1999 Stevens Prize for contribution to software methods. He is the author of a mainstream novel, Dark Harbor House (Down East Books, 2000). His consulting activity is mostly in the area of project management and litigation involving software-intensive endeavors.

Convention Center — Ballroom A Session B

Papers: Program Analysis

Chair: Peter F. Sweeney, *IBM T.J. Watson Research Center*

This year's program upholds the OOPSLA tradition of high-quality contributions to program analysis. The first paper in this session uses a constraint-based approach to implement points-to analysis for Java. For a restricted Java-like language, the second paper guarantees that well-typed programs are race-free, even as it allows objects from the same class to use different locking mechanisms. The last paper describes a combination of static and run-time techniques for detecting potential data races in concurrent Java programs.

Points-To Analysis for Java using Annotated Constraints

Atanas Rountev, *Rutgers University*

Ana Milanova, *Rutgers University*

Barbara G. Ryder, *Rutgers University*

A Parameterized Type System for Race-Free Java Programs

Chandrasekhar Boyapati, *MIT*

Martin Rinard, *MIT*

Object Race Detection

Christoph von Praun, *ETH Zürich*

Thomas Gross, *ETH Zürich*

Convention Center — Ballroom D Session C

Papers: Performance

Chair: Brian Foote, *The Refactory, Inc.*

Programmers often shun programming language features they see as exhibiting poor or unpredictable performance. This session highlights three advanced implementation techniques that promise to relieve programmers of such concerns. The first paper shows how exceptions in Java can be implemented with unexceptional overhead. The second paper examines how to minimize the cost of asking a deceptively simple question: Is this type a subtype of another? Sophisticated implementation strategies described in the third paper nearly eliminate the customary performance penalty of Java interface invocations.

A Study of Exception Handling and Its Dynamic Optimization in Java

Takeshi Ogasawara, *IBM Tokyo Research Laboratory*

Hideaki Komatsu, *IBM Tokyo Research Laboratory*

Toshio Nakatani, *IBM Tokyo Research Laboratory*

Efficient Subtyping Tests with PQ-Encoding

Yoav Zibin, *Technion*

Joseph (Yossi) Gil, *Technion*

Efficient Implementation of Java Interfaces: Invokeinterface Considered Harmless

Bowen Alpern, *IBM T.J. Watson Research Center*

Anthony Cocchi, *IBM T.J. Watson Research Center*

Stephen Fink, *IBM T.J. Watson Research Center*

David Grove, *IBM T.J. Watson Research Center*

Derek Lieber, *IBM T.J. Watson Research Center*

3:30 pm - 5:00 pm

Convention Center — Ballroom B-C

Session A

Panel: How Do Requirements Relate to Objects?

Moderator: Hermann Kaindl, *Siemens AG Austria*

While most object-oriented methods today include something like object-oriented analysis (OOA), they do not say much about requirements. Typically, they treat requirements just as a textual input in the form of a short problem statement. In the field of requirements engineering, in contrast, object-oriented modeling is often considered as a means to create and represent a requirements model. Apparently, the relationship between requirements and objects remains an open question.

In the light of such controversial views, this panel intends to discuss the following questions (among others):

- Are requirements just preliminary textual statements?
- Can requirements be modeled as objects?
- Can an OOA model serve as a requirements model?

Panelists:

Ivar Jacobson, *Rational Software Corp.*

Craig Larman, *Valtech*

Joaquin Miller, *Financial Systems Architects*

Rebecca Wirfs-Brock, *Wirfs-Brock Associates*

Desmond D'Souza, *Kinetium*

Steve Mellor, *Project Technology, Inc.*

Convention Center — Ballroom A

Session B

Papers: Resource Management

Chair: Doug Lea, *SUNY Oswego*

When applications share resources, infrastructure services and virtual machines must help prevent interference. They must also manage resource consumption and optimize resource access among the competing activities. The authors of the first paper illustrate how JVMs can isolate applications to appear as if they were separate operating system processes. The second paper explains how to limit resource consumption in Java applications. The session's closing paper describes how to maximize the efficiency of database resources by optimizing queries.

Multitasking without Compromise: A Virtual Machine Evolution

Grzegorz Czajkowski, *Sun Microsystems Laboratories*

Laurent Daynès, *Sun Microsystems Laboratories*

Portable Resource Control in Java: The J-SEAL2 Approach

Walter Binder, *CoCo Software Engineering*

Jarle Hulaas, *University of Geneva*

Alex Villazón, *University of Geneva*

Incremental Computation of Complex Objects Queries

Hiroaki Nakamura, *IBM Tokyo Research Laboratory*

Convention Center — Ballroom D

Session C

Practitioner Reports: Software Reuse

Chair: Jim Doble, *Tavve Software Company*

Many organizations have looked to object technology as a means to achieve significant gains in software reuse. Results, however, have run the gamut from stunning success to unmitigated disaster. The first paper in this session presents a survey of results from a variety of reuse-oriented projects within Mentor Graphics Corporation over the past 12 years. The last two papers describe experiences and lessons learned on individual projects involving the development of reusable frameworks.

Mentor Graphics Corporation's Experiences with Object-Oriented Software Technology Reuse

Phil Brooks, *Mentor Graphics Corporation*

Steve Aho, *Mentor Graphics Corporation*

Since 1989, Mentor Graphics Corporation has undertaken a number of large-scale projects where software reuse, in various forms, was a major goal. A team of seven senior software development engineers recently examined software reuse practices within Mentor Graphics in order to figure out what works, what doesn't, and why. Several actual projects, some that were successful and others that were not, are reviewed. The findings from these project experiences, along with the combined experiences of the members of the team, lead to conclusions on how organizational structure combined with the various reuse strategies presented aid or hinder reuse.

Construction of an Object-Oriented Framework for Text Mining

James W. Cooper, *IBM T. J. Watson Research Center*

Edward C. So, *IBM T. J. Watson Research Center*

Christian L. Cesar, *IBM T. J. Watson Research Center*

Robert L. Mack, *IBM T. J. Watson Research Center*

This paper describes the construction of a Java class library for text mining and information retrieval. The system consists of Facades around a database, a search engine, and a text mining tool. The authors discuss the design of the object models used for each of these elements, how these object models evolved as different databases and search engines became available, how they needed to evolve the system further in work with their customer, and the eventual fate of the system after the customer adopted the final version of the code, showing what was learned from the experience.

Applying Design Patterns and Other Technologies to On-Line Application Servers for Bill Payment, Bill Delivery, and Banking

Don Barton, *CheckFree Corporation*

This paper discusses how an organization gained high levels of reuse and developer concurrency, as well as scalability and availability by leveraging techniques from OOPSLA and other sources. The author discusses how a team and a process were built around the use of design patterns, frameworks, metaobject protocols, refactoring, code generation, and a custom build environment with integrated testing.

8:30 am - 10:00 am

**Convention Center — Ballroom B-C
Session A****Invited Talk: Pervasive XML: InfoSet-Based Software Integration****Don Box, DevelopMentor**

The XML InfoSet is XML's underlying data model and is increasingly being used to put XML into scenarios that would otherwise be considered too performance-sensitive to support a text-oriented solution. This talk looks at the role of the classic XML InfoSet as well as the more modern Post-Schema Validation InfoSet (PSVI) as a way of exposing applications, components, and data in general to consumers written in arbitrary languages running on arbitrary platforms.

Don Box is a co-founder of DevelopMentor, a developer services company that provides education and support to the software industry at large. Don's research interests include component software integration, programming for concurrency, and XML-based serialization and metadata protocols. Don is a series editor at Addison-Wesley and is the author of Essential COM and a co-author of Effective COM and Essential XML, all from Addison-Wesley. Don is a contributing editor and columnist at Microsoft Systems Journal (now called MSDN Magazine) and an occasional contributor to XML.com. Don is also a co-author of the Simple Object Access Protocol specification and a member of the W3C Schemas Working Group. Don has a Master's Degree in Computer Science from the University of California at Irvine.

10:30 am - 12:00 noon

Convention Center — Ballroom B-C

Session A

Panel: The IMPACT Project: Determining the Impact of Software Engineering Research upon Practice

Moderator: Dr. Leon J. Osterweil
University of Massachusetts

This panel looks both backward and forward at the effect of software engineering research on software development practice. It will evaluate the impact of software engineering to date, identifying the sorts of contributions that have had substantial impact and the research modalities that have been more successful, and then explore directions that software engineering research might most effectively pursue to meet the challenging demands of the future.

Scheduled panelists:

Dave Thomas, *Bedarra Corp. and Carleton University*

Richard Gabriel, *Sun Microsystems, Inc.*

Alexander L. Wolf, *University of Colorado, Boulder*

Barbara G. Ryder, *Rutgers University*

Convention Center — Ballroom A

Session B

Papers: Dynamic Optimization

Chair: Craig Chambers, *University of Washington*

Dynamic compilation, as found in Java “just-in-time” compilers, affords many new optimization opportunities compared to traditional static compilation. “Partial Method Compilation using Dynamic Profile Information” uses on-line profiling of basic-block execution frequencies to re-optimize only the hot paths of methods dynamically. “A Dynamic Optimization Framework for a Java Just-In-Time Compiler” presents a thorough study of a state-of-the-art Java system and argues for including an interpreter as a component of a Java virtual machine in contrast to compile-only JVMs. “Dynamic Optimistic Interprocedural Analysis: A Framework and an Application” uses a flexible dependency framework to incorporate a simple whole-program analysis safely in the dynamic compiler—even in the face of dynamic class loading.

Partial Method Compilation using Dynamic Profile Information

John Whaley, *Stanford University*

A Dynamic Optimization Framework for a Java Just-In-Time Compiler

Toshio Suganuma, *IBM Tokyo Research Laboratory*

Toshiaki Yasue, *IBM Tokyo Research Laboratory*

Motohiro Kawahito, *IBM Tokyo Research Laboratory*

Hideaki Komatsu, *IBM Tokyo Research Laboratory*

Toshio Nakatani, *IBM Tokyo Research Laboratory*

Dynamic Optimistic Interprocedural Analysis: A Framework and an Application

Igor Pechtchanski, *New York University*

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

Convention Center — Ballroom D

Session C

Intriguing Technology Papers: Expressive and Adaptive Systems

Chair: Brent Hailpern, *IBM T.J. Watson Research Center*

In expressive systems, core business objects show through directly to users, and all user actions are initiated through a noun-verb style of interaction on those objects. Having users and developers speak a common language improves the process of requirements analysis and prototype development.

Adaptive object models form an approach to object-oriented information systems emphasizing flexibility and dynamic configurability. Business rules are stored in a database instead of in code. The object model that the user cares about is part of the database, and the object model of the code is just an interpreter of the users' object model.

Expressive Systems: A Radical Approach to Business Systems Design

Richard Pawson, *Computer Sciences Corporation*

Simon Dobson, *Trinity College, Dublin*

Architecture and Design of Adaptive Object-Models

Joseph Yoder, *University of Illinois at Urbana-Champaign*

Federico Balaguer, *University of Illinois at Urbana-Champaign*

Ralph Johnson, *University of Illinois at Urbana-Champaign*

1:30 pm - 3:00 pm

Convention Center — Ballroom B-C

Session A

Invited Talk: Software Transparency and Object Technologies



L. Peter Deutsch, Aladdin Enterprises

We define software as transparent to the extent that it is *conceived, developed, documented, licensed, distributed, and cared for* to intentionally facilitate *reading, understanding, analysis, validation, confidence; repair, adaptation, extension, evolution; interoperation, integration, incorporation; sharing; and use*. Software transparency, in all its aspects, is a key to the usefulness of software over its lifetime.

The *Open Source Definition* codifies an important set of rules that address the licensing (legal) aspect of transparency; the other aspects are either considered business issues, or issues of technological “best practice.” Transparency is rarely considered as a gestalt.

When objects are used well as a software engineering technology for analysis and design within a program, the technology naturally encourages many aspects of transparency. On the other hand, objects can also be used to create opaque boundaries around functionality and data. We will do our best to refute the arguments of security, long-term evolvability, and business advantage often used to justify such boundaries.

Dr. L. Peter Deutsch received the Ph.D. in Computer Science from U.C. Berkeley in 1973. Subsequently at Xerox PARC, he helped develop the Interlisp-D, Cedar Mesa, and Smalltalk-80 programming systems, and was the principal creator of PS, the first high-performance implementation of the Smalltalk language and programming environment on microprocessor-based hardware. From 1986 to 1991, as Chief Scientist at ParcPlace Systems, he was a principal designer of a highly-portable successor to PS. Since 1986, he has been an independent consultant (dba Aladdin Enterprises), best known as the creator of Ghostscript, a highly portable, high-quality, Open Source implementation of the PostScript language. In 1993, he was a co-recipient of the ACM Software System Award, and was also named a Distinguished Alumnus of the U.C. Berkeley Computer Science program. In 1999-2000, he served on the board of the Open Source Initiative. He is a member of ACM, IEEE, CPSR, EFF, and the League for Programming Freedom.

Convention Center — Ballroom A Session B

Papers: Modules

Chair: Krzysztof Czarnecki, *DaimlerChrysler Research and Technology*

Objects are the programming model of choice for implementing components. Yet there's always room for improvement in modularity and encapsulation, especially for component-based development. The session's opening paper describes a rich model and attendant language for building components that goes beyond the simple class-as-a-component paradigm while allowing easy integration into an object-oriented language such as Java. Packaging classes and class fragments into mixin modules is the thrust of the second paper, which presents a type-safe approach to programming using such modules. The last paper relates a static analysis technique for improving encapsulation by detecting objects that escape from packages.

Jiazzi: New-Age Components for Old-Fashioned Java

Sean McDirmid, *University of Utah*

Matthew Flatt, *University of Utah*

Wilson C. Hsieh, *University of Utah*

Modular Mixin-Based Inheritance for Application Frameworks

Dominic Duggan, *Stevens Institute of Technology*

Ching-Ching Techaubol, *Stevens Institute of Technology*

Encapsulating Objects with Confined Types

Christian Grothoff, *Purdue University*

Jens Palsberg, *Purdue University*

Jan Vitek, *Purdue University*

Convention Center — Ballroom D

Session C

Practitioner Reports: Applied Techniques

Chair: Granville Miller, *TogetherSoft*

A lot of ideas and approaches get discussed at an OOPSLA conference, but do they really work in the real world? The three reports in this session share insights and advice gained from actual application of a variety of OOPSLA topics: UML-based analysis and design, test-first design, and software frameworks.

Closing the Software Development Gaps with UML

Brian Berenbach, *Siemens Corporate Research, Inc.*

This paper presents an approach that has been used successfully on large projects, leveraging UML to successfully bridge the analysis-design and design-implementation gaps in the Software Development Lifecycle (SDLC). In addition, a set of guidelines for determining the “completeness” of the analysis model is introduced.

Evolution of Test and Code via Test-First Design

Jeff Langr, *Object Mentor, Inc.*

This paper demonstrates how test coverage and code quality is improved through the use of test-first design, starting with an example of code written without the use of automated tests, then iterating through the exercise of completely rebuilding the code, test by test, and comparing the results of the two approaches.

Object-Based Frameworks For Mobile Robots

Jeffrey Will, *Dept. of Electrical and Computer Engineering, Valparaiso University*

Donald Roberts, *The Refactory, Inc.*

John Reid, *Deere & Company Technical Center, John Deere*

Noboru Noguchi, *School of Agriculture, Hokkaido University*

Eric Benson, *Bioresource Engineering, University of Delaware*

This paper describes the underlying software framework used to develop several outdoor autonomous robots. This framework was shown to exhibit great amounts of reuse, supported parallel platforms and parallel applications, reduced training times for new contributors, and was successfully implemented in a wide variety of applications.

3:30 pm - 5:00 pm

Convention Center — Ballroom B-C
Session A**Panel: The OO Survivor Game**Moderator: Laura Hill, *Sun Microsystems, Inc.*

Help Big Brother determine the fate of the stalwart contestants on “OO Survivor” as they compete to avoid the garbage collector on its merciless search to cull the weakest link after each round of play. Competitors will match wits and strain eyesight as they peer into the future to describe what they see in the areas of architecture, project management, language design, methodology, and distributed computing. Let the games begin!

Panelists:

Stephen J. Mellor, *Project Technology, Inc.*John Daniels, *Syntropy Limited, UK*Martine Devos, *Independent Consultant*Brian Foote, *The Refactory, Inc.*Joaquin Miller, *Financial Systems Architects*Kevin Tyson, *Ocean Power Corporation*

Convention Center — Ballroom A

Session B

Papers: Languages II

Chair: Clemens Szyperski, *Microsoft Research*

At its core, object-orientation appeals to simplicity—a simplicity that doesn’t necessarily cater to every problem in systems, languages, and applications. Refined language-level abstractions are one way to resolve the mismatch. This session presents three such abstractions. “On Objects and Events” introduces language-integrated support for object-oriented event-based systems. “Visitor Combination and Traversal Control” proposes abstractions for the flexible combination and factoring of visitors. In “Object-Oriented Composition Untangled,” the authors posit compound references as a mechanism for expressing and unifying conventional composition approaches such as inheritance, aggregation, and delegation.

On Objects and Events

Patrick T. Eugster, *Swiss Federal Institute of Technology*

Rachid Guerraoui, *Swiss Federal Institute of Technology*

Christian H. Damm, *University of Aarhus*

Visitor Combination and Traversal Control

Joost Visser, *CWI*

Object-Oriented Composition Untangled

Klaus Ostermann, *Siemens AG*

Mira Mezini, *Darmstadt University of Technology*

Convention Center — Ballroom D

Session C

Intriguing Technology Papers: TACO and SALSA

Chair: Peri Tarr, *IBM T. J. Watson Research Center*

TACO is a template-based object platform for cluster architectures that provides the flavor of distributed data-parallel programming based on distributed object groups. TACO supports polymorphism, genericity, functional programming, and concurrency control in the context of distributed parallel computing.

SALSA is an Actor-based programming language built atop of Java, supporting mobility and dynamically reconfigurable distributed object systems. SALSA’s programming features include asynchronous communication, universal naming, migration, and token-passing continuations.

TACO — Prototyping High-Level Object-Oriented Programming Constructs by Means of Template-Based Programming Techniques

Joerg Nolte, *Real World Computing Partnership*

Yutaka Ishikawa, *Real World Computing Partnership*

Mitsuhisa Sata, *Real World Computing Partnership*

Programming Dynamically Reconfigurable Open Systems with SALSA

Carlos Varela, *IBM T.J. Watson Research Center*

Gul Agha, *University of Illinois at Urbana-Champaign*

8:30 am - 10:00 am

Convention Center — Ballroom B-C
Session A**Invited Talk: Harnessing Convection Currents of Information****Alistair Cockburn, *Humans and Technology***

In developing software, we swim in and depend on a sea of information, trying to find the bit we need to continue. Sometimes it is all too much, sometimes it is not enough, or is of the wrong kind. Sometimes we have a sense that somewhere in the organization, someone has the information we need. It may even be we ourselves, if only we knew it.

This talk is aimed at giving you a way to work with what is all around you, but may not have good words for or did not know you could alter.

We shall reexamine software development as a cooperative game of invention and communication, organizing the convection currents of information, adding information radiators and eliminating drafts, to improve the rate at which we develop.

Alistair has been a field and theoretical methodologist for ten years, studying software developers the way ethnographers study tribal societies — by going native and creating theories, in alternation. He is known as the leading expert on use case writing and software project leadership patterns, and as one of the original authors of the Agile Software Development Manifesto. His recently constructed Crystal family of methodologies is what one evaluator called “the least constraining that could possibly work,” built around a deep belief in the frailties and strengths of people, and on harnessing convection currents of information.

10:30 am - 12:00 pm

Convention Center — Ballroom B-C
Session A**Panel: The Future of Distributed Objects — .NET vs Java**

With the recent introduction of several new distributed object technologies, the future of this area is at stake. This panel will debate the appropriateness of the various technologies in a variety of application spaces.

Moderator: Michael Stal, *Siemens AG*

Panelists:

Don Box, *DevelopMentor*

Brad Merrill, *Microsoft*

Sanjiva Weerawarana, *IBM T.J. Watson Laboratories*

Convention Center — Ballroom A Session B

Papers: Software Engineering

Chair: Jim Doble, *Tavve Software Company*

Software is among the most intangible of human artifacts, forcing programs to take on different guises as we attempt to depict, comprehend, and maintain them. The first paper shows, quite vividly, how class categories and relationships can be conveyed through graphical depictions the authors call “class blueprints.” The second paper examines how to identify a regression suite within a set of test cases so that the suite contains only the tests that exercise the latest changes. The final paper describes a UML environment in which both models and the meta-models that define them are built from objects. These objects, in turn, constitute the program, which can be executed directly.

A Categorization of Classes Based on the Visualization of Their Internal Structure: The Class Blueprint

Michele Lanza, *University of Berne*

Stéphane Ducasse, *University of Berne*

Regression Test Selection for Java Software

Mary Jean Harrold, *Georgia Institute of Technology*

James A. Jones, *Georgia Institute of Technology*

Tongyu Li, *Georgia Institute of Technology*

Donglin Liang, *Georgia Institute of Technology*

Alessandro Orso, *Georgia Institute of Technology*

Maikel Pennings, *Georgia Institute of Technology*

Saurabh Sinha, *Georgia Institute of Technology*

S. Alexander Spoon, *Georgia Institute of Technology*

Ashish Gujarathi, *Citrix Systems*

The Architecture of a UML Virtual Machine

Dirk Riehle, *SKYVA International*

Steven Fraleigh, *SKYVA International*

Dirk Bucka-Lassen, *Object Oriented, Ltd.*

Nosa Omorogbe, *SKYVA International*

Convention Center — Ballroom D Session C

Balloon Debate: Educators' Symposium Follow-Up

Empowering People: What Are the Most Important Criteria for Effectively Training People in OT?

Moderators:

Kevlin Henney, *Curbralan Ltd., UK*

Jutta Eckstein, *Objects in Action, Germany*

Professionals with experience on real projects, good communication skills, and the knowledge to exploit object technology aren't born that way—people have to be educated. What content and process criteria should quality training programs focus on to produce qualified people and ensure their ongoing professional development?

We must reach consensus on these criteria quickly, because our panelists happen to be in a hot air balloon that is rapidly losing altitude. To stay aloft, they must get rid of excess weight—and fast! Each person, representing a training criterion, will try to convince us that they're not expendable. Everyone is invited to participate by asking our balloonists questions and voting on whom to jettison.

Flying Off:

Bruce Anderson, *IBM EMEA Component Technology Services, UK*

Ken Auer, *RoleModel Software, Inc.*

Joseph Bergin, *Pace University*

Frank Buschmann, *Siemens AG, Germany*

Alan O'Callaghan, *De Montfort University, UK*

Alistair Cockburn, *Humans and Technology*

Nicolai Josuttis, *Solutions in Time, Germany*

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

Pete McBreen, *McBreen Consulting, Canada*

1:30 pm - 3:00 pm

Convention Center — Ballroom B-C
Session A**Panel: Is “Software Engineering” the Wrong Metaphor?
And Why Should We Care?**

Moderator: Alistair Cockburn, *Humans and Technology*

In an industry that is supposed to be driving the future, we see many recurring patterns of dysfunction and self-destructive behavior in the planning and execution of software development projects. Many have questioned typical practices of hiring, educating, and managing software-related professionals. Many of these problems seem to be rampant. Dilbert and many other parodies illustrate the recurring symptoms, but do they really identify the problem?

The response to undisciplined software development has been to search for a metaphor that helps the world envision a more disciplined approach to software that will produce reliable software at a predictable cost. It seems that “software engineering” is the metaphor that has stuck. Some argue that the plethora of problems we face in our industry stem from the use of the “software engineering” metaphor. Others argue that the problems stem from the misapplication of the metaphor.

Is it time for a new metaphor?

This panel explores the positive and negative roles the “software engineering metaphor” plays in various types of software development efforts and discusses the practicalities of other metaphors as a cure to the problems we face. The panelists offer widely diversified perspectives on the topic.

Panelists:

Pete McBreen, *McBreen Consulting, Canada*

Ken Auer, *RoleModel Software, Inc.*

Eugene Wallingford, *University of Northern Iowa*

Tom Ball, *Microsoft Research*

David Smith, *Timeline Computer Entertainment*

Convention Center — Ballroom A Session B

Papers: Garbage Collection

Chair: Satoshi Matsuoka, *Tokyo Institute of Technology*

Automated storage management in the form of garbage collection (GC) is becoming an integral part of object-oriented system implementation. Indeed, superior GC technology improves robustness, speeds execution, and makes better use of available memory. The first paper here shows that pretenuring (i.e., selectively bypassing generational collection by directly allocating objects in the “old” space) is effective when guided by profiling information. The second paper reexamines the classic heap exhaustion problem and its customary solutions—whether to grow the heap or to garbage-collect—in a modern Java setting, demonstrating a strategy that results in surprisingly efficient execution. The third paper presents a novel algorithm for reference counting in a multithreaded environment, virtually eliminating expensive inter-thread atomic operations through clever per-thread bookkeeping of pointer updates.

Pretenuring for Java

Stephen M. Blackburn, *University of Massachusetts*

Sharad Singhai, *University of Massachusetts*

Matthew Hertz, *University of Massachusetts*

Kathryn S. McKinley, *University of Massachusetts*

J. Eliot B. Moss, *University of Massachusetts*

Controlling Garbage Collection and Heap Growth to Reduce the Execution Time of Java Applications

Tim Brecht, *University of Waterloo and HP Labs*

Eshrat Arjomandi, *York University*

Chang Li, *York University*

Hang Pham, *York University*

An On-the-Fly Reference Counting Garbage Collector for Java

Yossi Levroni, *Microsoft*

Erez Petrank, *Technion*

3:30 pm - 5:00 pm

Convention Center — Ballroom D
Session A**OOPSLA 2002 Kick-off
(Ice Cream Social)****DesignFest/CodeFest Wrap-Up**

The Ice Cream social will also be the setting for the DesignFest/CodeFest Wrap-Up. Throughout the week, teams of software designers have taken part in the DesignFest sessions. In addition, a small number of student teams (CodeFest) have implemented some of these designs. The Ice Cream Social is a chance for OOPSLA participants, whether or not they participated in DesignFest, to see the designs produced during the earlier sessions, as well as to view demos of the final software. Designers will be available to discuss their experiences (both good and bad) of working in their design team, and the CodeFest teams will describe the problems they encountered while implementing the designs.

NOTES:

TUTORIALS

Chair: Craig Larman, *Valtech*

The OOPSLA conference is well known for its high quality tutorials that span a wide range of relevant and timely topics. OOPSLA 2001 has worked hard to continue this tradition by offering 70 carefully selected tutorials. The complete tutorial program covers a broad spectrum of topics selected to appeal to the interests of a wide range of participants, including practitioners, managers, and researchers. The tutorials are categorized below for ease of selection; individual tutorials may be found in multiple categories.

All tutorials include a lunch provided by OOPSLA.

TUTORIALS AT A GLANCE

Tutorials by Date

Sunday, 14 October 2001

Full Day – 8:30 am – 5:00 pm

- | | | |
|---|--|-----------------------------|
| 1 | A Brief Tour of Responsibility-Driven Design | Convention Ctr —
Room 20 |
| 2 | Testing Object-Oriented Software Systems | Convention Ctr —
Room 13 |
| 3 | Usage-Centered Design: An Agile Model-Driven Process for Object-Oriented User Interface Design | Convention Ctr —
Room 25 |
| 4 | Concepts of Object-Oriented Programming | Convention Ctr —
Room 22 |
| 5 | Lo-Fi Design Strategies for Creating Highly Usable Object-Oriented User Interfaces | Convention Ctr —
Room 24 |

Half Day – Morning – 8:30 am – 12:00 noon

- | | | |
|----|--|-------------------------------------|
| 6 | Inside High-Quality Software Architectures | Convention Ctr —
Room 18 |
| 7 | Dungeons and Patterns! | Marriott Hotel —
Meeting Room 11 |
| 8 | Introduction to Writing Use Cases | Marriott Hotel —
Florida Salon V |
| 9 | Object-Oriented Design of Human-Computer Interaction | Convention Ctr —
Room 15 |
| 10 | Introducing Patterns (or Any New Idea) into Organizations | Convention Ctr —
Room 14 |
| 11 | Introduction to Concurrent Object-Oriented Programming in Java | Convention Ctr —
Room 16 |
| 12 | Agile Methodologies | Convention Ctr —
Room 19 |

TUTORIALS AT A GLANCE

TUTORIALS

Sunday Morning (continued)

- | | | |
|----|--|--------------------------------------|
| 13 | How to Manage the Change from COBOL to OOP | Marriott Hotel —
Salon A |
| 14 | Component and Service Architecture Modeling with UML | Convention Ctr —
Room 21 |
| 15 | XML, XSD, and SOAP as a Better Component Model | Marriott Hotel —
Florida Salon VI |
| 16 | An Introduction to Design Patterns | Marriott Hotel —
Florida Salon IV |
| 17 | Producing GUIs with Java | Marriott Hotel —
Meeting Room 12 |

Half Day – Afternoon – 1:30 pm – 5:00 pm

- | | | |
|----|--|--------------------------------------|
| 18 | Designing Concurrent Object-Oriented Programs in Java | Convention Ctr —
Room 16 |
| 19 | Building Parsers with Java | Marriott Hotel —
Meeting Room 11 |
| 20 | Daily Builds Are for Wimps | Convention Ctr —
Room 15 |
| 21 | Designing with Patterns | Marriott Hotel —
Florida Salon IV |
| 22 | The .NET Framework: The Common Language Runtime
and C# | Marriott Hotel —
Florida Salon VI |
| 23 | Garbage Collection | Marriott Hotel —
Meeting Room 12 |
| 24 | Advanced Use Case Writing | Marriott Hotel —
Florida Salon V |
| 25 | Fractal Patterns and Frameworks in UML — Towards UML
2.0? | Convention Ctr —
Room 21 |

TUTORIALS

TUTORIALS AT A GLANCE

Monday, 15 October 2001

Full Day – 8:30 am – 5:00 pm

26	Aspect-Oriented Programming with AspectJ™	Convention Ctr — Room 20
27	Software Architecture: It's What's Missing From OO Methodologies	Convention Ctr — Room 13
28	Improving Your Use Cases	Convention Ctr — Room 22
29	Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects	Convention Ctr — Room 15
30	Object-Oriented Reengineering	Marriott Hotel — Meeting Room 12

Half Day – Morning – 8:30 am – 12:00 noon

31	Patterns at Work	Marriott Hotel — Florida Salon VI
32	Designing an Agile Methodology	Convention Ctr — Room 19
33	Exposing and Consuming Web Services with .NET	Convention Ctr — Room 21
34	Efficient Architectures for Object-Oriented Component-Based Middleware	Marriott Hotel — Salon A
35	Extreme Programming Live!	Marriott Hotel — Florida Salon IV
36	Patterns and Architectures for J2EE Systems	Marriott Hotel — Florida Salon V
37	Refactoring: Improving the Design of Existing Code	Convention Ctr — Room 16

Half Day – Afternoon – 1:30 pm – 5:00 pm

38	Efficient Implementation of Object-Oriented Programming Languages	Marriott Hotel — Salon A
39	Making the Software Process Transparent by Using Intelligent Agents	Marriott Hotel — Florida Salon V
40	Surviving Object-Oriented Projects	Marriott Hotel — Florida Salon VI
41	Refactoring to Patterns	Convention Ctr — Room 21
42	How to Really Fail at Software Architecture	Convention Ctr — Room 19

TUTORIALS AT A GLANCE

TUTORIALS

Tuesday, 16 October 2001

Full Day – 10:00 am – 5:00 pm (short lunch break)

- | | | |
|----|-------------------------------------|-------------------------------------|
| 43 | The Art of Writing Use Cases | Marriott Hotel —
Meeting Room 12 |
| 44 | Architecting Large Business Systems | Convention Ctr —
Room 25 |

Half Day – Afternoon – 1:30 pm – 5:00 pm

- | | | |
|----|--|-------------------------------------|
| 45 | The UML's Object Constraint Language (OCL) —
Specifying Components | Marriott Hotel —
Meeting Room 11 |
| 46 | J2ME Design and Development Considerations | Convention Ctr —
Room 16 |
| 47 | Embedded Systems in C++ — C++ Idioms, Patterns, and
Architecture for Constrained Systems | Convention Ctr —
Room 5 |
| 48 | Adaptive Object-Model Architecture: How to Build
Systems That Can Dynamically Adapt to New Business
Requirements | Convention Ctr —
Room 15 |
| 49 | No Stone Unturned: An Introduction to Test-First
Programming | Convention Ctr —
Room 3 |
| 50 | Designing Software Architecture for Quality: The ADD
Method | Convention Ctr —
Room 22 |
| 51 | Creating Responsive, Scalable Systems | Convention Ctr —
Room 13 |
| 52 | Leading Retrospectives on OO Projects: Looking Back to
Move Forward | Convention Ctr —
Room 14 |
| 53 | Business Modeling with the UML | Marriott Hotel —
Meeting Room 1 |
| 54 | XP Meets UML: Development Processes for eTechnology | Convention Ctr —
Room 23 |
| 55 | Component-Based Design: A Complete Worked Example | Convention Ctr —
Room 24 |
| 56 | Developing Java Applications for Small Spaces | Convention Ctr —
Room 6 |

Wednesday, 17 October 2001*Half Day – Afternoon – 1:30 pm – 5:00 pm*

57	Patterns for Making Your Business Objects Persistent in a Relational Database World	Convention Ctr — Room 3
58	Creativity in Software Development	Marriott Hotel — Meeting Room 11
59	Architectures for Integrating Business Logic into J2EE	Convention Ctr — Room 22
60	Planning Agile Projects	Convention Ctr — Room 16
61	Designing Small Memory Software: Development Patterns for Systems with Limited Memory	Convention Ctr — Room 25
62	Reflection in Java	Convention Ctr — Room 5
63	Ruby for the Impatient	Marriott Hotel — Meeting Room 12
64	Realizing Extreme Programming as a Strategic Weapon for Innovation	Convention Ctr — Room 23
65	Advanced Extreme Programming Testing Techniques	Convention Ctr — Room 24
66	C++ Idioms	Convention Ctr — Room 13
67	Patterns and Techniques for Developing Performance Effective Enterprise Java Beans	Marriott Hotel — Meeting Room 1
68	Pair Programming: Experience the Difference	Convention Ctr — Room 15
69	Objects vs. The Web	Convention Ctr — Room 14
70	OPEN: A Flexible OO/CBD Process for Software-Intensive Systems Development	Marriott Hotel — Meeting Room 2

TUTORIALS BY TRACK

Fundamentals

- 1 A Brief Tour of Responsibility-Driven Design (Sun. Full Day)
- 2 Testing Object-Oriented Software Systems (Sun. Full Day)
- 4 Concepts of Object-Oriented Programming (Sun. Full Day)
- 8 Introduction to Writing Use Cases (Sun. AM)
- 12 Agile Methodologies (Sun. AM)
- 13 How to Manage the Change from COBOL to OOP (Sun. AM)
- 16 An Introduction to Design Patterns (Sun. AM)

.NET Technologies

- 15 XML, XSD, and SOAP as a Better Component Model (Sun. AM)
- 22 The .NET Framework: The Common Language Runtime and C# (Sun. PM)
- 33 Exposing and Consuming Web Services with .NET (Mon. AM)

Agile Methods and Extreme Programming

- 12 Agile Methodologies (Sun. AM)
- 20 Daily Builds Are for Wimps (Sun. PM)
- 32 Designing an Agile Methodology (Mon. AM)
- 35 Extreme Programming Live! (Mon. AM)
- 37 Refactoring: Improving the Design of Existing Code (Mon. AM)
- 49 No Stone Unturned: An Introduction to Test-First Programming (Tue. PM)
- 54 XP Meets UML: Development Processes for eTechnology (Tue. PM)
- 60 Planning Agile Projects (Wed. PM)
- 64 Realizing Extreme Programming as a Strategic Weapon for Innovation (Wed. PM)
- 65 Advanced Extreme Programming Testing Techniques (Wed. PM)
- 68 Pair Programming: Experience the Difference (Wed. PM)

Architecture

- 6 Inside High-Quality Software Architectures (Sun. AM)
- 14 Component and Service Architecture Modeling with UML (Sun. AM)
- 27 Software Architecture: It's What's Missing From OO Methodologies (Mon. Full Day)
- 29 Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects (Mon. Full Day)
- 34 Efficient Architectures for Object-Oriented Component-Based Middleware (Mon. AM)

TUTORIALS

TUTORIALS BY TRACK

- 36 Patterns and Architectures for J2EE Systems (Mon. AM)
- 42 How to Really Fail at Software Architecture (Mon. PM)
- 44 Architecting Large Business Systems (Tue. Full Day)
- 48 Adaptive Object-Model Architecture: How to Build Systems That Can Dynamically Adapt to New Business Requirements (Tue. PM)
- 50 Designing Software Architecture for Quality: The ADD Method (Tue. PM)
- 51 Creating Responsive, Scalable Systems (Tue. PM)
- 57 Patterns for Making Your Business Objects Persistent in a Relational Database World (Weds. PM)
- 59 Architectures for Integrating Business Logic into J2EE (Wed. PM)
- 67 Patterns and Techniques for Developing Performance Effective Enterprise Java Beans (Wed. PM)
- 69 Objects vs. The Web (Wed. PM)

Components

- 14 Component and Service Architecture Modeling with UML (Sun. AM)
- 22 The .NET Framework: The Common Language Runtime and C# (Sun. PM)
- 25 Fractal Patterns and Frameworks in UML — Towards UML 2.0? (Sun. PM)
- 34 Efficient Architectures for Object-Oriented Component-Based Middleware (Mon. AM)

Concurrency

- 11 Introduction to Concurrent Object-Oriented Programming in Java (Sun. AM)
- 18 Designing Concurrent Object-Oriented Programs in Java (Sun. PM)
- 29 Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects (Mon. Full Day)

Emerging Technologies

- 15 XML, XSD, and SOAP as a Better Component Model (Sun. AM)
- 22 The .NET Framework: The Common Language Runtime and C# (Sun. PM)
- 26 Aspect-Oriented Programming with AspectJ™ (Mon. Full Day)
- 30 Object-Oriented Reengineering (Mon. Full Day)
- 33 Exposing and Consuming Web Services with .NET (Mon. AM)
- 39 Making the Software Process Transparent by Using Intelligent Agents (Mon. PM)
- 63 Ruby for the Impatient (Wed. PM)

Internet Technologies

- 15 XML, XSD, and SOAP as a Better Component Model (Sun. AM)
- 22 The .NET Framework: The Common Language Runtime and C# (Sun. PM)
- 33 Exposing and Consuming Web Services with .NET (Mon. AM)
- 36 Patterns and Architectures for J2EE Systems (Mon. AM)
- 55 Component-Based Design: A Complete Worked Example (Tue. PM)
- 59 Architectures for Integrating Business Logic into J2EE (Wed. PM)
- 69 Objects vs. The Web (Wed. PM)

Java Technologies

- 11 Introduction to Concurrent Object-Oriented Programming in Java (Sun. AM)
- 17 Producing GUIs with Java (Sun. AM)
- 18 Designing Concurrent Object-Oriented Programs in Java (Sun. PM)
- 19 Building Parsers with Java (Sun. PM)
- 26 Aspect-Oriented Programming with AspectJ™ (Mon. Full Day)
- 36 Patterns and Architectures for J2EE Systems (Mon. AM)
- 46 J2ME Design and Development Considerations (Tue. PM)
- 56 Developing Java Applications for Small Spaces (Tue. PM)
- 59 Architectures for Integrating Business Logic into J2EE (Wed. PM)
- 62 Reflection in Java (Wed. PM)
- 67 Patterns and Techniques for Developing Performance Effective Enterprise Java Beans (Wed. PM)

Languages (except Java)

- 15 XML, XSD, and SOAP as a Better Component Model (Sun. AM)
- 23 Garbage Collection (Sun. PM)
- 38 Efficient Implementation of Object-Oriented Programming Languages (Mon. PM)
- 47 Embedded Systems in C++ — C++ Idioms, Patterns, and Architecture for Constrained Systems (Tue. PM)
- 63 Ruby for the Impatient (Wed. PM)
- 66 C++ Idioms (Wed. PM)

Meta-level and Reflective

- 26 Aspect-Oriented Programming with AspectJ™ (Mon. Full Day)
- 48 Adaptive Object-Model Architecture: How to Build Systems That Can Dynamically Adapt to New Business Requirements (Tue. PM)
- 62 Reflection in Java (Wed. PM)

Middleware

- 34 Efficient Architectures for Object-Oriented Component-Based Middleware (Mon. AM)
- 36 Patterns and Architectures for J2EE Systems (Mon. AM)
- 59 Architectures for Integrating Business Logic into J2EE (Wed. PM)
- 67 Patterns and Techniques for Developing Performance Effective Enterprise Java Beans (Wed. PM)

Patterns

- 7 Dungeons and Patterns! (Sun. AM)
- 10 Introducing Patterns (or Any New Idea) into Organizations (Sun. AM)
- 16 An Introduction to Design Patterns (Sun. AM)
- 21 Designing with Patterns (Sun. PM)
- 25 Fractal Patterns and Frameworks in UML — Towards UML 2.0? (Sun. PM)
- 29 Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects (Mon. Full Day)
- 31 Patterns at Work (Mon. AM)
- 36 Patterns and Architectures for J2EE Systems (Mon. AM)
- 41 Refactoring to Patterns (Mon. PM)
- 57 Patterns for Making Your Business Objects Persistent in a Relational Database World (Weds. PM)
- 67 Patterns and Techniques for Developing Performance Effective Enterprise Java Beans (Wed. PM)

People, Process, and Project Management

- 10 Introducing Patterns (or Any New Idea) into Organizations (Sun. AM)
- 39 Making the Software Process Transparent by Using Intelligent Agents (Mon. PM)
- 40 Surviving Object-Oriented Projects (Mon. PM)
- 52 Leading Retrospectives on OO Projects: Looking Back to Move Forward (Tue. PM)
- 58 Creativity in Software Development (Wed. PM)
- 60 Planning Agile Projects (Wed. PM)
- 70 OPEN: A Flexible OO/CBD Process for Software-Intensive Systems Development (Wed. PM)

Refactoring and Reengineering

- 30 Object-Oriented Reengineering (Mon. Full Day)
- 37 Refactoring: Improving the Design of Existing Code (Mon. AM)
- 41 Refactoring to Patterns (Mon. PM)

Requirements Analysis

- 8 Introduction to Writing Use Cases (Sun. AM)
- 24 Advanced Use Case Writing (Sun. PM)
- 28 Improving Your Use Cases (Mon. Full Day)
- 43 The Art of Writing Use Cases (Tue. Full Day)
- 53 Business Modeling with the UML (Tue. PM)

Small or Mobile

- 46 J2ME Design and Development Considerations (Tue. PM)
- 47 Embedded Systems in C++ — C++ Idioms, Patterns, and Architecture for Constrained Systems (Tue. PM)
- 56 Developing Java Applications for Small Spaces (Tue. PM)
- 61 Designing Small Memory Software: Development Patterns for Systems with Limited Memory (Wed. PM)

Testing

- 2 Testing Object-Oriented Software Systems (Sun. AM)
- 20 Daily Builds Are for Wimps (Sun. PM)
- 49 No Stone Unturned: An Introduction to Test-First Programming (Tue. PM)
- 65 Advanced Extreme Programming Testing Techniques (Wed. PM)

UML

- 4 Concepts of Object-Oriented Programming (Sun. Full Day)
- 14 Component and Service Architecture Modeling with UML (Sun. AM)
- 25 Fractal Patterns and Frameworks in UML — Towards UML 2.0? (Sun. PM)
- 45 The UML's Object Constraint Language (OCL) — Specifying Components (Tue. PM)
- 53 Business Modeling with the UML (Tue. PM)
- 54 XP Meets UML: Development Processes for eTechnology (Tue. PM)

Usability and UI

- 3 Usage-Centered Design: An Agile Model-Driven Process for Object-Oriented User Interface Design (Sun. Full Day)
- 5 Lo-Fi Design Strategies for Creating Highly Usable Object-Oriented User Interfaces (Sun. Full Day)
- 9 Object-Oriented Design of Human-Computer Interaction (Sun. AM)
- 17 Producing GUIs with Java (Sun. AM)

1

A Brief Tour of Responsibility-Driven Design

Sunday, Full Day
Convention Ctr — Room 20

Rebecca Wirfs-Brock,
Wirfs-Brock Associates
Alan McKean,
Wirfs-Brock Associates

Responsibility-Driven Design offers practical advice for designing, implementing, and redesigning with responsibilities. In a responsibility-based model, objects play specific roles and occupy well-known positions in the application architecture. Each object is accountable for a specific portion of the work. They collaborate in clearly defined ways, contracting with each other to fulfill the larger goals of the application. By creating a “community of objects,” assigning specific responsibilities to each, we build a collaborative model of our application. Objects are more than simple bundles of logic and data ... they are service-providers, information-holders, structurers, coordinators, controllers, and interfacers to the outside world! Each must know and do its part! Thinking in these terms enables you to build powerful, flexible applications.

This tutorial, which includes new material from our forthcoming book, will be an example-based tour of Responsibility-Driven Design. It presents our latest innovations and practical techniques. Topics include: finding and evaluating the qualities of candidate design objects, mapping roles to classes and interfaces, strategies for assigning object responsibilities, deciding on the control style of an application, effective ways to describe collaborations, how to organize a design by specifying contractual relations and obligations, and techniques for increasing a design’s flexibility and clarity.

Attendee Background: Participants should be familiar with object concepts and be looking for practical techniques, guidelines and a design process that emphasizes modeling the behavioral aspects of a software system.

Presenters: *Rebecca Wirfs-Brock is president of Wirfs-Brock Associates, a firm specializing in the transfer of object analysis and design expertise to organizations and individuals through training, mentoring, and consulting. Rebecca has been involved with object technology since its infancy. She is the inventor of the set of development practices known as Responsibility-Driven Design. From development on the Tektronix implementation of Smalltalk in the early 1980s, through years of development and training experience, she is recognized as one of only a few knowledgeable and influential practitioners of object-oriented design. She spent 17 years as a Software Engineer at Tektronix, where she managed the first commercial Smalltalk effort and was the technical lead for the development of Color Smalltalk. Recently, she has authored use cases for a telecommunications framework and an online banking system and has mentored teams in use case writing, design, architecture and managing incremental, iterative object-technology projects. She practices what she teaches!*

Alan McKean is Vice President and Director of Educational Services at Wirfs-Brock Associates. Alan McKean has devoted most of his career applying principles of design and adult learning to find better ways to communicate technical and design information. A student of R. Buckminster Fuller and a graduate of the University of Oregon with a Masters in Computer Science, he specializes in system architecture and object-oriented design and programming. Alan has delivered over a hundred workshops on designing and programming object-oriented software during his 10+ years at Instantiations, Digitalt, and Wirfs-Brock Associates. Alan was a keynote speaker at the OOPSLA Educator’s Symposium in 1995 and has been invited to speak at this year’s Educators’ Symposium. Prior to his training experience, Alan was a Director at Dynamix, Inc., a computer game company, where he invented and developed a toolset for synchronizing animated images with actors’ recorded voices and a suite of Smalltalk-based tools for managing computer game sound effects and music.

2

Testing Object-Oriented Software Systems

Sunday, Full Day

Convention Ctr — Room 13

John McGregor, *Clemson University*

The use of object-oriented software construction techniques and iterative, incremental processes influence the organization, structure, and execution of testing activities in a project. The techniques presented in the tutorial are intended to provide a scalable process that can be tailored to the size of a project and the criticality of the type of application. The comprehensive test plan, presented in the tutorial, integrates the construction process and the testing process to produce an efficient and complete development process.

This tutorial is divided into three parts:

(1) specific techniques supported by small examples to illustrate specific testing algorithms, (2) techniques for testing system level models using enhanced inspection and review procedures and (3) a process for system testing presented within the context of a complete testing process for object-oriented systems.

Instructional objectives: The participant will be able to define test cases from use cases. The participant will be able to build test suites that reuse test cases from related uses. The participant will be able to adapt a generic testing process to his/her corporate and project environments. The participant will be able to prioritize tests based on information in the use cases.

Lecture/discussion: 70% **Exercises:** 30%

Attendee Background: Attendees should be familiar with object-oriented concepts and at least one object-oriented programming language. It will be helpful if the attendee is familiar with basic software testing techniques to the level gained by practical experience.

Presenter: *Dr. John D. McGregor is a senior partner in Korson-McGregor and an associate professor of computer science at Clemson University. Dr. McGregor has conducted funded research for organizations such as the National Science Foundation, DARPA, IBM, and AT&T. Dr. McGregor has developed testing techniques for object-oriented software and developed custom testing processes for a variety of companies. Dr. McGregor is co-author of Object-oriented Software Development: Engineering Software for Reuse (Van Nostrand Reinhold) and is co-author of A Practical Guide to Testing Object-Oriented Software (Addison-Wesley, 2001). He writes a column on Testing and Quality for the Journal of Object-oriented Programming (JOOP) published by SIGS Publishing. He has published numerous articles on software development focusing on design and quality issues. Dr. McGregor's research interests include software engineering specifically in the areas of process definition, design quality, testing and measurement. Dr. McGregor has given tutorials for several years at OOPSLA and ECOOP. He presents to 10 - 12 conferences per year as well as offering industrial courses to demanding technical audiences.*

3

Usage-Centered Design: An Agile Model-Driven Process for Object-Oriented User Interface Design

*Sunday, Full Day
Convention Ctr — Room 25*

Larry Constantine,
*University of Technology, Sydney;
Constantine & Lockwood, Ltd.*

James Noble,
Victoria University of Wellington

Agile processes and lightweight methods are increasingly popular but share with their ponderous predecessors an inattention to usability and user interface design. Usage-centered design is a proven industrial-strength process for designing highly usable and innovative solutions to interaction-intensive problems. It has been applied with marked success to projects ranging from automation programming tools to classroom information systems to e-commerce Web sites. Through alternating lectures, applied exercises, and discussions, this tutorial introduces a streamlined process for quickly and efficiently designing improved user interfaces supported by robust internal software. Essential use cases—a simplified, generalized, and abstract improvement on conventional use cases—are used to model tasks and to guide the design of user interfaces that effectively support the real needs of users. Through actual application to a compressed but representative case study problem, participants will learn, how to employ ordinary index cards and accelerated modeling sessions to quickly understand and prioritize user roles and user tasks and to organize the needed user interface contents. The emphasis will be on modeling techniques that yield the greatest payoff from the least effort in design, techniques that are well-suited to producing world-class designs through iterative, time-boxed development within compressed release cycles.

Attendee Background: Some experience with use cases and familiarity with the basic concepts and techniques of object-orientation are assumed.

Understanding of the basic principles of usability and user interface design would be helpful but is not mandatory.

Presenters: *Larry Constantine is Adjunct Professor of Computing Sciences, University of Technology, Sydney, and Director of Research and Development for Constantine & Lockwood, Ltd., the international design and consulting firm he co-founded. A pioneer of modern software engineering practice and a recognized authority on the human side of software, he is the co-inventor of essential use cases and usage-centered design. He has conducted hundreds of seminars and tutorials in nineteen countries and his publications include sixteen books and nearly 150 papers.*

Dr. James Noble is a lecturer at the Victoria University of Wellington, New Zealand, and a Consulting Associate with Constantine & Lockwood, Ltd. He is the co-author of Small Memory Software: Patterns for Systems with Limited Memory (Addison-Wesley 2000), and numerous published papers on software design, user interface design, and design patterns. He has extensive lecturing and teaching experience, including tutorials at OOPSLA, TOOLS Pacific, and OzCHI.

4

Concepts of Object-Oriented Programming

Sunday, Full Day
Convention Ctr — Room 22

Raimund Ege,
Florida International University

This tutorial defines and teaches the basic object-oriented concepts, illustrates their advantages, and introduces the components and features of object-oriented programming languages and development environments. The tutorial enables an attendee to make an informed decision about what language/environment will best serve his/her software development needs. The tutorial has two major parts: Part 1 discusses in detail all object-oriented concepts and uses UML and Java to illustrate them. The focus will be on a precise non-confusing definition of the core concepts and terminology, such as object, instance, class, interface, attribute, service, message passing, hierarchy, inheritance, polymorphism, late binding, memory management, access specification, and packaging. Part 2 then compares the major object-oriented programming languages: C++, Java, Smalltalk, and others. The comparison is done with a double focus: (1) how does the language support and enforce the concepts, and (2) how does the language help software development (to that effect, I have a small case study program, that will be solved in all languages). Whether and how each language supports advanced concepts, like multiple and repeated inheritance, genericity, interfaces, is discussed in detail.

Attendee Background: Attendees are software professionals who are interested in learning the fundamental concepts and advantages of object-oriented programming and how to apply them in a modern software development environment. No previous knowledge of object-oriented concepts is assumed. The attendees should have a fundamental background in computer science and/or computer programming.

Presenter: Raimund K. Ege is an Associate Professor of Computer Science at the Florida International University, Miami. He is author of *Programming in an Object-Oriented Environment* (Academic Press, 1992), and *Object-Oriented Programming with C++* (Academic Press, 1994). He is an active researcher in the area of object-oriented concepts, and their application to programming, user interfaces, databases, simulation, and software engineering. He has presented numerous successful tutorials at major conferences (OOPSLA, ECOOP, TOOLS). The tutorials were consistently rated highest and won praise from organizers and attendees.

5

Lo-Fi Design Strategies for Creating Highly Usable Object-Oriented User Interfaces*Sunday, Full Day**Convention Ctr — Room 24**Luke Hohmann, Independent Consultant*

Even if a software development project creates extensive and complete object-oriented analysis and design models, it will still be perceived as a failure if the user interface is poorly constructed. To be perceived as truly successful, the system must meet the needs of the user. This is best done by designing a user interface that is effective, appealing, intuitive, and easy to learn. In other words, you must create a highly usable object-oriented user interface. Participants of this tutorial will learn how to design highly usable object-oriented user interfaces using the latest in lo-fi prototyping techniques by creating such designs in small groups. Upon completion of this tutorial, participants will be able to: define usability and its relation to object technology; define the role of lo-fi and hi-fi prototyping; design and test lo-fi prototypes according to timeless principles of usability; and implement lo-fi prototypes in a manner that is consistent with the underlying domain model.

Attendee Background: Participants should have a basic knowledge of object-oriented analysis and design, use cases, and scenarios; and be involved in the design and implementation of a project utilizing a graphical user interface. Knowledge of a specific object-oriented programming language is not required.

Presenter: *Luke Hohmann is an independent consultant, committed to coaching his clients to greater levels of performance. Mr. Hohmann is author of Journey of the Software Professional: A Sociology of Software Development (Prentice-Hall), as well as numerous articles on software engineering management. A skilled instructor and speaker, Mr. Hohmann has been invited to many conferences. Mr. Hohmann can be contacted through e-mail at LukeHohmann@yahoo.com.*

6

Inside High-Quality Software Architectures

Sunday, Morning

Convention Ctr — Room 18

Frank Buschmann,
Siemens AG, Germany

In this tutorial we examine the secrets of high-quality software architectures: how are they specified, what are their properties, and how are they implemented. The result is a set of principles consisting of methodological steps, concrete design goals that help constructing and implementing software architectures successfully as well as a set of properties that such architectures expose. To illustrate these steps, goals, and properties we use a running example from the industrial automation domain.

Attendee Background: Sound knowledge in Object Technology.

Presenter: *Frank Buschmann is a software engineer at Siemens Corporate Technology in Munich, Germany. His research interests include Object Technology, Application Frameworks, and specifically Patterns. Frank has been involved in several concrete industrial software development projects. Frank is co-author of Pattern-Oriented Software Architecture — A System of Patterns.*

7

Dungeons and Patterns!

Sunday, Morning

Marriott Hotel — Meeting Room 11

Steve Metsker, *Capital One*

William Wake, *Capital One*

“Dungeons and Patterns” is a hands-on tutorial for exploring and learning about design patterns. Learning design patterns will help you become a more powerful object-oriented developer. Unfortunately, a single reading of *Design Patterns* won’t magically implant design pattern recognition skills in your brain. You have to learn patterns by doing, which means you have to start applying patterns before you can apply them—a monstrous dilemma! The solution is to practice patterns in a playful setting where slip-ups are profitable and painless. In this tutorial you will join an adventure with others at your table, seeking the treasure of patterns hidden within a dungeon replete with structural traps, motivational pitfalls, and implementation monsters. These barriers will succumb to strong collaboration with your table-mates and will yield to effective application of the patterns in *Design Patterns*. Dungeons and Patterns will deepen your understanding and strengthen your skills at recognizing and applying design patterns.

Attendee Background: Attendees should have tried reading *Design Patterns* at least once. No experience with role-playing games is required.

Presenters: Steve Metsker is a researcher and author who explores and writes about ways to expand the abilities of developers. Steve’s articles have explained how to maintain relational integrity in object models, how to solve logic puzzles in Java, and how the conception of “object” differs between Plato and the OO languages. Steve’s most recent publication is the book, *Building Parsers with Java*.

William Wake is interested in XP, patterns, human-computer interaction, and information retrieval. He is the author of, *Extreme Programming Explored*, and the inventor of the *Test-First Stoplight* and the *XP Programmer’s Cube*.

8

Introduction to Writing Use Cases

Sunday, Morning

Marriott Hotel — Florida Salon V

Alistair Cockburn,

Humans and Technology

A use case is a way of describing the required behavior of a system, centered around what the system offers its users. Use cases are easy to read, and simple in concept, but surprisingly tricky to write. This tutorial is for the beginning use case writer, to see what a use case looks like, the basics of how to write one, and how to organize people to write, review and use them. The tutorial will be part lecture, and part hands-on exercises. Attendees will brainstorm a list of use cases for a system, write a main scenario, and uncover failure scenarios. The exercises are designed to allow the attendees to practice the writing skills, and discover where use cases get difficult. At the end of the tutorial, the attendee will have the basic vocabulary of use cases, will have seen examples of good and bad ones, and will have experienced the variations in writing that will show up in real use cases.

Attendee Background: This tutorial is for people just beginning to write or consider use cases. No particular background is required.

Presenter: *Alistair Cockburn is a highly regarded instructor and is known as one of the premier experts on use cases. His book, Writing Effective Use Cases, set the standard in the area and was nominated for Software Development's Jolt book award in 2001. Alistair has taught use case writing since 1994, and has also acted as consultant on project management, object-oriented design, and methodology to the Central Bank of Norway, the IBM Consulting Group, and the First Rand Bank of South Africa. Materials that support his workshops can be found at <http://members.aol.com/acockburn>, <http://crystalmethodologies.org> and <http://usecases.org>.*

9

Object-Oriented Design of Human-Computer Interaction

Sunday, Morning

Convention Ctr — Room 15

Mary Beth Rosson, Virginia Tech

Object-oriented design methods are claimed to reduce the gap between the problem domain and the software system. This has important implications for the design of human-computer interaction: A software model that mirrors the real world should reduce the cognitive distance between how a system works and the mental models that users build to use and understand the software. This tutorial explores how to apply object-oriented thinking to the design of human-computer interaction. The methods discussed are part of a general scenario-based framework for usability engineering. In this framework, a scenario is a narrative of the goals, actions, and reactions of actors pursuing goals with an interactive system. The tutorial presents scenario-based techniques for developing and integrating object-oriented views of requirements, activity design, user interface design, and usability evaluation. Throughout, design rationale is captured, serving to raise and discuss the implications that object-oriented concepts will have for the user experience. Concepts and techniques are introduced briefly, then illustrated with examples. The format will be lecture interspersed with presentation and discussion of the examples.

Attendee Background: General knowledge of object-oriented concepts, interest in use-centered design of interactive systems.

Presenter: *Mary Beth Rosson is an associate professor of computer science at Virginia Tech. She is an expert in human-computer interaction (HCI), and the author of numerous research papers on the relationship between HCI and object-oriented design. Rosson has given research papers and tutorials at the ACM SIGCHI, OOPSLA, and ECOOP conferences and has served in many leadership positions in SIGCHI and SIGPLAN. Most recently, she was General Chair of OOPSLA 2000.*

10

Introducing Patterns (or Any New Idea) into Organizations

Sunday, Morning

Convention Ctr — Room 14

Mary Lynn Manns,
University of North Carolina at Asheville

Linda Rising, *Independent Consultant*

Many people who have attended OOPSLA or other conferences find new ideas that they wish to take back to their organizations, but then struggle to make something happen. This tutorial will help participants understand what successful change agents have learned while attempting to introduce new ideas into their organizations. The lessons learned have been documented in an evolving pattern language titled *Introducing Patterns (or any new idea) into Organizations*. This session will examine the problems and solutions documented in this language through the simulation of attempts to introduce a new idea, such as patterns, into an organization.

Attendee Background: Anyone in the software business who is trying to introduce patterns (or any new idea) into an organization will find this tutorial useful. We assume that attendees are familiar with the notion of patterns.

Presenters: *Mary Lynn Manns is on the faculty at the University of North Carolina at Asheville. During the past three years, she has studied the issues in introducing patterns into organizations. She has also taught patterns in industry and done numerous other presentations on the topic.*

Linda Rising has a Ph.D. from Arizona State University in the area of object-based design metrics. Her background includes university teaching experience as well as work in industry in the areas of telecommunications, avionics, and strategic weapons systems. She has been working with object technologies since 1983. She is the editor of A Patterns Handbook, The Pattern Almanac 2000, and Design Patterns in Communication Software.

11

Introduction to Concurrent Object-Oriented Programming in Java

Sunday, Morning

Convention Ctr — Room 16

David Holmes, *DSTC Pty Ltd.*

Doug Lea,
*State University of New York
(SUNY) at Oswego*

Concurrent programming has mostly been the domain of systems programmers rather than application developers, but Java's support of concurrency has enticed many to try their hand at concurrent applications. However concurrent programming poses many traps for the unwary. This tutorial demonstrates various design patterns and techniques for constructing concurrent applications in Java and for managing that concurrency. On the language side we look at Java's mechanisms to support concurrent programming. On the design side we look at object structures and design rules that can successfully resolve the competing forces (safety, liveness, efficiency, coordination, reusability) present in concurrent software design problems. Participants will acquire comprehensive knowledge of the concurrency support provided by the Java language and core classes, as well as insight into some threading issues within the Java libraries. They will be exposed to a range of design approaches to assist them in developing safe, concurrent applications in Java and other object-oriented languages.

Attendee Background: This tutorial targets anyone involved, or planning to get involved, in the development of concurrent object-oriented applications. It is assumed that the attendee is familiar with basic OO concepts and has a working knowledge of the Java programming language.

Presenters: *David Holmes is a Senior Research Scientist at the Cooperative Research Centre for Enterprise Distributed Systems Technology (DSTC Pty, Ltd.), in Brisbane, Australia. He completed his Ph.D. in the area of synchronization within object-oriented systems and has been involved in concurrent programming for a number of years. He is a co-author of the third edition of the Java Series book, The Java Programming Language.*

Doug Lea is a professor of Computer Science at the State University of New York at Oswego. He is author of the Java Series book, Concurrent Programming in Java: Design Principles and Patterns, co-author of the book, Object-Oriented System Development, and the author of several widely used software packages, as well as articles and reports on object-oriented software development.

12

Agile Methodologies

Sunday, Morning
Convention Ctr — Room 19

Jim Highsmith,
Information Architects, Inc.

In the past two years, a wide range of publications (Software Development, IEEE Software, Cutter IT Journal, Software Testing and Quality Engineering, and even the Economist) have published articles on what Martin Fowler calls the New Methodologies. There has been a rapidly rising interest in these new approaches to software development such as Extreme Programming, Scrum, Adaptive Software Development, Feature-Driven Development, and Dynamic Systems Development Methodology. Furthermore, scores of organizations have developed their own “lighter” approach to building software. Recently, representatives from each of the New Methodologies met, formed the Agile Alliance, and developed common purpose and principles to help others think about software development, methodologies, and organizations, in new “more agile” ways. This workshop, given by Jim Highsmith, developer of one of the Agile Methodologies (Adaptive Software Development), and one of the authors of the *Manifesto for Agile Software Development*, addresses key questions: What are Agile Methodologies? What problem domains do Agile Methodologies address? What are the common principles behind Agile Methodologies? What are the similarities and differences between the various Agile Methodologies?

Attendee Background: The tutorial is targeted at software development managers, project managers, and team leaders. Basic project management knowledge will be helpful.

Presenter: *Jim Highsmith is director of Cutter Consortium's e-Project Management Practice, president of Information Architects, Inc., and author of Adaptive Software Development: A Collaborative Approach to Managing Complex Systems (Dorset House, 2000). He has 30 years experience as a consultant, software developer, manager, and writer. Jim has published dozens of articles in major industry publications and his ideas about project management in the Internet era were featured in recent issues of ComputerWorld and the Economic Times in India. In the last ten years, he has worked with both IT organizations and software companies in the US, Europe, Canada, South Africa, Australia, Japan, India, and New Zealand to help them adapt to the accelerated pace of development in increasingly complex, uncertain environments.*

13

How to Manage the Change from COBOL to OOP

Sunday, Morning

Marriott Hotel — Salon A

Markus Knasmüller, *BMD Systemhaus*

After solving the problems Y2K and Euro, the last big challenges for Cobol-programmers are over. Therefore most of them have to look for new fields of activity, but these are combined with new programming techniques like object-oriented programming. However, introducing object-oriented programming to old-style programmers is a rather hard task. This tutorial shows how this job was done at BMD Steyr, Austrians leading producer of accountancy software. It is a perfect support for everybody who wants to introduce or teach object-oriented programming. After presenting background information why one should change and how this change should be accompanied, a special course for former Cobol programmers is presented. Experiences, as well as tips and tricks, will round up the presentation.

Attendee Background: The participants should have basic knowledge of traditional programming languages like Cobol or PL/I and should have the wish to change to object-oriented programming.

Presenter: *Markus Knasmüller is head of the software department at BMD Systemhaus, Austrians leading producer of accountancy software. In this position he was responsible for the change of 50 programmers and 5 millions lines of code from COBOL to OOP. He is author of various research papers and books (for example: From COBOL to OOP, dpunkt, 2001) and has experience in teaching object-oriented programming at the university as well as in industry. Markus holds a Ph.D. in computer science and a degree in management information systems.*

14

Component and Service Architecture Modeling with UML

Sunday, Morning

Convention Ctr — Room 21

Desmond D'Souza, Kinetium

The word “architecture” often bestows instant importance to pretty powerpoint drawings and vague hand waving. And while e-business demands flexible configuration of components and web-services, those components and services will only plug together if they conform to shared “pluggable” architecture standards. Believing “architecture keeps designers from needless creativity,” we outline a clear definition of architecture and architectural style based on UML packages, patterns, and refinement. Elements and rules of an architectural style are separated from the designs which use them, and component architectures use an abstract component-connector model.

Attendee Background: Attendees should be familiar with the UML.

Presenter: *Desmond D'Souza is founder and president of Kinetium. He is co-author of Objects, Components, and Frameworks With UML: The Catalysis Approach (Addison Wesley 1998), and a respected speaker internationally. He was previously senior vice president of component-based development at Platinum Technology and at Computer Associates. Kinetium provides client solutions that leverage shareable architectures for model-driven development and integration of systems, with a current focus on light-weight modeling architecture and methods. Desmond can be reached at dsouzad@acm.org.*

15

XML, XSD, and SOAP as a Better Component Model

Sunday, Morning

Marriott Hotel — Florida Salon VI

Don Box, *DevelopMentor*

The Simple Object Access Protocol (SOAP) is an XML-based protocol for exposing servers, services, components or objects over the web. SOAP codifies the use of existing technologies such as XML, XML Schema Definition (XSD) language, and HTTP to allow code to be accessed in an interoperable and Internet-friendly fashion. This tutorial covers the following topics: The XML Protocol Stack, the XML Schema Language, HTTP Myths vs. Reality, SOAP Encoding, SOAP Framing, SOAP and Extensibility, Architecture of a SOAP runtime, and Architecture of a SOAP application

Attendee Background: Attendees should be familiar with the basics of object-oriented programming and moderately comfortable with some RPC or messaging based technology such as CORBA, DCOM or RMI.

Presenter: *Don Box is a co-founder of DevelopMentor, a developer services company that provides education and support to the software industry at large. Don's research interests include component software integration, programming for concurrency, and XML-based serialization and metadata protocols. Don is a series editor at Addison Wesley and is the author of Essential COM, and a co-author of Effective COM, and Essential XML, all from Addison Wesley. Don is a contributing editor and columnist at Microsoft Systems Journal (now called MSDN Magazine) and an occasional contributor to XML.com. Don is also a co-author of the Simple Object Access Protocol specification and a member of the W3C Schemas Working Group. Don has a Master's Degree in Computer Science from the University of California at Irvine.*

16

An Introduction to Design Patterns

Sunday, Morning

Marriott Hotel — Florida Salon IV

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

Designing object-oriented software is hard, and designing reusable object-oriented software is even harder. Experience shows that many object-oriented systems exhibit recurring structures or “design patterns” of communicating and collaborating objects that promote extensibility, flexibility, and reusability. This tutorial describes a set of fundamental design patterns and, through a design scenario, demonstrates how to build reusable object-oriented software with them. The tutorial covers the roles design patterns play in the object-oriented development process: how they provide a common vocabulary, reduce system complexity, and how they act as reusable architectural elements that contribute to an overall system architecture.

Attendee Background: Attendees should understand basic object-oriented concepts, like polymorphism and type versus interface inheritance, and should have had some experience designing object-oriented systems. No prior knowledge of design patterns is required. Familiarity with Java is recommended.

Presenter: *John Vlissides is a member of the research staff at the IBM T.J. Watson Research Center in Hawthorne, NY. He has practiced object-oriented technology for over a decade as a designer, implementer, researcher, lecturer, and consultant. John is author of Pattern Hatching, co-author of Design Patterns and Object-Oriented Application Frameworks, and co-editor of Pattern Languages of Program Design 2. He has published many articles and technical papers on object-oriented themes in general and design patterns in particular. John is a columnist for Java Report and serves as Consulting Editor of Addison-Wesley's Software Patterns Series. He has a Ph.D. in Electrical Engineering from Stanford University.*

17

Producing GUIs with Java

Sunday, Morning

Marriott Hotel — Meeting Room 12

Fintan Culwin,

South Bank University: London

The Java Foundation Classes supply a collection of user interface components. This tutorial attempts to introduce a representative selection of the most common and useful of them, showing how they can be combined to produce effective user interfaces. To accomplish this efficiently it is necessary to start with a representation of the required behavior of the interface and derive the detailed design from it.

Objectives:

- provide an introductory overview of the widgets supplied by the JFC;
- introduce the usability heuristics and style guides that can be employed in the detailed design of user interfaces;
- show how State Transition Diagrams (STDs) can describe the required behavior of an interface;
- introduce the Java event dispatch/listener model;
- introduce and illustrate layout management policies;
- illustrate the use of STD, Class, Instance, Interface Layout and Object Interaction diagram notations;
- illustrate the realization of detailed three-layer designs in Java;
- introduce the resource management techniques, which improve the presentation of an interface.

Attendee Background: An intermediate level tutorial for attendees who have an initial familiarity with OO concepts and wish to develop further understanding in the context of GUI construction. Most of the exposition is at the source code level.

Presenter: *Fintan Culwin is a Reader in Software Engineering Education at South Bank University: London specializing in Software Engineering and HCI, particularly in the integration of usability considerations in the earliest stages of production processes. He has published five books, including two on Java, and is currently completing a sixth on the JFC. He has published extensively on Internet issues and has presented sessions on the Web and Java at a series of international conferences including: SIGCSE, BCS HCI, ITiCSE, CHI and OOPSLA.*

18

Designing Concurrent Object-Oriented Programs in Java

Sunday, Afternoon

Convention Ctr — Room 16

Doug Lea,

State University of New York
(SUNY) at Oswego

David Holmes, DSTC Pty Ltd.

Concurrent programming has mostly been the domain of systems programmers rather than application developers, but Java's support of concurrency has enticed many to try their hand at concurrent applications. Effectively creating and managing concurrency within an application poses many design choices and trade-offs. This tutorial looks at more advanced issues in designing concurrent applications. It describes mechanisms for introducing concurrency into applications (threads, message-passing, asynchronous calls) and different models for application architectures, such as data-flow and event-driven designs. The tutorial also shows how concurrency controls can be abstracted into reusable support classes, and finally discusses how concurrent components and applications should be documented. Participants will learn how concurrent applications can be structured in different ways and how different mechanisms can be used to effect concurrent behavior. They will be exposed to a range of design patterns and techniques for introducing and managing concurrency within their applications and how to create reusable concurrency abstractions.

Attendee Background: This tutorial targets anyone involved, or planning to get involved, in the development of concurrent object-oriented applications. It is expected that the attendee is very familiar with OO concepts and the Java language, and has a good working knowledge of Java's concurrency mechanisms.

Presenters: *Doug Lea is a professor of Computer Science at the State University of New York at Oswego. He is author of the Java Series book, Concurrent Programming in Java: Design Principles and Patterns, co-author of the book, Object-Oriented System Development, and the author of several widely used software packages, as well as articles and reports on object-oriented software development.*

David Holmes is a Senior Research Scientist at the Cooperative Research Centre for Enterprise Distributed Systems Technology (DSTC Pty, Ltd.), in Brisbane, Australia. He completed his Ph.D. in the area of synchronization within object-oriented systems and has been involved in concurrent programming for a number of years. He is a co-author of the third edition of the Java Series book, The Java Programming Language.

19 Building Parsers with Java

Sunday, Afternoon

Marriott Hotel — Meeting Room 11

Steve Metsker, *Capital One*

By learning to write parsers you learn to bridge the gap between computers and the users of your language. You can nestle a new language into any niche, defining how your users interact with computers using text. This workshop introduces the basics of building a language from Sequence, Alternation, and Repetition objects. With these three objects, you can create any syntax-free language. In this session you will spend a large portion of class time writing parsers, using the tools this workshop introduces. You will learn when to create an XML-based language and when to use Java. You will also learn how to design a language and how to generate a working parser from this design.

Attendee Background: Attendees should be experienced Java developers.

Presenter: *Steve Metsker is a researcher and author who explores and writes about ways to expand the abilities of developers. Steve's articles have explained how to maintain relational integrity in object models, how to solve logic puzzles in Java, and how the conception of "object" differs between Plato and the OO languages. Steve's most recent publication is the book, Building Parsers with Java.*

20 Daily Builds Are for Wimps

Sunday, Afternoon

Convention Ctr — Room 15

Michael Two, *Thoughtworks*

Over the last couple of years we've been building a large J2EE application. One of the biggest lessons we've learned is to follow the Extreme Programming (XP) approach to Continuous Integration of our 250+ KLOC system. This session focuses on how we turned a project that needed days of fiddling around to get a build into a project that delivers a fully tested build every hour. We'll go through our automated testing process using JUnit and Excel based acceptance test driver. We will also talk about a set of open source tools we have developed to automate build processes using Ant. We will explore the code for the tools that connect to the source code control system, run code generators, compile, deploy, test and publish the build.

Attendee Background: Participants should be familiar with Java and basic XML syntax.

Presenter: *Michael is a developer and XP advocate at ThoughtWorks working on a very large J2EE application using XP. After studying physics in college he chose a career in software once he realized that staying up all night in an office is more fun than staying up all night in a lab. Michael wrote labor schedule optimization software in C++ before joining Thoughtworks in 1999.*

21

Designing with Patterns

Sunday, Afternoon

Marriott Hotel — Florida Salon IV

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

Design patterns are making the transition from curiosity to familiarity. Now that many people know *what* they are, they want to know *how* best to apply them. This tutorial shows how to leverage patterns in the software design process. It reveals the thinking behind pattern application—including when *not* to use a seemingly applicable pattern. It shows how the right patterns can improve a design and how the wrong patterns can degrade one. Students thus learn to apply design patterns to maximum benefit.

Attendee Background: Attendees should be well-grounded in object technology and should be familiar with the design patterns in *Design Patterns: Elements of Reusable Object-Oriented Software*, by Gamma, et al. Familiarity with Java is recommended.

Presenter: *John Vlissides is a member of the research staff at the IBM T.J. Watson Research Center in Hawthorne, NY. He has practiced object-oriented technology for over a decade as a designer, implementer, researcher, lecturer, and consultant. John is author of Pattern Hatching, co-author of Design Patterns and Object-Oriented Application Frameworks, and co-editor of Pattern Languages of Program Design 2. He has published many articles and technical papers on object-oriented themes in general and design patterns in particular. John is a columnist for Java Report and serves as Consulting Editor of Addison-Wesley's Software Patterns Series. He has a Ph.D. in Electrical Engineering from Stanford University.*

22

**The .NET Framework:
The Common Language Runtime and C#***Sunday, Afternoon**Marriott Hotel — Florida Salon VI**Don Box, DevelopMentor*

The Common Language Runtime is a new implementation of many existing ideas in component technology. The CLR is a type-centric multi-paradigm component model and runtime that supports object-oriented programming, interface-based programming, and aspect-oriented programming. This tutorial covers the following topics: Managed Types Vs. Unmanaged Types, Managed Execution Vs. Unmanaged Execution, Programming Language vs. The Runtime, The CLR Type System, Loading and Linking, Runtime Type Management, Context and Remoting, and Web Services

Attendee Background: Attendees should be familiar with the basics of object-oriented programming and moderately comfortable with systems-programming issues such as thread and process management.

Presenter: *Don Box is a co-founder of DevelopMentor, a developer services company that provides education and support to the software industry at large. Don's research interests include component software integration, programming for concurrency, and XML-based serialization and metadata protocols. Don is a series editor at Addison Wesley and is the author of Essential COM, and a co-author of Effective COM, and Essential XML, all from Addison Wesley. Don is a contributing editor and columnist at Microsoft Systems Journal (now called MSDN Magazine) and an occasional contributor to XML.com. Don is also a co-author of the Simple Object Access Protocol specification and a member of the W3C Schemas Working Group. Don has a Master's Degree in Computer Science from the University of California at Irvine.*

23

Garbage Collection

Sunday, Afternoon

Marriott Hotel — Meeting Room 12

Richard Jones, *University of Kent*

Eric Jul, *University of Copenhagen*

This tutorial presents the issues facing modern high performance garbage collectors and examines the approaches taken by state of the art garbage collectors. Participants will gain a deeper insight into the operation of modern, high performance garbage collectors. The tutorial will enable participants to evaluate the benefits and costs of such garbage collection algorithms, to understand the implications for their code and to make informed choices between collectors.

Attendee Background: Participants will be experienced programmers familiar with basic garbage collection technology (for example having attended the introductory GC tutorial — although there would be some overlap). Basic knowledge of OO implementation would be useful but not essential.

Presenters: *Richard Jones is a Senior Lecturer and Deputy Director of the Computing Laboratory at the University of Kent. He is the prime author of the book on Garbage Collection. His interests include programming languages and their implementation, storage management and distributed systems. He is a member of the Steering Committee of the International Symposium on Memory Management and was Programme Chair for ISMM'98. He has presented several tutorials at OOPSLA and ECOOP.*

Eric Jul is an Associate Professor and Head of the distributed systems group at DIKU, the Dept. of Computer Science, University of Copenhagen. He is co-designer and principal implementer of the Emerald distributed object-oriented programming language. His interests include distributed, OO languages, operating systems support including distributed storage management and object-oriented design and analysis. He was Programme Chair for ECOOP'98. He has presented tutorials regularly at OOPSLA and ECOOP.

24

Advanced Use Case Writing

Sunday, Afternoon

Marriott Hotel — Florida Salon V

Alistair Cockburn,

Humans and Technology

This tutorial is a chance for practiced use case writers to gather and ask the hard questions. What is the difference between Extends and Includes? What is the difference between a business use case and a system use case? When do we stop drawing pictures and start writing text? What expertise is required in the use case writing team? How do we control the mixed levels of writing across people? Where do I put the UI design, the data descriptions and all the other requirements? Can non-technical people write the use cases? Who reviews the use cases? How do we keep use cases writers from infringing on design? The tutorial is structured as part lecture, part workshop, and part open question-and-answer. The lecture introduces the new Stakeholders & Interests model of use cases, along with the notions of different design scopes and goal levels. The workshop portion gives the attendees a chance to try their hands at resolving certain kinds of frequently occurring problems, to sharpen their skills. The open question-and-answer section allows the attendees to ask questions currently plaguing them at work, and even trade answers and experiences amongst themselves.

Attendee Background: Attendees must have written some use cases and be familiar with basic use case concepts.

Presenter: *Alistair Cockburn is a highly regarded instructor and is known as one of the premier experts on use cases. His book, Writing Effective Use Cases, set the standard in the area and was nominated for Software Development's Jolt book award in 2001. Alistair has taught use case writing since 1994, and has also acted as consultant on project management, object-oriented design, and methodology to the Central Bank of Norway, the IBM Consulting Group, and the First Rand Bank of South Africa. Materials that support his workshops can be found at <http://members.aol.com/acockburn>, <http://crystalmethodologies.org> and <http://usecases.org>.*

25

Fractal Patterns and Frameworks in UML — Towards UML 2.0?

Sunday, Afternoon

Convention Ctr — Room 21

Desmond D'Souza, Kinetium

The UML can be used in a simple and consistent way to (a) use a “plug-in” framework approach from business and requirements patterns, through architecture and design patterns, to code, (b) treat “objects” and “use-cases” in a fractal manner, from business to code, with patterns of refinement, (c) specify and design components using “types” and “collaborations,” and (d) define component architectures based on an extensible “kit” of architectural modeling elements. This tutorial shows how UML pattern models can be used in a fractal approach to modeling and design.

Attendee Background: Attendees should be familiar with the UML.

Presenter: *Desmond D'Souza is founder and president of Kinetium. He is co-author of the CATALYSIS Method (Addison Wesley, 1998), and a respected speaker internationally. He was previously senior vice president of component-based development at Platinum Technology and at Computer Associates. Kinetium provides client solutions that leverage shareable architectures for model-driven development and integration of systems, with a current focus on light-weight modeling architecture and methods. Desmond can be reached at dsouzad@acm.org.*

26

Aspect-Oriented Programming with AspectJ™

Monday, Full Day
Convention Ctr — Room 20

Gregor Kiczales,
Xerox PARC,
University of British Columbia
Erik Hilsdale, Xerox PARC

Aspect-oriented programming (AOP) is a technique for improving separation of concerns in software design and implementation. AOP works by providing explicit mechanisms for capturing the structure of crosscutting concerns.

AspectJ is a seamless aspect-oriented extension to Java™. It can be used to cleanly modularize the crosscutting structure of concerns such as exception handling, multi-object protocols, synchronization, performance optimizations, and resource sharing.

When implemented in a non-aspect-oriented fashion, the code for these concerns typically becomes spread out across entire programs. AspectJ controls such code-tangling and makes the underlying concerns more apparent, making programs easier to develop and maintain. This tutorial will introduce aspect-oriented programming and show how to use AspectJ to implement crosscutting concerns in a concise, modular way. We will use numerous examples to develop participants' understanding of aspect-oriented programming through AspectJ. We will also demonstrate AspectJ's integration with IDEs such as JBuilder 4.0 and Forte4J, and emacs. AspectJ is freely available at <http://www.aspectj.org>

Attendee Background: Attendees should have experience doing object-oriented design and implementation, and should be able to read Java code. No prior experience with aspect-oriented programming or AspectJ is required.

Presenters: *Gregor Kiczales is Professor of Computer Science and Xerox/Sierra Systems/NSERC Chair of Software Design at the University of British Columbia. He is also a Principal Scientist at the Xerox Palo Alto Research Center, where he leads the group that has developed aspect-oriented programming and AspectJ. The focus of his research is enabling programmers to write programs that, as much as possible, look like their design. Prior to developing aspect-oriented programming he worked on open implementation, metaobject protocols, and the CLOS object-oriented programming language. He is co-author of The Art of the Metaobject Protocol, a key work in computational reflection. He has given numerous invited talks, lectures, and tutorials at conferences, universities, and in industry.*

Erik Hilsdale is a member of the research staff at Xerox's Palo Alto Research Center. As a member of the AspectJ project team, he focuses on language design, pedagogy, and compiler implementation. He has written several conference and workshop publications in programming languages. He is an experienced and energetic instructor in programming languages with a long background with AspectJ.

27

Software Architecture: It's What's Missing From OO Methodologies

Monday, Full Day
Convention Ctr — Room 13

Jim Doble, *Tavve Software Company*
Gerard Meszaros, *Clearstream Consulting*
Ron Crocker, *Motorola, Inc.*

Explore the challenges associated with the development of large-scale, real-life, proprietary, object-oriented, distributed, embedded, and multi-tier software systems, and discuss the path of a new professional discipline: the software architect. The software architect needs to be able to organize software systems, and make strategic design decisions, to achieve business goals related to system availability, security, scalability, survivability, long-lived flexibility, large-scale granularity, data quality and maintenance, system metrics and reports, packaging and delivery mechanisms. These issues are not commonly explored via UML or other popular modeling approaches, but are nevertheless critical to the success of modern software development projects. The teaching style for this tutorial is case-driven, and hands-on in nature. Attendees will be divided into teams to work on architecture problems. Throughout the day, working sessions will be intermixed with instructor lectures, to achieve a participatory learning experience. The goal is that attendees will learn both from the instructors and from each other. As a result, the attendee is assumed to have experience building at least one real-life software system of substantial size.

Attendee Background: Attendees should either be currently working as software architects, trying to establish a software architecture practice within their company, or working on software systems where they believe an increased emphasis on architecture is needed. Attendees should have experience building at least one real-world software system of substantial size.

Presenters: *Jim Doble has worked as software developer, manager, and architect within the telecommunications industry for over 19 years. He started his career with Nortel Networks, primarily working on central office switching systems, spent two years with Allen Telecom developing cellular infrastructure products, recently worked for Motorola, Inc. on software architectures for cellular phones, and is currently employed as a principal engineer at Tavve Software Company, developing network management solutions. In addition to architecture, Jim's technical interests include patterns, prototyping, and tools development.*

Gerard Meszaros is an acknowledged expert in software architecture and patterns. He has led or participated in workshops on software architecture at OOPSLA since 1994. He has published patterns in the first three volumes of "Pattern Languages of Program Design." His clients include Nova Gas Transmission, Tandem Computers, TELUS Communications, Digital Technics, Intelligent Databases, TransCanada Pipelines, DMR, and IBM. He has been invited to speak or participate in panels at OOPSLA, PLOP, and other national and international conferences.

Ron Crocker is a Senior Member of Technical Staff in the Network and Advanced Technology department in Motorola, Inc. where he is responsible for cellular system architecture and design. He has over 15 years of experience with object-oriented technologies, starting as a C++ guinea pig.

28

Improving Your Use Cases

Monday, Full Day
Convention Ctr — Room 22

Bruce Anderson,
IBM Component Technology Services
Paul Fertig, IBM Global Services

This tutorial is a working session to help you produce effective use cases for functional requirements. We will go beyond simple examples to deal with many of the issues you will face (and we have faced!) in dealing with different kinds of systems, clients, and developers. We will look at some specific topics, such as how use cases relate to business process models, using generic use cases, distinguishing envisioning from designing, effort estimation from use cases, and system exceptions. We will also look at the process of running use case workshops, and at the relation between use cases and other requirements artefacts such as the business rules catalog and non-functional requirements. Attendees are encouraged to bring specific problems for discussion, supported by shareable documents if possible.

Attendee Background: You should have written some use cases and have experience of producing requirements documents. Knowledge of OO would be useful but is not essential.

Presenters: Bruce Anderson, Senior Consultant in IBM Component Technology Services, has been using use cases in his consulting work for several years. He has worked with clients in the banking, insurance, petroleum, and telecom industries. Bruce served on the OOPSLA '98 use case panel, and taught tutorials on use cases at OOPSLA in 1999 and 2000, the latter with Paul.

Paul Fertig, Senior IT Architect in IBM Business Innovation Services, has been responsible for requirements gathering and architecture in large services contracts for a number of years. He has worked with clients in the telecom, retail and investment banking industries. Paul co-authored a book on OO applications which has been a key influence on IBM's world-wide software development method.

29

Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects

Monday, Full Day
Convention Ctr — Room 15

Douglas Schmidt,
University of California, Irvine

Developing software for distributed systems that effectively utilizes concurrency over high-speed, low-speed, and mobile networks is a complex task. This tutorial describes how to apply patterns and frameworks to alleviate the complexity of developing concurrent and distributed communication software. These patterns and framework components have been used successfully by the speaker on production communication software projects at hundreds of commercial companies for telecommunication systems, network management for personal communication systems, Web-based content delivery systems, electronic medical imaging systems, real-time aerospace systems, distributed interactive simulations, and automated stock trading. The tutorial illustrates by example how to significantly simplify and enhance the development of communication software that effectively utilizes concurrency and distribution via the use of:

- OO design techniques — such as patterns, layered modularity, and data/control abstraction
- OO language features — such as abstract classes, inheritance, dynamic binding, and parameterized types
- Middleware — such as object-oriented frameworks for infrastructure middleware (such as ACE) and distribution middleware (such as CORBA ORBs)

The material presented in this tutorial is based on the book, *Pattern-Oriented Software Architecture: Patterns for Concurrent and Distributed Objects* (Wiley 2000), which is the second volume in the highly acclaimed Pattern-Oriented Software Architecture (POSA) series.

Attendee Background: The tutorial is intended for software developers who are familiar with general object-oriented design and programming techniques (such as patterns, modularity, and information hiding) fundamental OO programming language features (such as classes, inheritance, dynamic binding, and parameterized types), basic systems programming concepts (such as process/thread management, synchronization, and interprocess communication), and networking terminology (such as client/server architectures and TCP/IP).

Presenter: *Dr. Schmidt is an Associate Professor in the Electrical and Computer Engineering Department at the University of California, Irvine. He is currently also serving as a program manager the DARPA Information Technology Office (ITO) where he is leading the national research effort on distributed object computing middleware. His research focuses on design patterns, implementation, and experimental analysis of object-oriented techniques that facilitate the development of high-performance, real-time distributed object computing middleware on parallel processing platforms running over high-speed networks and embedded system interconnects. Dr. Schmidt is an internationally recognized and widely cited expert on distributed object computing patterns, middleware frameworks, and Real-time CORBA, and has published widely in top IEEE, ACM, IFIP, and USENIX technical journals, conferences, and books. His publications cover a range of experimental systems topics including high-performance communication software systems, parallel processing for high-speed networking protocols, real-time distributed object computing with CORBA, and object-oriented design patterns for concurrent and distributed systems.*

30

Object-Oriented Reengineering

Monday, Full Day

Marriott Hotel — Meeting Room 12

Serge Demeyer, *University of Antwerp*

Stéphane Ducasse, *University of Berne*

Surprising as it may seem, many of the early adopters of the object-oriented paradigm already face a number of problems typically encountered in large-scale legacy systems. Software engineers are now confronted with millions of lines of industrial source code, developed using object-oriented design methods and languages of the late 80s. These systems exhibit a range of problems, effectively preventing them from satisfying the evolving requirements imposed by their customers. This tutorial will share our knowledge concerning the reengineering of object-oriented legacy systems. We will draw upon our experiences with the FAMOOS project, to show you techniques and tools we have applied on real industrial OO systems to detect and repair problems. In particular, we will discuss issues like tool integration, design extraction, metrics, refactoring, and program visualisation.

Attendee Background: Participants should have practical programming experience in at least one OO language (Smalltalk, C++, Java, Eiffel, ...). Familiarity with UML is useful, though not required.

Presenters: *Serge Demeyer is a professor at the University of Antwerp (Belgium). He served as technical leader for the FAMOOS project and as such has been involved in the organization of several workshops (at ECOOP and ESEC) concerning object-oriented reengineering. He has given tutorials on Object-Oriented Reengineering at both OOPSLA and ECOOP and is currently writing a book reporting on his experience.*

Stéphane Ducasse is a post doctoral researcher at the Software Composition Group in Berne (Switzerland). He served as technical leader for the FAMOOS project and as such has been involved in the organization of several ECOOP workshops concerning object-oriented reengineering. He has given tutorials on Object-Oriented Reengineering at both OOPSLA and ECOOP and is currently writing a book reporting on his experience.

31

Patterns at Work

Monday, Morning

Marriott Hotel — Florida Salon VI

Frank Buschmann, *Siemens AG, Germany*

In this tutorial we present in detail a part of a concrete real-world system and how it is designed with patterns: the representation of physical storage in a warehouse management system as well as the client interface to this subsystem. Step by step we will re-play the process of the system's construction. We discuss the design problems that occur, present the patterns that could help in solving these problems, discuss design alternatives, and show how we actually applied the patterns we selected. By this we will see how the design of the system slowly grows and evolves towards the final architecture. We will also see and discuss how patterns are applied in practice and how they help building high-quality software with predictable properties. The tutorial concludes with a summary of our experiences from several projects in which we applied patterns: what worked, what could be improved, and what did we learn.

Attendee Background: Sound knowledge in object technology, basic knowledge of UML notation, basic knowledge of the pattern concept

Presenter: *Frank Buschmann is software engineer at Siemens Corporate Technology in Munich, Germany. His interests include object technology, frameworks, and patterns. Frank has been involved in many software development projects. He is leading Siemens' pattern research activities. Frank is co-author of Pattern-Oriented Software Architecture — A System of Patterns and Pattern-Oriented Software Architecture — Patterns for Concurrent and Networked Objects.*

32

Designing an Agile Methodology

Monday, Morning

Convention Ctr — Room 19

Alistair Cockburn,

Humans and Technology

The methodology of an organization is a social construction that includes the roles, skills, teaming, activities, techniques, deliverables, standards, habits and culture of the organization as it develops software. This tutorial starts with language and constructs needed to evaluate, compare, and construct methodologies. These include precision, accuracy, tolerance, relevance, and scale, along with the nine basic elements of a methodology. Several examples of effective, lightweight, and real methodologies are given, along with commentary on the social setting for each. The tutorial examines the conditions suited to shifting from a lighter to a heavier methodology and the penalty for doing so. The tutorial ends with the presentation of a small family of agile methodologies, optimized for productivity, making maximum use of human, face-to-face communication. Considerations about success and failure in affecting culture are visited again at the end. Learn to identify and diagnose the parts of your organization's methodology, and learn ways to make it more effective. Attendees should have significant software team experience, preferably but not necessarily OO, and must have used at least one methodology and thought about others.

Attendee Background: Experienced developers, team leaders, methodologists, and technology selectors trying to choose or design a methodology for their organization.

Presenter: Alistair Cockburn, founder of *Humans and Technology*, was special advisor to the Central Bank of Norway for object technology and software project management, and the designer of the IBM Consulting Group's first OO development methodology. His books, *Surviving Object-Oriented Projects* and *Writing Effective Use Cases*, have garnered praise from practitioners for being pragmatic and readable. He is an expert on use cases, object-oriented design, project management, and software methodologies. He has been the technical design coach and process consultant on projects ranging in size from 3 to 90 people. Materials that support Alistair's workshops can be found at <http://members.aol.com/acockburn> and <http://crystalmethodologies.org>.

33

Exposing and Consuming Web Services with .NET

Monday, Morning

Convention Ctr — Room 21

Casey Chesnut, iigo

This tutorial will demonstrate Web Services in the .NET Framework. A Web Service is application logic accessible through standard web protocols and data formats. They are an integral part of the .NET Framework. The tutorial will be divided into 2 parts. The first part will focus on how to expose a Web Service for clients to access, and the second part will cover how to consume the exposed Web Service from a variety of clients. Some time will be spent exploring applicable architectures and modeling techniques for Web Services. Throughout the presentation, non-functional requirements will also be considered (e.g. security, authentication, performance, etc.) The tutorial's objective is to give the audience an intermediate-level introduction to Web Services development in the .NET Framework, as well as design decisions that are pertinent to the programming model. The audience will be exposed to XML, SOAP, UDDI, WSDL, ASP.NET, and C#. This tutorial will be presentation based with code examples.

Attendee Background: The target audience will be Software Engineers, although Management will be interested to get a glimpse at Web Services and the different business models that are made possible. Basic understanding of Internet technologies will be helpful.

Presenter: Casey Chesnut is Vice President of Technology for iigo, Inc. He specializes in cutting-edge technologies and has most recently been concentrating on Web Services in the .NET Framework. He holds two Masters degrees in software engineering.

34

Efficient Architectures for Object-Oriented Component-Based Middleware

*Monday, Morning
Marriott Hotel — Salon A*

Michael Stal, Siemens AG, Germany

Due to the importance of distribution and object technologies, infrastructures for distributed object computing and component-based middleware have become commonplace. However, it is not sufficient to just read the specification of standards such as Java RMI, EJB, or CORBA, and then build applications using these standards. On the one hand, the transparency provided by these platforms helps developers to master the complexity of building distributed systems, but on the other hand, it is necessary to know the infrastructure's internal architectural design to leverage it efficiently. Unfortunately, the architectural principles behind infrastructures are not documented anywhere. Here, patterns come to our rescue. They do not only enable the solution of recurring problems in software development, but also help us to look inside existing software in order to understand it and leverage it efficiently. Thus, the goal of the tutorial is to show the basic principles behind distributed object computing and component-based middleware. Patterns from existing pattern books will be introduced step-by-step to reveal the overall architecture of these infrastructures. These patterns will not only help to understand middleware, but will also be applicable for the development of any distributed systems. In the first part of the tutorial we use patterns to explain the basic architecture of object-oriented middleware from a user perspective. In the second part we will dive into the internals of middleware frameworks.

Attendee Background: Attendees should be familiar with distributed systems. They should have basic experience with Java and C++. Knowledge with patterns is not required.

Presenter: *Michael Stal works as a Senior Principal Engineer for Siemens Corporate Technology where he is head of the Middleware & Application Integration Team. His main research areas include Object-Oriented Middleware, Patterns, Software Architecture, Web Technologies, and Component-based Software Development. Michael is Siemens representative at the OMG, and former member of the C++ standardisation working group X3J16. He is co-author of the books, Pattern-Oriented Software Architecture - A System of Patterns and Pattern-Oriented Software Architecture - Vol. 2: Patterns for Concurrent and Networked Objects. In addition, he serves as editor-in-chief of the German Java Spektrum magazine. Michael has published articles in many magazines and given talks at many conferences world-wide.*

35**Extreme Programming Live!***Monday, Morning**Marriott Hotel — Florida Salon IV**William Wake, Capital One**Steve Metsker, Capital One*

Extreme Programming (XP) is an agile software development method that emphasizes ongoing user involvement, automated testing, and pay-as-you-go design. This tutorial introduces XP practices through hands-on exercises:

- Planning Game: User Stories, On-Site Customer
- Programming Game: Test-First Programming, Unit Testing, Pair Programming
- Refactoring Game: Code Smells, Once-and-Only-Once, Refactoring

The exercises are paper-based and use a fireworks factory as their domain. Student volunteers help play the part of the customer and the unit testing framework. As a participant, you will help create a live simulation of several key practices of Extreme Programming.

Attendee Background: Some familiarity with object-oriented concepts is helpful; no prior experience with XP is needed.

Presenters: *William Wake is interested in XP, patterns, human-computer interaction, and information retrieval. He is the author of Extreme Programming Explored and the inventor of the Test-First Stoplight and the XP Programmer's Cube.*

Steve Metsker is a researcher and author who explores and writes about ways to expand the abilities of developers. Steve's articles have explained how to maintain relational integrity in object models, how to solve logic puzzles in Java, and how the concept of "object" differs between Plato and the OO languages. Steve's most recent publication is the book, Building Parsers with Java.

36

Patterns and Architectures for J2EE Systems

Monday, Morning

Marriott Hotel — Florida Salon V

Kyle Brown, IBM

When people view the J2EE (Java 2 Platform, Enterprise Edition) specifications, too often all they see is a “bag of APIs” without a way to understand how the specifications work together in an application server. In this tutorial we will examine some template architectures for successful J2EE systems and show how a common set of design patterns can be applied to help designers navigate through the J2EE problem space. We will examine common client-side pitfalls and discuss the pros and cons of different design options, discuss how Java Servlets and JavaServerPages (JSPs) can work with and within technologies like Apache Struts, XML, and Extensible Style Sheets (XSL), and discuss how EJB (Enterprise JavaBean) systems can be structured to maximize component reuse while reducing wasted programmer effort.

Attendee Background: This tutorial is targeted to Java programmers and designers, with at least some exposure to J2EE technologies (a reading knowledge of the J2EE specification and the associated API specifications will be sufficient). Programmers who have had experience with one or more of the J2EE technologies will gain the most from this review of how all the technologies fit together and how problems are solved using the entire J2EE framework.

Presenter: *Kyle Brown is an Executive Java Architect with IBM's WebSphere Services unit. He is an experienced presenter at OOPSLA and other industry conferences. He has over twelve years of experience with object-oriented systems, and has been specializing in Enterprise Java systems since 1997. He is a co-author of The Design Patterns Smalltalk Companion and Enterprise Java Programming with IBM WebSphere, both published by Addison Wesley Longman.*

37

Refactoring: Improving the Design of Existing Code

Monday, Morning

Convention Ctr — Room 16

Martin Fowler, *ThoughtWorks, Inc.*

Josh MacKenzie, *ThoughtWorks, Inc.*

Almost every expert in object-oriented development stresses the importance of iterative development. As you proceed with iterative development, you need to add function to the existing code base. If you are really lucky, that code base is structured just right to support the new function while still preserving its design integrity. Of course, most of the time we are not lucky, and the code does not quite fit what we want to do. You could just add the function on top of the code base. But soon this leads to applying patch upon patch, making your system more complex than it needs to be. This complexity leads to bugs, and cripples your productivity. Refactoring is all about how you can avoid these problems by modifying your code in a controlled manner. Done well, you can make far-reaching changes to an existing system quickly, and without introducing new bugs. You can even take a procedural body of code and refactor it into an effective object-oriented design. With refactoring as part of your development process you can keep your design clean, make it hard for bugs to breed and keep your productivity high. In this tutorial we'll show you an example of how a lump of poorly designed code can be put into good shape. In the process we'll see how refactoring works, demonstrate a handful of example refactorings, and discuss the key things you need to do to succeed. This tutorial is an introduction to refactoring. No prior refactoring experience is assumed and the content covers much the same ground as opening a couple of chapters of the refactoring book.

Attendee Background: developers and analysts

Presenters: *Martin Fowler is the Chief Scientist for ThoughtWorks Inc., an Internet professional services provider specializing in the delivery of highly strategic B2B e-Commerce solutions. For a decade he was an independent consultant pioneering the use of objects in developing business information systems. He's worked with technologies including Smalltalk, C++, object and relational databases, and EJB with domains including leasing, payroll, derivatives trading and healthcare. He is particularly known for his work in patterns, the UML, lightweight methodologies, and refactoring. He has written four books: Analysis Patterns, Refactoring, the award winning UML Distilled, and Planning Extreme Programming.*

Josh MacKenzie has been with ThoughtWorks for three years, serving as a developer, architect, and team lead. He has worked on projects in equipment leasing, insurance, and industrial supply and purchasing. These projects have utilized a wide variety of technologies, including J2EE, XML, Forte, and LDAP. Josh has also been instrumental in the exploration and adoption of agile methodologies on ThoughtWorks' projects. Prior to ThoughtWorks, Josh served as a Senior Engineer for Motorola, Inc. Energy Systems, where he designed and developed real-time testing and analysis software for electrochemical capacitors. He holds a B.A. in Physics and Mathematics, and an almost-M.S. in Chemical Engineering. Josh presented tutorials at JavaCon2000 on "Refactoring" and "Business Objects in J2EE."

38

Efficient Implementation of Object-Oriented Programming Languages

Monday, Afternoon

Marriott Hotel — Salon A

Craig Chambers,

University of Washington

How are object-oriented languages implemented? What features of object-oriented languages are expensive? What compiler optimizations have been developed to make object-oriented languages more efficient? This tutorial addresses these questions. After identifying the main features of object-oriented languages that are challenging to implement efficiently, three classes of implementation techniques are presented. First, run-time system techniques such as virtual function dispatch tables (including complications due to multiple inheritance and virtual inheritance) and inline caches are described. Second, static intra- and interprocedural analyses are discussed that seek to identify at compile-time the possible classes of message receivers, in order to reduce or eliminate the overhead of dynamic binding. Third, ways in which dynamic execution profiles can be exploited to complement static analysis techniques are described. To assess the relative importance of the techniques, empirical measurements of the effectiveness of many of these techniques, as implemented in the Vortex optimizing compiler, are presented for large benchmarks written in Java, C++, and Cecil.

Attendee Background: Attendees should be familiar with the features of object-oriented languages and also with traditional compiler techniques such as procedure inlining and data flow analysis.

Presenter: *Craig Chambers has been researching object-oriented language design and implementation since 1987, with publications in OOPSLA, ECOOP, ISOTAS, PLDI, POPL, PEPM, and TOPLAS on the topic. For his Ph.D. thesis at Stanford, he developed the first efficient implementation of the Self language, using optimizing dynamic compilation. Chambers is currently an Associate Professor of Computer Science & Engineering at the University of Washington, where he designed the Cecil language, heads the Vortex whole-program optimizing compiler project, and co-leads the DyC staged dynamic compilation project.*

39

Making the Software Process Transparent by Using Intelligent Agents

Monday, Afternoon

Marriott Hotel — Florida Salon V

Ivar Jacobson, *Rational*

Gunnar Övergaard, *Jaczone AB*

It has never been so hard to develop good software as today. Developers need more knowledge and skill than ever before. They need to be skilled in programming languages (e.g. Java, C++), system software platforms (.NET, J2EE), XML, middleware (WebSphere, Logicworks, etc.), the Unified Modeling Language, the Rational Unified Process, web architectures, etc. And they need to learn about these technologies faster than ever with almost no time for training and education. If they don't, their only rescue is to find shortcuts, use lightweight methodologies, and ignore well-proven best practices. And as usual, quality will suffer. There is another way. In this tutorial we will discuss how software agents can be used to reduce the gap between the individual developers' knowledge and what is needed. For instance, agents can minimize the process adoption thresholds so that the complexity of a process can become transparent to the developers and thus be perceived as lightweight. The individual developer will focus on the problem solving and creative part, letting the agents do the work that can be guided by formalized knowledge. We will discuss the process of formalizing knowledge as rules, how these rules will trigger in a given context, and how the agents can propose resolutions. Examples will be used to demonstrate the feasibility of agents in software development.

Attendee Background: System analysts, project leaders, software developers, people interested in methodologies, process development and software development tools
Required experience: Some experience with software development and UML.

Presenters: *Dr. Ivar Jacobson serves as Vice President of Business Engineering for Rational Software Corp. Dr. Jacobson is the founder of Objectory AB in Sweden, which merged with Rational Software in 1995. He was one of the three original designers of the UML in 1997. He is the principal author of three influential and best-selling books, Object-Oriented Software Engineering—A Use Case Driven Approach, The Object Advantage—Business Process Reengineering with Object Technology, Software Reuse: Architecture, Process, and Organization for Business Success, and The Unified Software Development Process.*

Gunnar Övergaard serves as Vice President Content Development at Jaczone AB, and holds a Ph.D. in Computer Science. Gunnar has worked with process development, consulting, and education in the object-oriented field since the mid-1980s. Gunnar worked as VP of process development at Objectory in the critical development of the origin to the Rational Unified Process. He has participated actively in the development of UML since 1995.

40

Surviving Object-Oriented Projects

Monday, Afternoon

Marriott Hotel — Florida Salon VI

Alistair Cockburn,

Humans and Technology

After a decade of OO projects, what have we learned? Develop in increments to form a habit of delivering; make sure you have a decent executive sponsor, project manager and technical leader; get training; and don't forget the lessons of the last four decades. This tutorial walks through appropriate expectations, project setup, technology selection, working with the domain model, with increments and iterations, prototyping and the death spiral, setting up teams, handling training, expanding, project management patterns and advice from hindsight. It includes over 60 specific strategies, truths, fixes, and recommendations. The tutorial is liberally sprinkled with case histories and insights from the participants themselves. It overlaps heavily with the book, *Surviving Object-Oriented Projects*. The tutorial is for anyone wanting to learn ways to get their OO project out of the mire, or avoid the mire in the first place.

Attendee Background: Neither being a OO novice nor an OO expert will interfere with the tutorial material. The only requirements is an interest in what makes a project work well.

Presenter: Alistair Cockburn, founder of *Humans and Technology*, was special advisor to the Central Bank of Norway for object technology and software project management, and the designer of the IBM Consulting Group's first OO development methodology. His books, *Surviving Object-Oriented Projects* and *Writing Effective Use Cases*, have garnered praise from busy practitioners for being pragmatic and readable. He is an expert on use cases, object-oriented design, project management, and software methodologies. He has been the technical design coach and process consultant on projects ranging in size from 3 to 90 people. Materials that support Alistair's workshops can be found at <http://members.aol.com/acockburn> and <http://crystalmethodologies.org>.

41

Refactoring to Patterns

Monday, Afternoon
Convention Ctr — Room 21

Joshua Kerievsky, *Industrial Logic*

While Software Patterns are undeniably powerful design aids, many programmers tend to overuse them, prematurely introduce them, or implement them in unnecessarily heavyweight ways. Refactoring to Patterns encourages a simpler, more disciplined approach to using Patterns, based on the philosophy of Extreme Programming. Using this approach, programmers wait for the right time to refactor a Pattern into a system and do so using the simplest possible Pattern implementations. In this tutorial, we will examine five Design Patterns and five cases where we might refactor these Patterns into Java code. During the process, we will investigate when it makes sense to refactor to a Pattern, and what are simple implementations of each Pattern we add.

Attendee Background: This is an intermediate-level tutorial. Attendees will be expected to understand Java and have basic exposure to Design Patterns.

Presenter: *Programming professionally since 1987, Joshua Kerievsky is the founder and chief programmer of Industrial Logic, Inc. (<http://industriallogic.com>), a company specializing in Patterns and XP. As an XP Coach, mentor, and leader of intensive workshops, Joshua helps organizations learn and use the software industry's very best practices. Joshua can be reached at Joshua@industriallogic.com.*

42

How to Really Fail at Software Architecture

Monday, Afternoon
Convention Ctr — Room 19

Luke Hohmann, *Independent Consultant*

Most books and lectures on software architecture focus on technical issues. This is clearly necessary, because software architecture must deal with technical concerns. A smaller subset focuses on other important issues such as “peopleware.” This is clearly necessary, for software systems are built by people to satisfy one or more needs. Unfortunately, few lectures focus on the business realities of software architecture. This tutorial addresses these business realities, for without addressing them your architecture will surely fail.

Attendee Background: Participants should have been a technical lead, first line manager, senior developer, or software architect for at least one project (including the one they’re working on right now, if this is their first).

Presenter: *Luke Hohmann is an independent consultant committed to coaching his clients to greater levels of performance. Mr. Hohmann is author of Journey of the Software Professional: A Sociology of Software Development from Prentice-Hall as well as numerous articles on software engineering management. A skilled instructor and speaker, Mr. Hohmann has been invited to many conferences. Mr. Hohmann can be contacted through e-mail at LukeHohmann@yahoo.com.*

43

The Art of Writing Use Cases

Tuesday, Full Day

Marriott Hotel — Meeting Room 12

Rebecca Wirfs-Brock,
Wirfs-Brock Associates

John Schwartz, Wirfs-Brock Associates

Use cases describe the behavior of a software system from an external usage perspective. There is an art to writing them clearly. Written carefully, use case models convey key usage specifications and can be tied to other requirements. Written poorly, use cases are confusing and ambiguous. This tutorial presents examples of good and bad use case descriptions, and practical techniques for writing three forms of descriptions: narratives, scenarios, and conversations. Narratives are high-level descriptions written from an external perspective. We show how to elaborate high-level descriptions, choosing either a scenario form, which emphasizes sequence, or a conversation, which highlights interactions between a user and the system. Tips for naming use cases, describing policies, errors, and exceptions, attaching other important information, describing meaningful pre- and post-conditions, and creating informative glossary entries are also presented. This tutorial will expose students to techniques for critically reading and revising use cases in various forms, techniques for asking probing questions and filling in use case details, and techniques for developing a use case model which interleaves both group activities with individual writing tasks and review.

Attendee Background: Attendees should be looking for practical ways to improve their writing. They should be familiar with writing and reading software requirements and usage descriptions. Attendees could benefit from an introduction to object concepts. However, an object background is not a prerequisite!

Presenters: *Rebecca Wirfs-Brock is president of Wirfs-Brock Associates, a firm specializing in the transfer of object analysis and design expertise to organizations and individuals through training, mentoring, and consulting. Rebecca has been involved with object technology since its infancy. She is the inventor of the set of development practices known as Responsibility-Driven Design. From development on the Tektronix implementation of Smalltalk in the early 1980s, through years of development and training experience, she is recognized as one of only a few knowledgeable and influential practitioners of object-oriented design. She spent 17 years as a Software Engineer at Tektronix, where she managed the first commercial Smalltalk effort and was the technical lead for the development of Color Smalltalk. Recently, she has authored use cases for a telecommunications framework and an online banking system and has mentored teams in use case writing, design, architecture and managing incremental, iterative object-technology projects. She practices what she teaches!*

John Schwartz is Vice President of Consulting Services at Wirfs-Brock Associates and a widely known and respected authority on object analysis and design. John has over 15 years of experience developing and managing object-oriented projects in telecommunications, medical, and CAD. He has served as Vice President and Director of Software Architecture of a 120-person telecom information technology group. While with ParcPlace Systems, he influenced the development of the Object Behavior Analysis method pioneered by Adele Goldberg and Kenny Rubin. John was chairman of the OMG's original Object Model Task Force, and developed the model that CORBA is based on. He has contributed to the definition and practical application of Object Behavior Analysis, and Responsibility-Driven Analysis and Design methodologies. He consults on design and methodology to major object-oriented projects. He has conducted over 100 tutorials and classes on object analysis, design, and programming.

44

Architecting Large Business Systems

Tuesday, Full Day

Convention Ctr — Room 25

Jens Coldewey, *Coldewey Consulting*

Alan O'Callaghan,
De Montfort University

Wolfgang Keller, *Generali Vienna Group*

Architecture is one of the key issues in successful software projects. In the recent years, many publications have discussed what architecture is about and how to describe it. However, there has been little information about how a good architecture is found and how to implement it into a project successfully. Based on the field research at De Montfort University and many years of hands-on experience, this tutorial teaches best-practices on how to develop an architectural model for business systems. It addresses both “green-field” projects and re-engineering efforts with a special emphasis on agile processes and on component architectures. The tutorial is designed interactively with more than half of the time dedicated for exercises and discussion.

Attendee Background: This tutorial aims at designers and project managers of object-oriented business systems who are interested in software architecture and who are open to interactive learning experiences. They should be familiar with object-oriented design patterns as published by Gamma, Helm, Johnson, and Vlissides in “Design Patterns — Elements of Reusable Object-Oriented Software”

Presenters: Jens Coldewey

(jens_coldewey@acm.org) is independent consultant in Munich, Germany, specialized in deploying agile processes and object-oriented techniques in large organizations. He consults architecture projects in several large projects. Jens Coldewey writes a column on Agile Processes in the German SIGS/101 magazine OBJEKTSpektrum.

Alan O'Callaghan (aoc@dmu.ac.uk) is Senior Lecturer in computer science at De Montfort University, Leicester, England. He has consulted in the migration of legacy systems to object and component-based systems in a number of industrial sectors and authored the ADAPTOR pattern language. He writes a column on migration in the SIGS/101 journal Application Development Advisor.

Wolfgang Keller

(wolfgang_keller@acm.org) is a principal architect for Generali Vienna Group. His responsibilities include the technical base for Generali's Phoenix line of insurance applications, product architecture, and project coordination for Generali's distributed development across parts of Europe.

45

The UML's Object Constraint Language (OCL) — Specifying Components

Tuesday, Afternoon

Marriott Hotel — Meeting Room 11

Jos Warmer, *Klasse Objecten*

Anneke Kleppe, *Klasse Objecten*

As the use of UML grows and UML is applied to more fields of software and systems engineering, the need for more precise specifications grows. This is crucial, for example, when generating code or test cases from specifications. In the component based world we need to be able to specify the behavior of components in a very precise way. This enables us to know whether components are plug-compatible and allows us to derive the behavior of assembled components and make sure that they have the desired behavior. For these purposes UML's Object Constraint Language is becoming more popular as a standardized and language independent specification mechanism. This tutorial shows the importance of constraints as an object-oriented specification technique and how they add value to the visual modeling techniques of UML. The OCL language itself and the connection with the visual UML diagrams is thoroughly explained. The final part of the tutorial will show how one can apply constraint modeling in UML to achieve the above described goals.

Attendee Background: The tutorial is targeted to people that have knowledge of and experience with analysis and design methods like UML. They should specifically have experience in developing object or class models.

Presenters: *Jos Warmer is senior consultant at Klasse Objecten. He is the chief architect of OCL and responsible within the UML core team for all matters concerning OCL.*

Anneke Kleppe is an independent OT consultant who founded her own company Klasse Objecten in 1995. She has developed her own training and mentoring program and has applied this with many clients. She has actively supported the UML core team on the subject of OCL. Anneke and Jos have co-authored the book entitled, The Object Constraint Language: Precise Modeling with UML, which has been published in the OT series by Addison-Wesley Longman, USA. They also wrote (Dutch) books on OMT and UML, published by Addison-Wesley.

46**J2ME Design and Development Considerations**

*Tuesday, Afternoon
Convention Ctr — Room 16*

David Hemphill,
Catapult Technologies, Inc.

Java 2 Micro Edition (J2ME) is Sun's Java 2 platform for consumer electronics and embedded devices. Writing software for limited devices offers unique challenges to the application developer. In many cases, developers are required to improvise in graphical user interface, persistent storage, I/O, and other areas in order to make the application work under resource constraints. The extremely varied nature of many of the targeted devices makes solving issues even more challenging. In this presentation, many of the difficulties and potential solutions of developing mobile, wireless, and other limited device J2ME applications will be addressed.

Attendee Background: Participants should be familiar with Java programming and have some familiarity with J2ME.

Presenter: *David Hemphill is a Senior Software Architect with Catapult Technologies, Inc. He has over ten years of experience in developing and architecting software systems. David's technical expertise lies in Java, J2EE, J2ME, EJB, UML, XML, and relational databases. He is a graduate of the University of Wisconsin, Eau Claire.*

47

Embedded Systems in C++ — C++ Idioms, Patterns, and Architecture for Constrained Systems

*Tuesday, Afternoon
Convention Ctr — Room 5*

Detlef Vollmann,
Vollmann Engineering GmbH

Embedded software is different: it often involves hard real-time constraints and very limited memory. But the challenges for embedded systems are even higher: they must be very reliable (99.999% is not good enough) and need to tackle a variety of different memory types (standard RAM, EEPROM, Flash, buffered RAM, ...). Another frequent design problem is the different kinds of processing tasks: interrupt handling, hardware control, application processes, all interconnected by a selection of communication means. To design a system in such an environment, a rich programming language like C++ seems to be quite useful. But on the other hand, it is often more difficult to keep the tight control necessary, e.g., for hard real-time requirements, with such a high-level language. This tutorial will present and discuss various techniques for designing and implementing typical embedded systems problems in C++. Different language features are analyzed with respect to embedded requirements. Participants of this tutorial will acquire a thorough understanding of the potentials of C++ for the development of systems with tight limitations. They will learn effective techniques to deal with these limitations even for complex systems.

Attendee Background: Participants should have a good working knowledge of ISO C++. Experience in the design of embedded systems will be helpful but is not essential.

Presenter: *Detlef Vollmann has a background of 15 years in software engineering and more than 10 years with object technology. As an independent consultant he supports several Swiss companies with the design of object-oriented systems. Since 1991, he has authored and taught courses in C++, Object-Oriented Technologies, Software Architecture, and Distributed Computing for major Swiss companies.*

48

Adaptive Object-Model Architecture: How to Build Systems That Can Dynamically Adapt to New Business Requirements

Tuesday, Afternoon
Convention Ctr — Room 15

Federico Balaguer,
Software Architecture Group -
Department of Computer Science,
University of Illinois

Joseph Yoder,
Software Architecture Group -
Department of Computer Science,
University of Illinois

Architectures that can dynamically adapt to changing requirement are sometimes called “reflective” or “meta” architectures. We call a particular kind of reflective architecture an “Adaptive Object-Model (AOM)” architecture. This tutorial will explain AOMs and how to implement them. An Adaptive Object-Model is a system that represents classes, attributes, and relationships as metadata. It is a model based on instances rather than classes. Users change the metadata (object model) to reflect changes in the domain. These changes modify the system’s behavior. In other word, it stores its Object-Model in a database and interprets it. Consequently, the object model is active, when you change it, the system changes immediately. We have noticed that the architects of a system with Adaptive Object-Models often claim this is the best system they have ever created, and they brag about its flexibility, power, and eloquence. At the same time, many of the developers find them confusing and hard to work with. This is due in part because the developers do not understand the architecture. This tutorial will give a description of this architectural style and will make it easier for developers to understand and build systems that need to quickly adapt to changing business requirements.

Attendee Background: A good knowledge of object concepts is required. It would be useful if participants have a basic understanding of frameworks, though it is not necessary. A general understanding of the GOF patterns is required and Fowler’s Analysis Patterns is helpful.

Presenters: *Federico Balaguer has been developing object-oriented software for over ten years. He has worked on many projects including J.P. Morgan in Argentina. He is currently working on implementing Martin Fowler’s Analysis Patterns at Illinois Department of Public Health and is also working with Professor Ralph Johnson on finishing his Ph.D.*

Joseph W. Yoder has worked on the architecture, design, and implementation of various software projects dating back to 1985. Recently he has taught Object-Oriented concepts including Patterns and Smalltalk to Caterpillar and the Illinois Department of Public Health (IDPH) analysts and developers, and has mentored many developers on the development applications being deployed across the state of Illinois such as the Newborn Screening application, the Refugee System, and the Food Drug and Dairy application. He is also coordinating the efforts of this development as the primary architect of the reusable frameworks being developed. Joe is the author of over two dozen published patterns and has been working with patterns for a long time, writing his first pattern paper in 1995, and chairing the PLoP’97, conference on software patterns.

49

No Stone Unturned: An Introduction to Test-First Programming

Tuesday, Afternoon
Convention Ctr — Room 3

Steve Freeman, *Big Blue Steel Tiger*

Writing effective unit tests is a skill that takes practice to do well but is at the core of Extreme Programming. During this tutorial we will demonstrate real examples that address situations many programmers find hard to test. We will also show how test-driven development improves the quality of the code produced. Finally, we will consider the practical boundaries of effective unit testing. This tutorial will improve attendees' understanding of how to write meaningful and effective unit tests, demonstrate live, test-driven development and pair-programming, and cover topics that many people find difficult when writing unit tests.

Attendee Background: The session is intended for working programmers who are interested in writing effective unit tests. They should be familiar with Java and, preferably, standard libraries such as JDBC and servlets. The tutorial has a bias towards web development, but the techniques it covers are applicable elsewhere.

Presenter: *Steve Freeman is a Principal Consultant at Big Blue Steel Tiger, where he develops e-commerce solutions and is also responsible for helping to move Big Blue Steel Tiger towards Extreme Programming. Prior to this, he ran the largest XP project in the UK at Lombard Risk Systems. He has degrees in Statistics and Music, and a Ph.D. in computer science from Cambridge University and has written software for research labs, shrink-wrap and bespoke systems.*

50

Designing Software Architecture for Quality: The ADD Method

Tuesday, Afternoon
Convention Ctr — Room 22

Len Bass,
Software Engineering Institute
Felix Bachmann, Robert Bosch, GmbH

It has long been recognized that quality attributes are, in large part, determined by the software architecture of a system. This recognition has been the basis for several software architecture analysis methods. The application of this recognition to the problem of designing the software architecture is a recent development, however. This tutorial will introduce the Attribute Driven Design (ADD) method. ADD is a method for designing the software architecture of a system or collection of systems based on an explicit articulation of the quality attribute goals for the system(s). The method is appropriate for any quality attributes but has been particularly elaborated for the attributes of performance, modifiability, security, reliability/availability and usability. The method has been used for designing the software architecture of products ranging from embedded to information systems.

Attendee Background: This half-day tutorial is designed for attendees who have practical knowledge of software architecture and experience in working with and designing large systems.

Presenter: *Len Bass is a senior software engineer at the Software Engineering Institute of Carnegie Mellon University (CMU). He has written or edited six books and numerous papers in a wide variety of areas of computer science including software engineering, human-computer interaction, databases, operating systems, and theory of computation. His most recent book, Software Architecture in Practice (co-authored with Clements and Kazman), received the Software Development Magazine's Productivity Award. He headed a group that developed a software architecture for flight training simulators that has been adopted as a standard by the U.S. Air Force. He also headed a group that developed a*

technique for evaluating software architectures for modifiability. He is currently working on techniques for the analysis of software architectures, on techniques for the development of software architectures for product lines of systems, and on the how to achieve usability through architectural means. He is the representative of the ACM to the International Federation of Information Processing technical committee on Software: Theory and Practice. Before joining CMU in 1986, he was professor and chair of the Computer Science Department at the University of Rhode Island. He received his Ph.D. in computer science in 1970 from Purdue University.

Mr. Bachmann is currently Project Manager for the Product Line approach within Robert Bosch, GmbH. In cooperation with the Product Line Systems Program of the Software Engineering Institute (SEI), he makes this approach available to the Bosch business units. Prior to his current assignment, he worked as a member of the Robert Bosch research institute with the software development departments to address the issues of more functions and higher quality in the "call-control software," — the core of telecommunication products. This is where he developed the foundation for the next generation of telecommunications software. As a result of these efforts, Bosch developed the method OTES (Objects Through Essential Services) in which Mr. Bachmann played a decisive role. Mr. Bachmann also defined the corresponding software development process that describes in three levels how to develop high quality software in a timely fashion.

51

Creating Responsive, Scalable Systems

Tuesday, Afternoon

Convention Ctr — Room 13

Connie Smith,
Performance Engineering Services
Lloyd Williams,
Software Engineering Research

Performance, both responsiveness and scalability, is an important quality of today's software. Yet, many software systems cannot be used as they are initially implemented due to performance problems. These performance failures can translate into significant costs due to damaged customer relations, lost income, and time and budget overruns to correct the problem. Our experience is that performance problems are most often due to fundamental architectural or design problems rather than inefficient coding. Thus, performance problems are introduced early in the development process but are typically not discovered until late, when they are more difficult and costly to fix. This tutorial presents a systematic, quantitative approach to cost-effectively designing performance into object-oriented software systems. It also presents an overview of principles and patterns for designing performance into software as well as antipatterns for recognizing and fixing common problems. Objective: Participants will be acquainted with a cost-effective, quantitative approach to managing software performance. They will learn practical techniques for diagnosing performance problems early in the development process when these problems can be fixed quickly and easily. They will also learn techniques for designing performance into software.

Attendee Background: Attendees should be familiar with object-oriented development. No background or experience in software performance engineering is required.

Presenters: *Connie U. Smith, Ph.D., a principal consultant of the Performance Engineering Services Division of L&S Computer Technology, Inc., is known for her work in defining the field of Software Performance Engineering (SPE) and integrating SPE into the development of new software systems. Dr. Smith received the Computer Measurement Group's prestigious AA Michelson Award for technical excellence and professional contributions for her SPE work. She authored the original SPE book, Performance Engineering of Software Systems, and approximately 100 scientific papers. She is the creator of the SPE-ED performance engineering tool. She is a frequent speaker at conferences and has delivered numerous keynote addresses on SPE. In her work at L&S Computer Technology she specializes in the development and support of the performance engineering tool, SPE-ED, applying performance prediction techniques to software, teaching SPE seminars, and research and writing on SPE.*

Dr. Lloyd G. Williams is principal consultant at Software Engineering Research where he specializes in the development and evaluation of software architectures that meet quality of service objectives such as: performance, reliability, modifiability, and reusability. Dr. Williams was previously Associate Professor of Computer Science at the University of Colorado at Denver. He also served as Director of the Rocky Mountain Institute of Software Engineering, a non-profit organization founded to promote research and education in software engineering. His work has emphasized the transfer of leading-edge software engineering technology into widespread use. He has presented professional development seminars and served as a consultant on software development for more than 100 organizations in the USA, Japan, and Europe. He has authored numerous technical articles and is a contributor to the AIAA Progress Series book, Aerospace Software Engineering.

52

Leading Retrospectives on OO Projects: Looking Back to Move Forward

*Tuesday, Afternoon
Convention Ctr — Room 14*

Norm Kerth, *Elite Systems*

Linda Rising, *Independent Consultant*

In the object-oriented world, it's our fate to experiment with new ideas. It started with objects and continued with OMT, Fusion, Objectory, UML, Patterns, Extreme Programming and so on. Our community is good at accepting new concepts but are we good at evaluating the results? "You have paid the tuition, now are you going to learn the lessons?" Carefully reviewing an OO project at its end is one of the most obvious ways of improving your software development process and building OO mastery. Sadly, such a review rarely happens — why? Because we don't know how to do it; we don't know how to deal with the emotions; we don't know how to address failure; and we don't know how to convert the lessons into new practices. In this tutorial, we look at an entire philosophy and methodology to lead an effective review of a significant OO project. Expect to learn a number of tools, techniques and skill areas necessary to be an accomplished facilitator.

Attendee Background: Managers, Project Leaders, Software Process Group Specialists, Trained Facilitators, Methodologists, Technical Leaders

Presenters: *Norm Kerth is an experienced software engineer and researcher focusing on specification and design activities, quality assurance, continuous process improvement, project management and growing effective teams. He has led retrospectives for over 20 years and critics predict his book, Project Retrospectives, will become "the next classic in our field." Norm has been a full time consultant since 1984 and helps firms improve their software engineering discipline. He has particular interest in objects, pattern languages, and building high performance teams. Prior to starting his company, Elite Systems, he was a professor at the University of Portland. He has a decade of engineering experience with Tektronix and is a master teacher, with over 30 years of experience in front of students.*

Linda Rising has a Ph.D. from Arizona State University in the area of object-based design metrics. Her background includes university teaching experience as well as work in industry in the areas of telecommunications, avionics, and strategic weapons systems. She has been working with object technologies since 1983. She is the editor of A Patterns Handbook, The Pattern Almanac 2000, and Design Patterns in Communication Software.

53

Business Modeling with the UML

Tuesday, Afternoon

Marriott Hotel — Meeting Room 1

Granville Miller, *TogetherSoft*

Developing models of your business can serve many purposes. These models can help you as you continually engineer your business to keep up with the competitive pressures of today's business environment. Additionally, they can help as you develop the software systems that give you competitive advantage. UML modeling elements such as use cases and business object models are excellent ways of capturing existing business processes or the processes of the future. These models can then be used to communicate requirements to software engineers, the organization at large, or the people who will be a vital part of carrying out the objectives of the process. In this tutorial, we describe methods of capturing processes and modeling them. We present UML as a method of describing them. Finally, we introduce some additions to the UML which will aid organizations to deploy these systems and processes in their organizations.

Attendee Background: This tutorial is geared toward business analysts, requirements engineers, and software managers and developers who wish to place their software solutions in the context of a business process.

Presenter: *Granville Miller is currently a mentor at TogetherSoft. He is the co-author of Advanced Use Case Modeling, Volume I, and has two more books slated for the end of the year. He has been active in the OO community and has given tutorials at several conferences. He has been working heavily with use cases as a method of business modeling for several years.*

54

XP Meets UML: Development Processes for eTechnology

Tuesday, Afternoon
Convention Ctr — Room 23

Alan Cameron Wills,
Trireme International Ltd.

Extreme Programming has justifiably become wildly popular, especially among developers of eCommerce systems, for whom lightweight process and responsiveness to changing requirements are prerequisites. And yet practices like analysing and documenting requirements upfront used to seem like a good idea. So did careful definition of the interfaces between components — an essential for CBD and EAI. This tutorial draws on the presenter's experience in constructing a corporate process that combines the best of iterative development together with requirements and interface specification techniques from the Catalysis approach. We will look at how to ensure that requirements analysis raises questions to the customer, rather than just recording use-cases; how to relate the incrementally-delivered features to the use-cases in the requirements model; how to use the requirements and interface models to generate tests; and how to ensure that the customer-developer link is enhanced rather than weakened by the mediation of the requirements analyst. Participants will be able to improve the processes in their own companies. Part of the session will be in workshop mode, and contributions from people with experience in this area are welcome.

Attendee Background: Some UML; some incremental development.

Presenter: *Alan Cameron Wills is a consultant in methods and process working in a variety of fields on both sides of the Atlantic. He is joint author of the Catalysis Approach to Component and Object Design.*

55

Component-Based Design: A Complete Worked Example

*Tuesday, Afternoon
Convention Ctr — Room 24*

John Daniels,
Syntropy Limited, UK

Much has been written about component-based design but most books and articles deal in generalizations and possibilities, rather than providing explicit and clear guidance. This tutorial will follow a small case study from requirements capture to code-ready specifications, and will set out the full client and server architectures needed to make it work. The modeling and specification techniques used will follow the UML, with the target technologies being EJB or COM+. A web-based UI is assumed. To complete the example, the tutorial will show how to implement the case study using EJB.

Attendee Background: The tutorial is aimed at modelers, system designers and architects. A working knowledge of UML is required, and some understanding of EJB or COM+ would be an advantage.

Presenter: *John Daniels is a consultant at Syntropy Limited, providing help with system architectures and development processes to a number of large corporations. He was previously Application and Technical Architect for Bankers Trust in London, and before that Managing Director of pioneering consulting and training company Object Designers Limited. He has applied object technology in a range of industrial and commercial applications since 1985. He has given tutorials at many object technology conferences, has prepared and delivered many training courses, and has published extensively. He is co-author of Designing Object Systems (Prentice-Hall 1994) and UML Components (Addison-Wesley 2001).*

56

Developing Java Applications for Small Spaces

*Tuesday, Afternoon
Convention Ctr — Room 6*

Chris Carpenter,
RoleModel Software Inc.

Chris Collins,
RoleModel Software Inc.

Java has always promised the ability to run on any size platform, from mainframes to wristwatches. Now the promise of supporting “small” platforms is truly here. This tutorial will teach the current state of the art with respect to developing Java applications for the J2ME and IBM VisualAge Micro Edition application environments (JAE’s). Developers accustomed to creating applications for the web, desktop, or workstation environments will receive instruction on how to approach developing their own small-environment Java applications from pioneers who have actually attempted (successfully and unsuccessfully) to build such applications using the different environments. There are significant differences in Sun’s and IBM’s approach to putting Java in small spaces. The benefits and limitations of each will be discussed. The tutorial will not only discuss how to get Java applications working on small devices but will also provide practical advice about when putting Java on a small device makes sense and when it does not. In addition, the tutorial will discuss integrating Java enabled devices into the much-ballyhooed Jini environment.

Attendee Background: Participants should be Java developers or technical leaders of Java projects.

Presenters: *Chris Carpenter is a senior level software engineer and architect working for RoleModel Software, Inc. Mr. Carpenter has been involved in object oriented development since 1991. He cut his object teeth building object-oriented distributed software frameworks written in Objective-C running on NeXTs. In the early 90’s he participated in architecting and building distributed systems frameworks based on the then emerging CORBA 1.0 specification. His skill at looking beyond the obvious and finding solutions to his customer’s problems has always been tied to the maxim “model the world the way you want it to be.” Recently, he has been involved in the design and prototype of Java in small, remote devices and their integration into infrastructures that rely upon the remote devices for system solutions. Mr. Carpenter is an author of the Automated Meter Reading System patent along with patents pending involving Java and Jini in remote devices joined to larger enterprise frameworks.*

Chris Collins is a Senior Software Developer at RoleModel Software, Inc. While at RoleModel, Chris has created an acceptance test framework, developed an embedded Java application for a new Motorola, Inc. cell phone platform, and ported JUnit to run on Sun’s J2ME platform. Before joining RoleModel in early 2000, he spent five years developing software for several organizations using many different languages for U.S. Department of Defense. Chris has a Masters in Computer Science and Software Engineering from the University of West Florida, currently teaches a Java programming course at North Carolina State University, has been an invited speaker on XP at Duke University, and presented a paper on process adaptation at XP2001.

57

Patterns for Making Your Business Objects Persistent in a Relational Database World

Wednesday, Afternoon
Convention Ctr — Room 3

Joseph Yoder, *The Refactory, Inc.*

For developing simple client-server applications, development environments such as VisualAge provide a visual language for generating the mappings of GUIs to database values and domain objects. For complex applications, tools such as TOPLink are very useful for simplifying the creation of persistent objects while hiding their implementation details. Quite often, application development requires tools for persistence that fall in between these two extremes. Just using the facilities provided by JDBC is not sufficient to work with objects. JDBC forces developers to work at the SQL level with rows and columns. Application developers do not want or need to write SQL statements to read or store their objects; they are busy solving the domain problem. This tutorial will describe how to make business objects persistent by mapping them to a relational database with minimal effort. It will also examine the patterns used to map domain-objects to a relational database. Participants of this tutorial will learn a set of patterns and a language-independent object model that can be used for mapping business objects to a relational database. They will also learn how to develop a data access layer along with the design patterns used in the database tools provided by VisualAge and TOPLink.

Attendee Background: Basic knowledge of object concepts is required. A general understanding of relational databases and/or SQL is required. An understanding of patterns can be useful, but it is not required. The examples will be in Java so understanding the basics of Java is also desirable, but not necessary to understand the object-model.

Presenter: *Joseph W. Yoder has worked on the architecture, design, and implementation of various software projects dating back to 1985. These projects have incorporated many technologies and range from stand-alone to client-server applications, multi-tiered, databases, object-oriented, frameworks, human-computer interaction, collaborative environments, and domain-specific visual languages. Recently he has taught object-oriented concepts including Patterns and Smalltalk to Caterpillar and the Illinois Department of Public Health (IDPH) analysts and developers, and has mentored many developers on the applications being deployed across the state of Illinois, such as the Newborn Screening application, the Refugee System, and the Food Drug and Dairy application. He is also coordinating these development efforts as the primary architect of the reusable frameworks being developed and used for these applications. Joe is the author of over two dozen published patterns and has been working with patterns for a long time, writing his first pattern paper in 1995, and chairing the PLoP'97 conference on software patterns.*

58

Creativity in Software Development

Wednesday, Afternoon

Marriott Hotel — Meeting Room 11

Pete McBreen, McBreen Consulting

Now, more than ever, software development requires innovative thinking. Our challenge has shifted from writing the code to identifying and evaluating new ideas, processes and applications. Creating software is one of the most creative activities that humans undertake. The main limitation in software is the human imagination, and the limits on that are all self imposed. Through the application of creativity, it is possible to create truly great software. This tutorial explores ideas about creativity and how they relate to software development. Specific topics that will be covered include - brainstorming techniques for eliciting requirements - creating and evaluating alternate designs - creativity and software development processes - creativity and quality assurance - creativity for programmers. Objective: On completion participants will understand how to apply creative thinking strategies to software development.

Attendee Background: Developers, team leaders and managers who need to step up to the challenges of developing great software.

Presenter: *Pete McBreen is a course designer, teacher, and project lead in object technology. He is responsible for ensuring that project teams make effective use of object technology on projects including project startup, methodology and tool selection, mentoring, process improvement, system design and quality assurance. With over 16 years of industry experience, he has been successfully using and teaching object-oriented techniques since 1989.*

59

Architectures for Integrating Business Logic into J2EE*Wednesday, Afternoon**Convention Ctr — Room 22*Josh MacKenzie, *ThoughtWorks, Inc.*Rebecca Parsons, *ThoughtWorks, Inc.*

J2EE has exploded onto the enterprise systems development world with a handy mix of powerful technologies. They make for an impressive list of features, but they are still young and it isn't so obvious how you actually put them together.

ThoughtWorks likes being on the bleeding edge, so in the last couple of years we've put into production some sizeable enterprise applications. In doing this we've learned a lot about the different architectures you can use and their relative merits. In this talk we'll walk you through half a dozen architectural designs that we've seen in the field. We'll review the issues that you need to understand to build enterprise applications and use these issues to evaluate the candidate architectures. In passing we'll discuss a number of key patterns for server-side object design.

Attendee Background: Knowledge of Java and J2EE.

Presenters: *Josh MacKenzie has been with ThoughtWorks for three years, serving as a developer, architect and team lead. He has worked on projects in equipment leasing, insurance and industrial supply and purchasing. These projects have utilized a wide variety of technologies, including J2EE, XML, Forte, and LDAP. Josh has also been instrumental in the exploration and adoption of lightweight methodologies on ThoughtWorks' projects. Prior to ThoughtWorks, Josh served as a Senior Engineer for Motorola Energy Systems, where he designed and developed real-time testing and analysis software for electrochemical capacitors. He holds a B.A. in Physics and Mathematics, and an almost-M.S. in Chemical Engineering. Josh presented tutorials at JavaCon2000 on "Refactoring" and "Business Objects in J2EE." ThoughtWorks, Inc. is a leading custom e-business application and platform development firm.*

Rebecca Parsons is a Senior Architect for ThoughtWorks, Inc., a leading custom e-business application and platform development firm. While at ThoughtWorks, Rebecca has worked on a large scale leasing system for a financial services subsidiary of a Fortune 100 company. Prior to joining ThoughtWorks, Rebecca was on the faculty of the School of Computer Science at the University of Central Florida where she taught compilers, operating systems, programming languages and computational biology. Rebecca has worked at Los Alamos National Laboratory as well as in industrial positions. She has spoken at both academic and industrial conferences and has served on program committees and editorial review boards for various conferences and publications. Rebecca received a Ph.D. in Computer Science from Rice University in 1992.

60

Planning Agile Projects

Wednesday, Afternoon

Convention Ctr — Room 16

Cara Taber, *ThoughtWorks, Inc.*

Ron Crocker, *Motorola, Inc.*

Most software projects are very poorly planned. They often have a very impressive chart on the wall describing a plan; but that plan is so out of sync with reality that it is more dangerous than useful. The painful truth is that many projects these days are faced with changing requirements, where even half way through a release cycle you still aren't sure what needs to go in the product. In such situations many principles of project planning are undermined, and if this isn't recognized planning falls apart. Despite the uncertainties, agile projects must be planned and can be controlled. In this tutorial we'll look at a simple yet effective technique that can be used to do that. The core of the ideas are based on the planning approach of XP (Extreme Programming) as described in Planning Extreme Programming. However we'll take the techniques and extend the ideas to cover a broader range of agile processes to allow the planning approaches to fit in with processes such as Crystal and RUP. The talk will cover the purpose of planning and the basic principles of XP style planning: four variables, project velocity, yesterday's weather, and division into release and iteration plans. With release planning we'll look at how requirements are chunked up into features (stories), the relationship between features and use cases, how features are estimated, how features are allocated to iterations. In iteration planning we'll look at the break down of features into tasks, allocation of tasks to people, sizing of tasks, and how an iteration is tracked. We'll look at scaling the planning process while sticking to the underlying values, based on experiences running larger projects and global multi-site development.

Attendee Background: no specific background required

Presenters: *Cara Taber has been at ThoughtWorks, Inc., an Internet professional services provider specializing in the delivery of highly strategic B2B e-Commerce solutions, for three and a half years. During that time she has been a developer, designer, analyst and project manager. In her current role as Release Plan Manager, she works closely with the team project manager focusing on planning the iterations and orchestrating the monthly 60-person all team iteration kick-off meeting. She has also published an article with Martin Fowler entitled "An Iteration in the Life of an XP Project," which appeared in the November 2000 issue of the Cutter IT Journal.*

Ron Crocker is a Senior Member of Technical Staff in the Network and Advanced Technology department in Motorola, Inc. where he is responsible for cellular system architecture and design. He has over 15 years of experience with object-oriented technologies, starting as a C++ guinea pig.

61

Designing Small Memory Software: Development Patterns for Systems with Limited Memory

Wednesday, Afternoon

Convention Ctr — Room 25

James Noble, *Computer Science, Victoria University of Wellington, NZ*

Charles Weir, *Penrillian*

Typical OO development techniques assume systems with relatively large memories. Developers working with tight memory requirements need the flexibility and encapsulation that OO can provide, but cannot afford to produce large systems. This tutorial will describe how you can use OO techniques in a memory-constrained environment. Using design patterns and practical examples, this tutorial will teach the most important techniques that successful OO designers use for small memory software. After attending this tutorial, participants will be able to:

- prepare a memory budget;
- design a software architecture and component interfaces to minimize memory use;
- track memory consumption through the development process; and
- tailor user interfaces for small software.

The tutorial balances direct presentations (for overviews and to present each pattern) and case study exercises (to reflect on patterns and see how they can be applied).

Attendee Background: This tutorial targets anyone planning, or involved in, development of OO applications in limited memory. This tutorial is most useful to developers with a year's experience using an OO language and technical team leaders. Experience of memory-limited systems is helpful but not essential.

Presenters: *Dr. James Noble has recently returned home to lecture at the Victoria University of Wellington, New Zealand. While in Sydney, he established the Sydney Patterns Group, the first patterns group in the Southern Hemisphere, and he has extensive experience lecturing, teaching, and mentoring with software design, user interface design, and design patterns.*

Charles Weir is co-founder and managing director of Penrillian, a software house specialising in components for mobile communicators. Charles has more than fifteen years' experience as a software engineer and consultant in OO techniques. He was Symbian technical lead for the Ericsson MC218 communicator project, and software architect for the Psion Series 5 Web Browser. He is co-author of the book, Small Memory Software, and has led many courses and workshops on OO design and implementation.

62

Reflection in Java

Wednesday, Afternoon
Convention Ctr — Room 5
Ira Forman, IBM

Nate Forman, *Liaison Technology*

The use of reflection is an important technique for improving productivity. Reflection facilitates development of programs that are easily adapted to requirement changes. With reflection one can develop software engineering tools that examine or produce code. Reflection facilitates testing and problem determination by facilitating the automation of more tedious tasks. In general, reflection improves the flexibility, extensibility, and reusability of one's code. The Java language contains a highly effective reflection facility. The tutorial explains the concept of reflection, the Java metaobjects (including both introspective and intercessional interfaces), the proxy class, and dynamic compilation and class loading. The limits of Java reflection are addressed in the context of what reflection is capable of in general. In addition, the tutorial demonstrates the efficacy of the Java reflection facility for solving practical problems. Such problems include: program/application testing, generation of code, inspection of code, and use of dynamic class loading in a framework for application extension.

Attendee Background: An attendee must be a competent Java programmer.

Presenters: *Dr. Ira R. Forman works for IBM in Austin. As a member of IBM's Object Technology Products Group, which produced the SOMobjects Toolkit, he worked on the SOM Metaclass Framework. He started working in the area of object-oriented programming in 1984, when he worked at ITT Programming Technology Center. Forman received his Ph.D. in Computer Science from the University of Maryland, where he studied under Harlan Mills. Forman's specialties are object-oriented programming, distributed systems, and object composition. He is the coauthor of two books, Interacting Processes: A Multiparty Approach to Coordinated Distributed Programming and Putting Metaclasses to Work: A New Dimension in Object-Oriented Programming.*

Nate Forman works for Liaison Technology where he designs and programs application frameworks for their products. His specialties are patterns and object-oriented programming. Forman holds a M.S.E. in Software Engineering from the University of Texas at Austin and a B.S. in Computer Science from the College of Engineering at Cornell University.

63

Ruby for the Impatient

*Wednesday, Afternoon**Marriott Hotel — Meeting Room 12*

Dave Thomas,

The Pragmatic Programmers, LLC

Andy Hunt,

The Pragmatic Programmers, LLC

Smalltalk was ahead of its time: we're just entering the decade of the untyped, flexible language. And by all accounts, Ruby could well be the language of that decade. Small, but tremendously expressive, Ruby is finding favor among all kinds of developers. From web applications to numerical simulations at NASA, Ruby is gaining popularity and mindshare. As a developer, you owe it to yourself to have a look at Ruby. Even if you never write a line of Ruby code, the ideas in the language can greatly improve the way you think about design and the ways you implement your programs. And if you do start writing Ruby, you'll discover the tremendous productivity and readability gains that are possible. In this tutorial, we're offering a fast-track way to learn the language, its libraries, and its philosophy. Ruby is so compact and tidy, we're confident that in just three short hours we'll have you reading and writing Ruby like an old-timer.

Attendee Background: Attendees will be familiar with the concepts of object orientation and programming. Some familiarity with a scripting language such as Perl or Python may help, but is not a requirement. Attendees who program in Smalltalk will find much of Ruby comfortably familiar.

Presenters: *Dave Thomas is prominent in the worldwide Ruby community. He co-authored the first English-language Ruby book, runs two Ruby web sites, manages a Ruby Wiki, and is a frequent contributor to the Ruby mailing lists. He has presented Ruby in Europe and the US, in lectures, and to local user groups. Dave is a partner in The Pragmatic Programmers, a software consultancy, and co-author of The Pragmatic Programmer.*

Andy Hunt is co-author of the best-selling book, The Pragmatic Programmer, the new Programming Ruby, and various articles. Between writing, traveling, woodworking and playing the piano, Andy finds time for his consulting business specializing in agile software development. Andy has been writing software professionally since the early 1980s, and currently based in Raleigh, NC. He is President of the RTP chapter of the ICCA and a member of the ACM and IEEE.

64

Realizing Extreme Programming as a Strategic Weapon for Innovation

*Wednesday, Afternoon
Convention Ctr — Room 23*

Ken Auer, *RoleModel Software, Inc.*

Roy Miller, *RoleModel Software, Inc.*

Extreme Programming seems to be more appropriate in some environments than in others. This tutorial explains why Extreme Programming is particularly suitable for those in a “new product” environment using object-oriented tools and techniques. It shows both business and technical players how they can use this process to effectively address these seemingly conflicting requirements:

- getting an idea to market fast while keeping quality high
- leveraging existing assets while quickly adapting to the changing demands of the market and investors
- making the best use of key people in expanding their market reach while not sacrificing their current market
- building a cohesive team in the midst of constant change
- keeping up with leading technology while still getting current work done,

Included in this tutorial will be a participatory “Extreme Hour” simulation showing how business and technology roles work together in XP to keep development and business moving together toward a common goal at the fastest pace possible.

Attendee Background: The target audience is anyone interested in exploring a new approach to leveraging object-oriented programming, systems, and languages in the development of new software products. The assumption is that attendees will have at least heard of XP and know something about it. However, this is not a prerequisite.

Presenters: *Ken Auer is President, Founder and Master Craftsman of RoleModel Software, Inc. He has been active in the development of object oriented software since 1985. In late 1998, RoleModel Software began building the first Extreme Programming Software Studio based on his vision. This is a place where apprentices, skilled journeyman, and software masters work together in an environment of continuous learning with extremely effective modes of collaboration to produce unusually adaptable and robust software for their clients. He is also the co-author of XP Applied, scheduled for publication by Addison-Wesley in October 2001.*

Roy Miller is a Software Developer at RoleModel Software, Inc. Prior to joining RoleModel, Roy spent six years with Andersen Consulting (now Accenture), most recently as a Project Manager. Roy is a contributing author to IBM's developerWorks Java Zone, has co-authored a book in the Addison Wesley XP Series (XP Applied, scheduled for publication in October 2001), and was a featured panelist at the “Business of XP” fishbowl at XP2001.

65

Advanced Extreme Programming Testing Techniques*Wednesday, Afternoon**Convention Ctr — Room 24**Joseph Pelrine, Daedalos Consulting*

How much testing is enough? Too little? Too much? What do developers need to test? The available Extreme Programming literature differentiates between unit testing and functional testing, and gives unit testing during development a (well-deserved and much-needed) high priority, but fails to address a number of other important aspects of developer testing: GUI testing, performance testing, and packaging/delivery testing, for example. This tutorial will illustrate new techniques such as implementing “skins” for JUnit and SUnit, defining test resources for managing items which remain active over a series of tests (e.g. database connections), and automating or integrating various other tests into JUnit and SUnit. The tutorial will be only partly lecture-based. You are encouraged to present problems (and possible solutions) encountered in your work, which the group will address and attempt to solve. We’d like to give out some diff files for SUnit and JUnit, some TestCase extensions, etc., and either do a few proposed tasks, or sit down and see if we can help each other solve some of our testing problems. In order to do this, we’d like to ask you (if possible) to bring along your laptop with floppy drive, your favorite flavor of Smalltalk or Java, SUnit or JUnit, a power cable or fresh batteries, and other related stuff that you think you might need. Also, bring along some enthusiasm and “looking-for-fun” attitude, and we’re sure to have a ball.

Attendee Background: Since the tutorial is (partly) hands-on, participants should have some experience in both Smalltalk or Java and Extreme Programming.

Presenter: *Joseph Pelrine is an expert Smalltalk programmer with over 12 years extensive OT experience and has worked with Kent Beck, the originator of XP, for a number of years. A former columnist for the Smalltalk Report and noted international speaker, he is currently a senior consultant with Daedalos Consulting in Switzerland. He is coauthor of the book, Mastering ENVY/Developer, recently published by Cambridge University Press.*

66

C++ Idioms

Wednesday, Afternoon

Convention Ctr — Room 13

Jan Christiaan van Winkel, AT Computing

This tutorial will address non trivial C++ programming constructs that experienced programmers use frequently. These constructs are too small to be called patterns, but they are part of the language skills of the proficient C++ programmer. Therefore we call them idioms. Several idioms will be discussed, such as:

Traits. Traits classes are used frequently in the C++ standard. Some libraries are built around traits classes, for example the Boost library (www.boost.org). Using traits, it is possible to get hold of information about a type at compile-time, that will then influence behavior at runtime.

Intermediate objects. These are often used as proxy objects to write something in C++ that the syntax disallows, such as `a[1][2]` where `a` is of class type. They exist only in the expression in which they are used.

Resource management through the “Resource Acquisition is Initialization” idiom. Not only allocated memory has to be returned. If a function needs to change a formatting flag in an ostream, that change has to be undone when the function ends.

After the tutorial the attendee will recognize several idioms, and know when to use them. As with any language, knowing idioms will improve your fluency.

Attendee Background: The attendees are expected to know C++.

Presenter: *JC van Winkel has a B.S. and an M.S. in computer science (the M.S. from the Vrije Universiteit Amsterdam). He works at AT Computing, a small courseware and consulting firm in Nijmegen, the Netherlands. There he teaches UNIX and UNIX-related subjects, including C and C++. Except for 1995, J.C. van Winkel has presented tutorials at all OOPSLAs since 1993. He is the Dutch representative in the ISO C++ standardization committee SC22/WG21.*

67

Patterns and Techniques for Developing Performance Effective Enterprise Java Beans

Wednesday, Afternoon

Marriott Hotel — Meeting Room 1

Matjaz B. Juric, Ph.D., Assistant Professor

In the tutorial we will present patterns and techniques for the design and implementation of performance effective Enterprise Java Beans (EJB) components. We will substantiate our discussion with real-world examples and performance measurements for different scenarios. We will look at the basic facts regarding EJBs and performance, explain the underlying concepts, discuss the remote method invocations, fine and coarse grained interfaces, take a look at the value objects and input validation on the client side, learn what is the throughput problem, present the advantages of the facade pattern, discuss the dependent objects, take a look at the instance management algorithms in terms of performance, show, how to manage persistence, transactions, concurrency and how to avoid deadlocks, learn how to lazy load enterprise beans and reuse resources, discuss the advantages of smart stubs and show how to accelerate marshaling, learn how to tune the performance when deploying EJBs, give practical guidelines for achieving scalability, etc. We will focus on the performance relevant changes in the EJB 2.0 specification, particularly on local interfaces, home methods, new container managed persistence schema, and relationships. Sound design alone is not sufficient for good performance. There are performance differences hidden in the application servers as well. Therefore we will present performance measurement results with different application servers.

Attendee Background: Participants should be familiar with OO concepts, distributed component models, Java language, Enterprise Java Beans (EJB) and possibly with Java 2 Enterprise Edition.

Presenter: *Matjaz B. Juric holds a Ph.D. in computer and information science. Currently he is an Assistant Professor at University of Maribor, Faculty of Electrical Engineering and Computer Science. His Ph.D. work has received an award from the Slovenian IEEE Section and he participated in the OOPSLA Doctoral Symposium in 1999. He received several awards for articles and an award for his B.Sc. work (from Slovenian Society for Informatics). His research area covers distributed systems and object technology, with special emphasis on Java, distributed object systems (EJB, RMI, RMI-IIOP, CORBA, COM+), component development, and performance. He has been involved in performance analysis and optimization by the development of RMI-IIOP, an integral part of Java 2 platform, in cooperation with IBM Java Technology Centre, Hursley, UK. Juric has published more than 140 publications, and twelve original scientific papers. He has published a chapter in the book, More Java Gems, (Cambridge University Press) and has written a chapter on performance in the upcoming Wrox, Professional EJB Development book.*

68**Pair Programming: Experience the Difference***Wednesday, Afternoon**Convention Ctr — Room 15**Laurie Williams,**North Carolina State University**Robert Kessler,**University of Utah*

Pair programming is emerging as an important technique for developing higher quality code, faster. With pair programming, two software developers work on one computer, collaborating on the same design, algorithm, code, or test. This tutorial examines pair programming research results and anecdotal experiences of programmers who have transitioned to pair programming. It will discuss what works and what doesn't and will also explain techniques for fostering support in making a transition to pair programming, support from management, and support from peers. Hands-on activities will be used to demonstrate pair programming benefits. Participants will experience the difference between working alone and working in pairs. They will understand the research results that show pair programming works, learn how to pair program, what not to do when pairing, and how to transition to pair programming.

Attendee Background: This tutorial is targeted toward software developers and technical software development managers who are interested in transitioning to pair programming.

Presenters: *Dr. Laurie Williams is an assistant professor at North Carolina State University. In 2000, she completed her dissertation which demonstrated statistically that pair programmers were able to produce higher quality products in essentially half the time when compared to individual programmers. Prior to her recent academic career, Laurie worked at IBM for nine years. Laurie and Bob are collaborating on a book entitled, Pair Programming Illuminated, to be published by Addison-Wesley in 2002.*

Dr. Robert Kessler is a Professor of Computer Science and served as the last chairman of the University of Utah, Department of Computer Science (the department is now known as the "School of Computing"). He has founded several companies and served on the board of directors of others. Bob is an award-winning instructor having recently received the 2000 College of Engineering, Outstanding Teaching Award and the 2001 University of Utah, Distinguished Teaching Award.

69

Objects vs. The Web

Wednesday, Afternoon
Convention Ctr — Room 14

Alan Knight,
Cincom Systems of Canada
Naci Dai,
BEA Systems Inc.

Web development is the cool new paradigm. How do we stop ourselves from forgetting the lessons of previous paradigms and just hacking our way through? The web's core mechanisms lend themselves all too easily to cut-and-paste re-use, ad-hoc scripts, direct-to-database code, and fragmented business logic. In the name of time-to-market, too many of us abandon what we know and take the path of least resistance. There are capable and articulate people telling us this is exactly what we should be doing. We believe, that the lessons of objects do apply to the web. If we apply them wisely we can have good time-to-market for version 1.0, ship version 2.0 successfully, and improve the user experience along the way. But it's not obvious how to apply these lessons. What, if anything, is MVC for the web? What are the architectural layers? How do we support multiple channels within these layers, when presentation differences can easily creep into the domain logic? What about EJB? This tutorial surveys current web technologies with an emphasis on OO usage, provides best practices and examples from Java and Smalltalk, discusses myths and truths about components, and describes architecture and development practices that support good practices.

Attendee Background: Attendees should have a reasonable understanding of OO development. Experience with web development is helpful, although a basic familiarity with terms is adequate.

Presenters: Alan Knight works on Smalltalk web tools for Cincom Systems. Prior to that he was chief architect of the TOPLink family of object-relational mapping products with The Object People and WebGain. He has spoken extensively at conferences including OOPSLA, Smalltalk Solutions and Java One, and is co-author of the book, Mastering ENVY/Developer. He can be reached at knight@acm.org.

Naci Dai is an educator for BEA Systems Inc., prior to that he was the director of distributed computing with The Object People. He teaches object technology, design-patterns, and distributed computing. He leads and mentors web development projects for Fortune 500 companies. He has developed the distributed computing curriculum and services. He has a background in applied-engineering and computational physics. He has received his Ph.D. from Carleton University. He can be reached as nacidai@acm.org.

70

OPEN: A Flexible OO/CBD Process for Software-Intensive Systems Development*Wednesday, Afternoon**Marriott Hotel — Meeting Room 2*Brian Henderson-Sellers,
University of Technology, Sydney

The increased complexity associated with large-scale software-intensive systems development requires an increase in the sophistication of the methodology utilized. Following a general discussion on the value of processes, one specific OO/CBD example, OPEN (Object-oriented Process, Environment and Notation) is described in detail. Emphasis will be placed upon the need for flexibility of processes and how they can be constructed and configured to individual circumstances. Finally, some advice on how to transition to OO/CBD and deploy this process for the first time will be given. Objective: The tutorial objective is to present and understand the need for flexibility in process and how this can lead to an organizationally or project specific process instance using the OPEN Process Framework as an example. A secondary goal is to discuss how transition to an OO environment can be accomplished.

Attendee Background: Fully conversant with basic OO terminology and the need for a full lifecycle process. Experience with OO methodologies is advantageous. Those who would benefit most, would include project managers, systems developers, analysts and designers.

Presenter: *Brian Henderson-Sellers is Director of the Centre for Object Technology Applications and Research and Professor of Information Systems at University of Technology, Sydney (UTS). He is author of nine books on object technology and is well-known for his work in OO methodologies (MOSES, COMMA and OPEN) and in OO metrics. In 1999, he was voted number 3 in the Who's Who of Object Technology (Handbook of Object Technology, CRC Press, Appendix N). He is currently a member of the Review Panel for the OMG's Software Process Engineering Model (SPEM) standards initiative.*

NOTES:

TUTORIALS

WORKSHOPS

WORKSHOPS

Chair: Peter F. Sweeney,
IBM T.J. Watson Research Center

The workshop program for OOPSLA 2001 includes a large number of workshops that spans a wide range of relevant and timely topics. Workshops are intensive collaborative sessions where groups of object technologists meet to surface, discuss, and attempt to solve challenging problems facing their field. Workshops also provide the opportunity for representatives of a technical community to coordinate efforts, and to establish collective plans of action.

To ensure a sufficiently small group for effective interaction, workshop attendance is managed by the organizers of each workshop. Prospective attendees are required (typically) to submit a position paper outlining their opinions on an aspect of the workshop's topic. Participants have been chosen based on the relevance of their position paper to the workshop theme. Workshop presentations are at the discretion of the organizers but all attendees are expected to contribute to the discussion. After the workshop, the organizers are responsible for reporting results to the object community via a short summary that will appear on the workshop's Web site.

Workshops are full day events that occur on either of the first two days of the conference. Workshop organizers and participants must register at the conference for at least the day of their workshop. Workshop participants must check-in at the registration desk before attending the workshop, preferable the evening before. The workshops are categorized below in tracks for ease of selection; individual workshops may be found in multiple tracks. For more information on a particular workshop, please visit its web page.

WORKSHOPS BY DATE

Sunday, October 14, 2001

1	Feyerabend - Redefining Computing	Marriott Hotel — Salon B
2	Refining the Practices of Extreme Programming	Marriott Hotel — Florida Salon I
3	Agile Software Development Methodologies: Raising the Floor or Lowering the Ceiling	Marriott Hotel — Florida Salon II
4	Human Issues of Agile Processes	Marriott Hotel — Florida Salon III
5	Fifth Workshop on Pedagogies and Tools for Assimilating Object-Oriented Concepts	Marriott Hotel — Meeting Room 2
6	Advanced Separation of Concerns	Marriott Hotel — Meeting Room 6
7	Specification and Verification of Component-Based Systems	Marriott Hotel — Meeting Room 1
8	Testing Enterprise JavaBeans	Marriott Hotel — Meeting Room 3
9	Managing Variability in Domain Engineering Using OO Technology	Marriott Hotel — Meeting Room 13
10	Architectural Patterns for Wireless Computing	Marriott Hotel — Salon C
11	Towards Patterns and Pattern Languages for OO Distributed Real-time and Embedded Systems	Marriott Hotel — Salon D
12	Semantics of Enterprise Integration	Marriott Hotel — Meeting Room 5
13	Third Workshop on Best-practices for Business Rules Design and Implementation	Marriott Hotel — Meeting Room 7
14	Workshop on C++ Template Programming	Convention Ctr — Room 5-6
15	Parallel/High-Performance Object-Oriented Scientific Computing	Marriott Hotel — Meeting Room 4
16	Domain-Specific Visual Languages	Marriott Hotel — Meeting Room 10
17	Objects, XML, and Databases	Convention Ctr — Room 3

WORKSHOPS

Monday, October 15, 2001

- 17 Objects, XML, and Databases
- 18 Agile Processes Workshop
- 19 Object-Oriented Web Services
- 20 Software Archeology: Understanding Large Systems

CHANGE IN SCHEDULE

- 21 Patterns and Techniques for Designing Object-Oriented Mobile Wireless Systems
- 22 Beyond Design: Patterns (mis)used
- 23 The Three-Tier Architecture Pattern Language
- 24 Engineering Complex Object-Oriented Solutions for Evolution
- 25 Tenth OOPSLA Workshop on Behavioral Semantics: Back to Basics
- 26 Business Aligned Software Architectures
- 27 Representing Software Architectures
- 28 Software Visualization
- 29 UML Modeling for Enterprise Distributed Object Computing
- 30 Generative Programming
- 31 First OOPSLA Workshop on Language Mechanisms for Programming Software Components

WORKSHOPS BY DATE

Convention Ctr —
Room 5

Marriott Hotel —
Florida Salon I

Marriott Hotel —
Salon B

Marriott Hotel —
Florida Salon II

*Will now be combined
with Workshop 10 and
held on Sunday,
Marriott Hotel —
Salon C.*

Convention Ctr —
Room 6

Marriott Hotel —
Florida Salon III

Marriott Hotel —
Meeting Room 2

Marriott Hotel —
Meeting Room 3

Marriott Hotel —
Meeting Room 7

Marriott Hotel —
Meeting Room 13

Marriott Hotel —
Meeting Room 4

Marriott Hotel —
Meeting Room 5

Marriott Hotel —
Meeting Room 11

Marriott Hotel —
Meeting Room 6

WORKSHOPS BY TRACK

Agile Methods

The workshops in this track explore issues about agile, lightweight software development methodologies.

- 2 Refining the Practices of Extreme Programming
- 3 Agile Software Development Methodologies: Raising the Floor or Lowering the Ceiling
- 4 Human Issues of Agile Processes
- 18 Agile Processes Workshop

Architecture

The workshops in this track explore many aspects of software architecture including enterprise applications, architectural patterns, and the specification and design of architectures.

- 9 Managing Variability in Domain Engineering Using OO Technology
- 10 Architectural Patterns for Wireless Computing
- 12 Semantics of Enterprise Integration
- 13 Third Workshop on Best-practices for Business Rules Design and Implementation
- 20 Software Archeology: Understanding Large Systems
- 23 The Three-Tier Architecture Pattern Language
- 24 Engineering Complex Object-Oriented Solutions for Evolution
- 25 Tenth OOPSLA Workshop on Behavioral Semantics: Back to Basics
- 26 Business Aligned Software Architectures
- 27 Representing Software Architectures
- 28 Software Visualization
- 29 UML Modeling for Enterprise Distributed Object Computing

Components

Component technology holds the promise of plug-and-play software and a new way to assemble applications. The workshops in this track focus on the design, implementation, verification, and testing of components.

- 7 Specification and Verification of Component-Based Systems
- 8 Testing Enterprise JavaBeans
- 9 Managing Variability in Domain Engineering using OO Technology
- 12 Semantics of Enterprise Integration
- 23 The Three-Tier Architecture Pattern Language
- 24 Engineering Complex Object-Oriented Solutions for Evolution
- 29 UML Modeling for Enterprise Distributed Object Computing
- 31 First OOPSLA Workshop on Language Mechanisms for Programming Software Components

Concurrency

The workshops in this track explore concurrency as it applies to wireless computing, distributed real-time and embedded systems, and scientific computing.

- 10 Architectural Patterns for Wireless Computing
- 11 Towards Patterns and Pattern Languages for OO Distributed Real-time and Embedded Systems
- 15 Parallel/High-Performance Object-Oriented Scientific Computing

Emerging Technologies

Emerging technologies are those new technologies that have the potential to have a big impact on object technology. The emerging technologies that the workshops in this track explore include wireless computing, web services, and new ways to use patterns.

- 1 Feyerabend - Redefining Computing
- 6 Advanced Separation of Concerns
- 9 Managing Variability in Domain Engineering using OO Technology
- 10 Architectural Patterns for Wireless Computing
- 12 Semantics of Enterprise Integration
- 16 Domain-Specific Visual Languages
- 17 Objects, XML, and Databases
- 19 Object-Oriented Web Services
- 22 Beyond Design: Patterns (mis)used
- 24 Engineering Complex Object-Oriented Solutions for Evolution
- 30 Generative Programming

Fundamentals

The workshops in this track explore fundamental issues underlying object-oriented technology. Some of those issues that are explored include reinventing computing, investigating object behavior visually, and determining the ways that programmers understand large systems quickly.

- 1 Feyerabend - Redefining Computing
- 7 Specification and Verification of Component-Based Systems
- 20 Software Archeology: Understanding Large Systems
- 28 Software Visualization

Internet Technologies

Although the impact of internet technologies on object technology is enormous, these technologies have been around long enough to no longer be categorized as emerging. The workshops in this track explore issues of integrating internet technologies into an enterprise, the next generation of web services, and how to engineer e-Commerce solutions.

- 17 Objects, XML, and Databases
- 19 Object-Oriented Web Services
- 24 Engineering Complex Object-Oriented Solutions for Evolution

Java Technologies

The workshops in this track explore focus on Enterprise JavaBeans: how to test them and the issues that arise when they are used as a framework for three-tier architectures.

- 8 Testing Enterprise JavaBeans
- 23 The Three-Tier Architecture Pattern Language

Languages

The workshops in this track explore the linguistic support that is needed to support component, generative, modular, and scientific programming with object technology.

- 6 Advanced Separation of Concerns
- 14 Workshop on C++ Template Programming
- 15 Parallel/High-Performance Object-Oriented Scientific Computing
- 16 Domain-Specific Visual Languages
- 30 Generative Programming
- 31 First OOPSLA Workshop on Language Mechanisms for Programming Software Components

Meta-level and Reflective

- 16 Domain-Specific Visual Languages
- 25 Tenth OOPSLA Workshop on Behavioral Semantics: Back to Basics

Middleware

Middleware is the layer of software that sits on top of an operating system and provides run-time services to applications. The workshops in this track explore the run-time support that is required for distributed real-time and embedded systems, wireless computing, and the integration of emerging technologies.

- 10 Architectural Patterns for Wireless Computing
- 11 Towards Patterns and Pattern Languages for OO Distributed Real-time and Embedded Systems
- 13 Third Workshop on Best-practices for Business Rules Design and Implementation
- 17 Objects, XML, and Databases
- 19 Object-Oriented Web Services
- 24 Engineering Complex Object-Oriented Solutions for Evolution

Patterns

The workshops in this track explore the codification of problems and their solutions for a wide range of activities and look for new ways to use this codification.

- 11 Towards Patterns and Pattern Languages for OO Distributed Real-time and Embedded Systems
- 12 Semantics of Enterprise Integration
- 13 Third Workshop on Best-practices for Business Rules Design and Implementation
- 22 Beyond Design: Patterns (mis)used
- 23 The Three-Tier Architecture Pattern Language
- 25 Tenth OOPSLA Workshop on Behavioral Semantics: Back to Basics

People and Project Management

The workshops in this track are concerned with the human aspects of object technology.

- 4 Human Issues of Agile Processes
- 5 Fifth Workshop on Pedagogies and Tools for Assimilating Object Oriented Concepts
- 25 Tenth OOPSLA Workshop on Behavioral Semantics: Back to Basics

Reengineering and Refactoring

- 6 Advanced Separation of Concerns

Requirements Analysis

- 25 Tenth OOPSLA Workshop on Behavioral Semantics: Back to Basics

Small and/or Mobile

The workshops in this track explore embedded and wireless systems.

- 10 Architectural Patterns for Wireless Computing
- 11 Towards Patterns and Pattern Languages for OO Distributed Real-time and Embedded Systems

Testing

- 8 Testing Enterprise JavaBeans

UI and Usability

- 28 Software Visualization

UML

- 29 UML Modeling for Enterprise Distributed Object Computing

**Feyerabend - Redefining Computing**

Sunday
Marriott Hotel — Salon B

Organizers:

Richard Gabriel, *Sun Microsystems, Inc.*
(rpg@dreamsongs.com)

Ron Goldman, *Sun Microsystems, Inc.*

Joseph Bergin, *Pace University*

Fifty years into the First Computing Era some of us in the computing arena have come to realize we've made a false start that can't be fixed, and for us to finally be able to produce lasting, correct, beautiful, usable, scalable, enjoyable software that stands the tests of time and moral human endeavor, we need to start over. Perhaps we'll be able to salvage some of what we've learned from the First Era, but we expect almost everything except the most mathematical fundamentals to be brushed aside.

This workshop is one in a series leading up to an event to reinvent computing. For that event we will be putting together the most diverse group of 75 people possible in a remote but subtly beautiful locale.

The result of the 2-week event will be the first steps toward a road map for massive rebuilding of computing — both as a theoretical endeavor and as a practice — and toward a plan to accomplish it.

In this workshop we will focus on technology ideas inspired from object technology to be used as a starting point for the Feyerabend Project.

URL: <http://www.dreamsongs.com/Feyerabend.html>

Tracks: Emerging Technologies; Fundamentals



Refining the Practices of Extreme Programming

Sunday
Marriott Hotel — Florida Salon I

Organizers:

Ken Auer, *RoleModel Software, Inc.*
(kauer@rolemodelsoft.com)

Ward Cunningham, *Cunningham & Cunningham*

Extreme Programming (XP) has been building some notoriety over the last few years. Although it is still an enigma to many who've heard about it, there are others who are actively doing it. This workshop will be a place where extreme programming practitioners can come together to challenge and encourage each other as they share what they've been doing and where they are going and work together to refine their art. Rather than introduce or defend XP, we will be working together to examine what works and doesn't work and determine how to improve the practice and enlarge the applicable domain. We will be applying some of the XP practices throughout the day to produce a living Best Practices hypertext manual.

URL: <http://www.rolemodelsoft.com/OOPSLA2001/xp>

Track: Agile Methods



Agile Software Development Methodologies: Raising the Floor or Lowering the Ceiling?

Sunday
Marriott Hotel — Florida Salon II

Organizers:

Laurie Williams, *North Carolina State University*
(williams@csc.ncsu.edu)

Steven Fraser, *Nortel Networks*

The use of agile, lightweight software development methodologies can raise the floor for organizations that have historically used ad hoc, “code and fix” processes. Transitioning from an immature to a disciplined process requires a behavioral and cultural change. These developers might find a transition to agile methodologies palatable where otherwise they would find a transition to a heavyweight process unthinkable. Many would agree that it would be beneficial if agile methodologies were adopted by organizations that now operate in “code and fix” mode.

Many organizations on the quest for higher and higher levels of CMM maturity are also considering and transitioning to various agile methodologies. Is this change beneficial to these organizations? Or, are they in fact, lowering the ceiling, and compromising beneficial practices that were previously incorporated? Often, these same organizations are under renewed pressure to react to innovations and market changes at warp speed. Heavyweight processes might prevent them from responding to these changes rapidly enough. Additionally, agile methodologies might allow practitioners from heavyweight organizations to “come out of the closet” and admit they were actually not following their process.

This workshop’s goal is to bring together practitioners who have experiences from both the “raise the floor” and the “lower the ceiling” perspectives and explore the issues involved in an organizations adoption of agile, lightweight software development methodologies.

URL: <http://collaboration.csc.ncsu.edu/agile>

Track: Agile Methods



Human Issues of Agile Processes

Sunday
Marriott Hotel — Florida Salon III

Organizers:

Jens Coldewey, *Coldewey Consulting, Germany* (jens_coldewey@acm.org)

Jutta Eckstein, *Objects in Action, Germany*

Pete McBreen, *McBreen Consulting, Canada*

Alastair Handley, *Pragmatic Software Consulting Services, Canada*

The “Manifesto for Agile Software Development” (<http://www.agilealliance.org/>) states the following important values for agile processes:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

In looking at these values, we find that there are many human issues that pervade the agile software development process. These human issues effect team management as well as client management and the ways in which they collaborate.

The intention of this sequel to last year’s OOPSLA workshop “Deploying Lightweight Processes” is to identify and illuminate these human issues. We are looking for insights that may help people and organizations prepare for the effects of Agile Processes and the cultural changes that they bring, as well as insights into the psychological systems on which Agile Processes are based.

URL: <http://www.coldewey.com/publikationen/conferences/oopsla2001/AgileWorkshop/>

Tracks: Agile Methods; People and Project Management



Fifth Workshop on Pedagogies and Tools for Assimilating Object-Oriented Concepts

Sunday
Marriott Hotel — Meeting Room 2

Organizers:

Jürgen Börstler, *Umeå University, Sweden*
(jubo@cs.umu.se)

Isabel Michiels, *Vrije Universiteit, Brussel, Belgium*

Successfully applying object technology during the software development life cycle requires a thorough understanding of object-oriented concepts. However, learning as well as lecturing these concepts have proven to be very difficult, especially for those who first adopted a procedural way of thinking.

Many OO teaching approaches are limited to teaching a specific object-oriented programming language. Furthermore they often use a strict bottom-up approach and focus on the control structure part of the language. This gives students/trainees the impression that there is a central unit of control in each program, which can be fully controlled by the developer. However, this does not match well with the idea that responsibilities are distributed among (encapsulated) objects.

The goal of this workshop is to share experiences about alternative teaching approaches and tools to improve the teaching and learning of the basic concepts of object technology rather than teaching a specific programming language. Participants will furthermore reflect upon the learning process to identify key requirements for the understanding of the basic concepts and propose feasible orderings of the teaching material.

URL: <http://www.cs.umu.se/~jubo/Meetings/OOPSLA01/>

Track: People and Project Management

**Advanced Separation of Concerns**

Sunday
Marriott Hotel — Meeting Room 6

Organizers:

Kris De Volder, *University of British Columbia, Canada*
(kdvolder@cs.ubc.ca)

Maurice Glandrup, *University of Twente, The Netherlands*

Siobhán Clarke, *Trinity College, Ireland*

Robert Filman, *NASA Ames Research Center*

Separation of concerns is at the core of software engineering. It refers to the ability to identify, encapsulate, and manipulate those parts of software that are relevant to a particular concept, goal, purpose, or issue. Object-orientation has significantly improved our ability to achieve good separation of concerns. Developers can now produce modular implementations of fairly complex systems. However, software complexity has since grown to tremendous proportions. Thus today we face even more challenging development tasks and the next generation of separation of concerns problems are emerging. Recently, many researchers have become aware of this fundamental problem and begun to gain some understanding about its underlying causes. They have proposed potential solutions based on, for example, new, crosscutting modularity mechanisms (aspects) and support for multiple dimensions of overlapping concerns. This workshop is intended to bring together those researchers and practitioners interested in pushing the frontier in this important area.

URL: <http://www.cs.ubc.ca/~kdvolder/Workshops/OOPSLA2001/ASoC.html>

Tracks: Emerging Technologies; Languages; Reengineering and Refactoring

**Specification and Verification of
Component-Based Systems**

Sunday
Marriott Hotel — Meeting Room 1

Organizers:

Dimitra Giannakopoulou, *NASA Ames Research Center*
(dimitra@ptolemy.arc.nasa.gov)

Gary Leavens, *Iowa State University*

Murali Sitaraman, *Clemson University*

This workshop's goal is to explore foundations for applying formal techniques to component-based systems. Specification and reasoning techniques are urgently needed to permit composition of systems from components, for which source code is unavailable. We would like to bring together researchers and practitioners in the areas of component-based software and formal methods, to address these problems. The idea is to focus more of the effort in formal methods on component-based systems. Besides functional behavior, we are also interested in reasoning about concurrency, mechanization and scalability, and performance (time and space). The participants will brainstorm about these and related topics in formal methods for component-based systems to understand both the problems involved and how formal techniques may be useful in solving them. The expected result of the meeting is an outline of collaborative research topics and a list of areas for further exploration.

URL: <http://www.cs.iastate.edu/~leavens/SAVCBS/index.html>

Tracks: Components; Fundamentals

**Testing Enterprise JavaBeans**

Sunday
Marriott Hotel — Meeting Room 3

Organizers:

Oliver Vogel, *Systor, Switzerland*
(oliver.vogel@systor.com)

Markus Voelter, *Mathema, Germany*

Components, in general, and Enterprise Java Beans (EJBs), in particular, are coarse-grained reusable units of software. Yet to be successfully reused, EJBs must exhibit certain quality characteristics, perhaps more so than other software artifacts. One successful way of creating quality software is testing. Extreme programming, and other light weight software development processes, have found that testing is an essential aspect of software development. However, testing EJB is different than testing normal code, because EJBs are executed on the server, inside the container which provides certain infrastructure services. Neither the EJB specification nor current EJB books cover testing in depth. Yet, systems having to fulfill quality of service requirements cannot be built without a clear understanding and adequate development of testing strategies.

This workshop seeks to explore successful techniques for EJB testing.

URL: <http://www.voelter.de/oopsla2001>

Tracks: Components; Java Technologies; Testing



Managing Variability in Domain Engineering using OO Technology

Sunday

Marriott Hotel — Meeting Room 13

Organizers:

Bedir Tekinerdogan, *University of Twente, The Netherlands*
(bedir@cs.utwente.nl)

Mehmet Aksit, *University of Twente, The Netherlands*

Krzysztof Czarnecki, *DaimlerChrysler, Germany*

Sholom Cohen, *Software Engineering Institute, USA*

Domain engineering aims to promote systematic reuse by focusing on the development of assets for families of applications within a given contextual scope instead of the development of single applications for a specific context. The activities of domain engineering require a system-family focus to deal systematically with variability across the family of applications.

Variability is a quality factor expressing the ease with which existing components can be adapted. Variability is both an important concern in domain engineering and in object-oriented (OO) software development. The basic goal of variability is to quantify reuse and ease of developing a family of applications.

OO technology encompasses a diversity of variability at different abstraction levels. At the analysis and design level, OO technology provides useful techniques such as use cases, scenarios, framework design and the Unified Modeling Language. At the implementation level, OO supports variability with language features such as polymorphism, inheritance and subtyping.

This workshop's goal is to identify the OO technologies that are useful for managing and expressing variability in domain engineering.

URL: <http://trese.cs.utwente.nl/oopsla01-variabilityWS/>

Tracks: Architecture; Components; Emerging Technologies

**Architecture Patterns for Wireless Computing**

Sunday
Marriott Hotel — Salon C

Organizers:

Steffen Schaefer, *IBM*
(steffens@acm.org)

Steve Marney, *EDS*

Tony Willis, *Fundamo*

Kari Parkkinen, *dna Finland*

In some respects, wireless solutions are not very different from standard e-Business systems and are implemented with the same or similar technology.

But clearly, the standard Web 'Request/Response' model, e.g., as implemented through WAP is not enough. Offline-work, messaging, and data replication are just a few examples where Wireless Computing goes beyond 'internet for small screens'. As in other e-Business systems, Java, C++ and XML are often used for implementation at the server side, as well as on the pervasive device. Using such well known technology as a base, this workshop aims at identifying reference architectures and design patterns that are specific, or at least typical for Wireless Computing.

The goal is to exchange experience about architecture issues for Wireless Computing. We will aim to produce tangible results, e.g., architecture or design patterns for specific issues, and try to reach consensus on approaches for resolving problems and identify strategies.

URL: <http://www.dnainland.fi/oopsla>

Tracks: Architecture; Concurrency; Emerging Technologies; Middleware; Small and/or Mobile



Towards Patterns and Pattern Languages for OO Distributed Real-time and Embedded Systems

*Sunday
Marriott Hotel — Salon D*

Organizers:

Michael Kircher, *Siemens*
(Michael.Kircher@mchp.siemens.de)

Prashant Jain, *Siemens*

Douglas Schmidt, *University of California, Irvine*

Angelo Corsaro, *Washington University, St. Louis*

Distributed real-time and embedded (DRE) systems are becoming ubiquitous, not just in the aerospace domain, but also in telecommunications, automotive, and process control. As software replaces functionality traditionally provided by hardware, better software development technologies are needed to ensure end-to-end system quality of service properties and to control lifecycle costs.

Increasingly, object-oriented (OO) technologies are being considered as the underpinning for mission-critical DRE systems. Because OO technologies were not initially intended for the time and space constraints of DRE systems, researchers and practitioners often struggle to apply OO techniques effectively so their tasks become easier, not harder.

What are common patterns and practices that can be applied to overcome many recurring challenges and constraints related to developing DRE systems? The goal of this workshop is to bring together researchers and practitioners from the field of DRE systems to begin documenting a common “pattern language” for DRE systems.

URL: <http://www.cs.wustl.edu/~mk1/RealTimePatterns>

Tracks: Concurrency; Middleware; Patterns; Small and/or Mobile

**Semantics of Enterprise Integration***Sunday**Marriott Hotel — Meeting Room 5***Organizers:**

Sergio de Cesare, *Brunel University, UK*
(sergio.decesare@brunel.ac.uk)

Mark Lycett, *Brunel University, UK*

Paul Allen, *Computer Associates, UK*

Grant Holland, *Sun Microsystems, USA*

Tony Morgan, *Unisys, UK*

The growth of the Internet and the expansion of e-commerce systems have accentuated pre-existing problems related with systems interoperability and the definition of common semantics throughout business organizations. Consequently, an increasing amount of cost and effort is dedicated to the integration of systems and applications. The issues involved, however, are not just related to infrastructure and technology. Indeed fundamental problems exist around the understanding of what 'concepts' are shared, how they relate and what mechanisms should be adopted to allow systems to communicate and interoperate at levels.

This workshop welcomes the participation of academics, researchers and practitioners interested in approaches aimed at achieving the integration of enterprise systems.

URL: http://www.site-project.org.uk/events/sei2001_workshop/

Tracks: Architecture; Components; Emerging Technologies; Patterns



Third Workshop on Best-practices for Business Rules Design and Implementation

Sunday
Marriott Hotel — Meeting Room 7

Organizers:

Ali Arsanjani, *IBM*
(arsanjan@us.ibm.com)

Joseph Yoder, *The Refactory, Inc.*

Alan Abrahams, *Cambridge University*

Reza Razavi, *Université Pierre et Marie Curie*

Jeff Oakes, *The Hartford*

Keith Levi, *Maharishi University of Management*

Rules are the heart of business logic. Businesses implement their information systems upon the pillars of business rules. Yet, writing scalable, robust, and adaptable business rules for business objects in today's e-business and n-tier architectures is a challenge! The goal of this workshop is to categorize and capture, in pattern format, the best and successful practices relating to the design and implementation of business rules within the context of business objects and components.

URL: www.mum.edu/cs_dept/aarsanjani/business-rules/oopsla2001

Tracks: Architecture; Middleware; Patterns

**Workshop on C++ Template Programming**

Sunday
Convention Ctr — Room 5-6

Organizers:

Yannis Smaragdakis, *Georgia Institute of Technology, USA*
(yannis@cc.gatech.edu)

Nicolai Josuttis, *Author and Consultant, Germany*

One of the most exciting research areas in C++ focuses on the use of templates, especially as a means to support generic and generative programming. A number of powerful, flexible, and useful techniques have been developed but these efforts appear isolated and are only known to a few experts. This workshop will gather the community of people interested in C++ template programming. The goal is dual: first, to increase the visibility of information likely to be helpful to other workers in the field of C++ template programming; second, to clearly establish, and hopefully advance, the state-of-the-art in C++ template programming.

URL: <http://www.oonumerics.org/tmpw01/>

Track: Languages

**Parallel/High-Performance Object-Oriented Scientific Computing**

Sunday
Marriott Hotel — Meeting Room 4

Organizers:

Kei Davis, *Los Alamos National Laboratory*
(kei@lanl.gov)

Joerg Striegnitz, *John von Neumann Institute for Computing*

Andrew Lumsdaine, *University of Notre Dame*

Bernd Mohr, *John von Neumann Institute for Computing*

Todd Veldhuizen, *Indiana University*

Jeremy Siek, *University of Notre Dame*

While object-oriented programming has been embraced in industry, particularly in the form of C++ and to an increasing extent Java, its acceptance by the parallel scientific programming community is still tentative. In this domain performance is invariably of paramount importance, where even the transition from FORTRAN 77 to C is incomplete, primarily because of performance loss. On the other hand, three factors together practically dictate the use of language features that provide better paradigms for abstraction: increasingly complex simulations, numerical algorithms, and hardware (e.g. deep memory hierarchies, numbers of processors, communication, and I/O).

This workshop seeks to bring together practitioners and researchers in this field to demonstrate or propose libraries, techniques, idioms, or general approaches to OO software development particularly relevant to high performance scientific computing; also their expected, realized, or intrinsic limitations or shortcomings.

URL: <http://www.c3.lanl.gov/~poosc01/>

Tracks: Concurrency; Languages

**Domain-Specific Visual Languages**

Sunday
Marriott Hotel — Meeting Room 10

Organizers:

Juha-Pekka Tolvanen, *MetaCase Consulting and University of Jyväskylä*
(jpt@metacase.com)

Kalle Lyytinen, *Case Western Reserve University*

Jeff Gray, *Vanderbilt University*

Steven Kelly, *MetaCase Consulting*

In a domain-specific visual language, the models are made up of elements representing things that are part of the domain world, not the code world. The language follows the domain abstractions and semantics, allowing developers to perceive themselves as working directly with domain concepts. Thus, the language raises the abstraction to the problem domain, making development work faster and easier.

Some of the issues that we would like to see addressed in this workshop are:

- Industry/academic experience reports describing success/failure in implementing and using domain-specific languages/tools
- Separation of concerns and the application of new modularity technologies (e.g., aspect-oriented/subject-oriented) to domain-specific languages
- Approaches to identify constructs for domain-specific languages
- Novel approaches for code generation from domain-specific models
- Issues of support/maintenance for systems built with DSL's
- Evolution of languages in accordance with domain
- Metamodeling frameworks and languages
- Tools for supporting domain-specific languages
- Specific domains where this technology can be most productive in the future (e.g. visual DSL's to describe aspects of embedded systems, product family, systems with multiple implementation platforms)

URL: <http://www.isis.vanderbilt.edu/OOPSLA2K1>

Tracks: Meta; Emerging Technologies; Languages

**17 Objects, XML, and Databases**

Sunday
Convention Ctr — Room 3

Monday
Convention Ctr — Room 5

Organizers:

Akmal Chaudhri, *Informix Labs, USA*
(akmal@soi.city.ac.uk)

Awais Rashid, *Lancaster University, UK*

During the past few years, there has been a considerable interest and growth in a number of new and emerging technologies, such as XML. For many organizations already using object-orientation with database management systems, XML data adds a new dimension that brings considerable flexibility and promise, but also adds new uncertainties and issues as to how to effectively manage that data. The recent trend towards XML servers, native XML databases and support for XML in existing relational databases is a testimony to the importance of this issue for the vendor community as well. However, whilst these commercial offerings are becoming available, what are the issues that need to be explored to effectively use Object-Orientation, XML and Database Systems? The goal of this workshop is to bring together academics, practitioners, users and vendors to explore the issues that these communities currently face in the integration and effective use of these technologies.

URL: <http://www.soi.city.ac.uk/~akmal/oopsla01.dir/01-workshop.html>

Tracks: Emerging Technologies; Internet Technologies; Middleware

**Agile Processes Workshop**

Monday
Marriott Hotel — Florida Salon I

Organizers:

Mike Beedle, *e-Architects, Inc.*
(beedlem@e-architects.com)

Ron Jeffries, *Object Mentor, Inc.*

Ken Schwaber, *Advanced Development Methods*

Robert Martin, *Object Mentor, Inc.*

Jim Highsmith, *Information Architects, Inc.*

Dave Thomas, *The Pragmatic Programmers*

The goal of this workshop is to understand and contrast the different agile processes in existence today such as: XP, SCRUM, ASD, Crystal, FDD, Pragmatic and DSDM. A detailed examination of each agile process will be conducted to understand its practices. Case studies for each process will be provided as empirical evidence. Similarities, differences, synergies and misfits among different agile processes practices will be examined both from practice and theory.

URL: <http://www.agilealliance.org/oopsla2001>

Track: Agile Methods

**Object-Oriented Web Services**

Monday
Marriott Hotel — Salon B

Organizers:

Brent Hailpern, *IBM T.J. Watson Research Center*
(bth@watson.ibm.com)

Don Batory, *University of Texas at Austin*

Richard Gabriel, *Sun Microsystems, Inc.*

Aditya Mathur, *Purdue*

Peri Tarr, *IBM T.J. Watson Research Center*

Sanjiva Weerawarana, *IBM T.J. Watson Research Center*

XML has started a revolution on the web: from static formatted content to dynamic self-defining information with real semantics. The addition of semantics is only the first step to providing real applications, known as “web services,” to internet users. The first generation of web-service infrastructure is in development, allowing one web service to issue a request to another and to register/describe/find a service to use (e.g., SOAP, WSDL, UDDI).

Eventually, web services will become electronic utilities that are delivered to end users over the internet, representing a critical new application domain in electronic commerce. Like traditional utilities, such as electricity, web services will be metered and customers will pay for their use. The terms of use (called service level agreements or SLAs) of the services will include functionality, performance, and reliability. The need to dynamically monitor and control SLAs and to respond to changing needs and resources, imposes challenging requirements on the design, development, deployment, and evolution of web services.

This workshop explore issues in, and new technologies and methodologies to support the engineering and deployment of, web services.

URL: <http://www.research.ibm.com/people/b/bth/OOWS2001.html>

Tracks: Emerging Technologies; Internet Technologies; Middleware

**Software Archeology:
Understanding Large Systems**

Monday
Marriott Hotel — Florida Salon II

Organizers:

Brian Marick, *Testing Foundations*
(marick@visibleworkings.com)

Ward Cunningham, *Cunningham and Cunningham*

Andrew Hunt, *The Pragmatic Programmers*

Dave Thomas, *The Pragmatic Programmers*

How do you come to grips with 1,000,000 lines of code right away?

Programmers are often given a large system they've not seen before, built by people they don't know, touched by many people since, documented sketchily if at all. They're told to improve it. Their task might be to fix a bug, add a feature, or complete a refactoring. They are under time pressure, so they need to minimize the total time spent learning and the time spent improving.

In this workshop, we will share techniques and approaches for understanding enough about a lot of code in not much time. We are concerned not just with speed, but also with confidence: how can you know you've made an improvement, not made the system worse?

URL: <http://www.visibleworkings.com/archeology>

Tracks: Architecture; Fundamentals

*CHANGE IN SCHEDULE***Patterns and Techniques for Designing
Object-Oriented Mobile Wireless Systems**

Will now be combined with Workshop 10 and held on Sunday, Marriott Hotel — Salon C.

Organizers:

Kirthika Parmeswaran, *Telcordia Technologies*
(kirthika@research.telcordia.com)

Andrew Campbell, *Columbia University*

Ravi Jain, *Telcordia Technologies*

Mobile applications and services are exploding in the market today. However, typically value-added applications have been developed as additions to the most basic application of all, which is mobile wireless voice and data access, or developed as specialized vertical services. Despite the proliferation of mobile wireless services being introduced in the marketplace, it is not yet clear which will be the “killer apps” and for which market segments and in which regions. Therefore, it is critical that frameworks be built to enable rapid creation and deployment of mobile wireless services leveraging not only reusable components but also design and deployment patterns, including the adaptation and optimization patterns essential in mobile wireless environments. The goal of this workshop is to identify the challenges as well as discuss and document the design patterns and OO techniques used for building mobile wireless systems. An aim of the workshop is to provide a forum for a stimulating discussion between the participants on the requirements and features of common APIs, both between mobile clients and servers, and between services and the mobile application framework.

URL: http://www.argreenhouse.com/society/mobility_workshop.shtml

Tracks: Components; Emerging Technologies; Patterns; Small and/or Mobile

22**Beyond Design: Patterns (mis)used**

Monday
Convention Ctr — Room 6

Organizers:

Christa Schwanninger, *Siemens, Germany*
(christa.schwanninger@mchp.siemens.de)

Elisa Baniassad, *University of British Columbia, Canada*

Gail Murphy, *University of British Columbia, Canada*

Vera Seidel, *Siemens, Germany*

In the past seven years, a strong pattern community has been built, as is reflected in a number of conferences. This community has largely focused on the writing and using of patterns. But design patterns are important for reasons beyond capturing and transferring information about recurring problems and their solutions. For instance pattern information can aid reverse engineering tasks, or it can be used to determine useful language improvements or extensions. Currently, research centered on such concepts occurs at the periphery of the pattern community, and hence there has been limited opportunity for researchers interested in other aspects of pattern research to interact with the pattern community. Our goal is to diversify and expand the pattern community with these additional views, to find points of collaboration and to share insights about patterns.

URL: <http://www.schwanninger.com/OOPSLA2001/>

Tracks: Emerging Technologies; Patterns

**The Three-Tier Architecture Pattern Language**

Monday
Marriott Hotel — Florida Salon III

Organizers:

Prashant Jain, *Siemens*
(Prashant.Jain@mchp.siemens.de)

Michael Kircher, *Siemens*

Kirthika Parameswaran, *Telecordia Applied Research Labs*

Three-tier applications have gained increasing acceptance and popularity in the software industry. A large number of distributed systems and enterprise applications are built using three-tier architectures. Component technologies such as Enterprise Java Beans (EJB) and CORBA Component Model (CCM) make use of three-tier architectures to provide frameworks for component development and deployment. The goal of this workshop is to discover and document the common patterns and pattern languages among the architectures of three-tier systems. The patterns, interwoven as they are, will be used to formulate a pattern language for three-tier systems. The workshop will be especially valuable to system architects as well as software developers attending OOPSLA. The pattern language formulated at the workshop will provide a powerful and effective means of documenting and describing architectures of three-tier systems.

URL: <http://www.cs.wustl.edu/~pjain/ThreeTierPatterns>

Tracks: Architecture; Components; Java Technologies; Patterns

**Engineering Complex Object-Oriented Solutions
for Evolution**

Monday
Marriott Hotel — Meeting Room 2

Organizers:

Siobhán Clarke, *Trinity College, Ireland*
(siobhan.clarke@cs.tcd.ie)

Simon Dobson, *Conceptual Drift Ltd., Ireland*

Vinny Cahill, *Trinity College, Ireland*

Robert Filman, *NASA Ames Research Center*

Complex object-oriented solutions present challenges with respect to evolution and adaptability. Yet evolution and adaptability are necessary requirements to support business-to-business processes across the internet. Advances are being made in software engineering support for complex development, such as component-based development, and advanced ways of separating concerns. An interesting suite of object-oriented solutions, that builds on both mature infrastructures (such as CORBA), and on newer platforms (such as Java and agent systems), have been used to developing the first generation of business-to-business e-commerce solutions.

Nevertheless, if more complex e-commerce solutions are to be engineered, rather than simply built, what new set of issues must be confronted? This workshop is intended to bring together researchers and practitioners to discuss the issues inherent in engineering complex solutions for evolution and adaptability. The goal is to identify those issues which directly affect the successful deployment and evolution of a complex system and which are poorly supported by current methods, and to explore directions for research and development to address these omissions.

URL: www.dsg.cs.tcd.ie/ecoose/oopsla2001/

Tracks: Architecture; Components; Emerging Technologies; Internet Technologies; Middleware

**Tenth OOPSLA Workshop on Behavioral Semantics: Back to Basics**

Monday
Marriott Hotel — Meeting Room 3

Organizers:

Haim Kilov, *Independent Consultant*
(haimk@acm.org)

Kenneth Baclawski, *Northeastern University*

The purpose of this workshop is to improve the understanding and use of precise semantics in object-oriented (OO) specifications and designs. The goal is to be a focal point that brings together theoreticians and practitioners to share their experience with making semantics precise, clear, concise and explicit in OO business specifications, business designs, and system specifications and designs. Papers can range from academic research to industrial “war stories.” This is the tenth anniversary of this OOPSLA workshop series. The specific emphasis this year is to return to the basics to recapture insights and ideas that might otherwise slip into oblivion (or at best be reinvented), while also looking forward to the future of the field.

URL: <http://www.ccs.neu.edu/home/kenb/oopsla2001>

Tracks: Architecture; Meta; Patterns; People and Project Management; Requirements Analysis

26**Business Aligned Software Architectures**

Monday
Marriott Hotel — Meeting Room 7

Organizers:

Vladimir Bacvanski, *InferData Corporation*
(vladimir@inferdata.com)

John Vernon, *JP Morgan Chase*

The goal of IT departments within large enterprises is to deliver technology solutions that support the services and operation of the business, i.e. its business architecture. One way to meet this goal is to have the application architecture specification be traceable to the business architecture specification. The specification of how a component specification will be implemented and deployed, its technical architecture, needs to be traceable to the application architecture. Recently, modeling and component technologies started playing an important role in development of enterprise applications, but modeling and components do not provide sufficient means for development of business aligned software architectures. What is needed is an architecture framework consisting of methods for specifying architectures, including the different views of architecture, guidelines for deciding the levels of abstraction at which architectures will be specified, and principles describing how abstractions are derived from each other.

Such a framework will meet the goals described above.

This workshop's goal is to identify key issues in developing business aligned software architecture and to identify a set of best practices for dealing with introduction of the architecture program in an enterprise.

URL: <http://www.inferdata.com/resources/oopsla2001/bizaligned>

Track: Architecture

**Representing Software Architectures**

Monday
Marriott Hotel — Meeting Room 13

Organizers:

Thad Scheer, *Sphere of Influence*
(tscheer@att.net)

Scott Pringle, *Lockheed Martin*

Chris Kauffman, *Epiphany Software Inc.*

In this workshop, software architects will exchange ideas about how to represent architectural structures and concepts in a commercial setting. The workshop will explore representation techniques that participants have found useful, essential (or irrelevant); and will discuss pitfalls in representing software architectures. The workshop's theme is carried forward from an OOPSLA 2000 birds-of-a-feather session on the same subject. Participants of the workshop will describe their expectations for architectural representations and share their experiences regarding what works and what doesn't. The workshop's focus is on establishing "practical" guidance, not on developing formal systems of complete specification. The objective of this workshop is to enhance our understanding of how software architecture is "best" represented (i.e. communicated) for commercial projects.

URL: http://home.att.net/~tscheer/OOPSLA_01/oopsla01.htm

Track: Architecture

**Software Visualization**

Monday
Marriott Hotel — Meeting Room 4

Organizers:

Wim De Pauw, *IBM T.J. Watson Research Center*
(wim@us.ibm.com)

Steven Reiss, *Brown University*

Gary Sevitsky, *IBM T.J. Watson Research Center*

This workshop looks at current work in the area of software visualization with an emphasis on software understanding through visualization. It explores new visualization techniques and their application to software problems at every phase of the development lifecycle. It also covers work on frameworks for gathering and analyzing data for software visualization, software visualization systems, the use of visualization in combination with other analysis techniques, and experiments and experiences with software visualization. In addition to providing an overview of current research in the area, it provides a forum for discussion and cooperation among researchers in this and related areas.

URL: <http://www.research.ibm.com/people/w/wim/oopsla2001>

Tracks: Architecture; Fundamentals; UI and Usability

**UML Modeling for Enterprise Distributed Object Computing**

Monday
Marriott Hotel — Meeting Room 5

Organizers:

Fred Cummins, *EDS*
(fred.cummins@eds.com)

Arne Berre, *SINTEF*

Cory Casanave, *Data Access Technologies*

Yoshi Nagase, *Tech-Arts and Consortium for Business Objects Promotion*

David Zenie, *Iona Technologies*

Sandy Tyndale-Biscoe, *Open-IT, Ltd.*

Hiroshi Miyazaki, *Fujitsu*

UML (Unified Modeling Language) has been widely accepted by the industry as the lingua franca for modeling object-oriented systems. In March of 1999, the Object Management Group (OMG) issued a Request for Proposals for a specification to define a UML Profile for Enterprise Distributed Object Computing (EDOC). Since that time, a number of companies have participated in the development of proposals that are in the process of converging toward a single specification. Preliminary proposals are currently available. In addition, OMG has introduced the Model Driven Architecture (MDA). MDA will redirect OMG efforts toward specifications expressed in UML and mapped to multiple technologies. The UML Profile for EDOC is expected to become a key element of MDA, providing the specification meta-model for enterprise systems. A final proposal of the UML Profile for EDOC is expected to be presented to the OMG in July of 2001.

The purpose of this workshop is to engage participation outside OMG and the specification submitter group to discuss how the UML profile might be refined, extended and incorporated in development tools, as well as to discuss related work that could leverage this profile.

URL: www.enterprise-component.com/oopsla2001

Tracks: Architecture; Components, UML

**Generative Programming***Monday**Marriott Hotel — Meeting Room 11***Organizers:**

Krzysztof Czarnecki,
DaimlerChrysler Research and Technology, Germany
(czarnecki@acm.org)

Greg Butler, *Concordia University, Canada*

Craig Cleveland, *Independent software consultant, USA*

Kris De Volder, *University of British Columbia, Canada*

Lutz Dominica, *Siemens Corporate Technology, Germany*

Ulrich Eisenecker, *University of Applied Sciences Kaiserslautern, Germany*

Yannis Smaragdakis, *Georgia Tech, USA*

This workshop invites practitioners, researchers, academics, and students to discuss experiences with generative techniques and the role of these techniques in object-oriented development.

- synergy between object-oriented technology, components, and generative techniques
- styles of generators (application generators, generators based on XML technologies, template languages (e.g., JSP), template metaprogramming, transformational systems, intentional languages, aspects, subjects, etc.), particularly their uses and limitations;
- generation of code artifacts, such as application logic, UIs, database schemas, and middleware integration;
- generation of non-code artifacts such as test cases, documentation, tutorials, and help systems;
- capturing configuration knowledge, for example, in DSLs, and extensible languages;
- influence on software architecture (e.g., building and customizing frameworks and applying patterns);
- testing generic and generative models; and
- industrial applications of generative technology.

This workshop's goal is to provide a forum for participants to share their experiences, to assess the state-of-the-art and the state-of-the-practice generative programming techniques, to consolidate successful techniques, and to identify the most promising application areas and open issues for future work.

URL: www.generative-programming.org/oopsla01-workshop.html

Tracks: Emerging Technologies; Languages

**First OOPSLA Workshop on Language Mechanisms for Programming Software Components**

Monday
Marriott Hotel — Meeting Room 6

Organizers:

Vugranam C. Sreedhar, *IBM T.J. Watson Research Center*
(sreedhar@watson.ibm.com)

David H. Lorenz, *Northeastern University*

Although there are many models for component-based software development, most of these models are based on sets of standards and frameworks (APIs), and are implemented on top of a mainstream object-oriented programming language. Very little research has been done in understanding and promoting the key concepts in component-oriented programming; that is, identifying what exactly is component-oriented programming and what language mechanisms exist for component-oriented style of programming. This workshop intends to bring together researchers, practitioners, and implementers to present their experience in component programming in a forum that will allow them to collaborate and exchange ideas. This workshop's goal is to address two questions:

- 1 What are the key ingredients of component-oriented programming?
- 2 How to express these key ingredients in a component-oriented programming language?

URL: <http://www.ccs.neu.edu/home/lorenz/oopsla2001/>

Tracks: Components; Languages

NOTES:

WORKSHOPS

DEMONSTRATIONS

DEMONSTRATIONS

Chair: Peter Kriens,
aQute B.V.

Demonstrations provide an opportunity to describe works-in-progress, to display applications of object-oriented technology, and to share unique and interesting technical aspects of object-oriented products, tools, or systems. Demonstrations are intended to be two-way interactions between the authors presenting state-of-the-art technology, and the audience, which has the opportunity to share ideas, interact with the authors in a small scale venue, and learn techniques used in developing innovative and high quality software. This year we have a selection of 20 demonstrations that cover a broad range from XML to scrolling, from nuclear claims systems to personal information agents.

Demonstration sessions last for 45 minutes. Attendance at demonstrations is on a first come, space available basis until the session capacity is reached. In perusing the schedule, please note that all demonstrations are offered both on Tuesday the 16th and Wednesday the 17th; this should make your conference scheduling plans easier.

Demonstrations take place in rooms 18, 19, 20, and 21, in the Tampa Convention Center.

Tuesday and Wednesday – 10:30 am - 11:15 am

Convention Ctr — Room 18

1 Developing and Evolving Java Applications using Coordination Contracts

João Gouveia, *Oblog Software SA*
 Georgios Koutsoukos, *Oblog Software SA*
 Michel Wermelinger, *Universidade Nova de Lisboa*
 José Luiz Fiadeiro, *Universidade de Lisboa*
 Luís Andrade, *ATX Software SA*

Coordination contracts are a modeling primitive to facilitate the evolution of software systems. Coordination contracts encourage the separation of computation from coordination aspects. They encapsulate the way components interact, and capture the business rules or other volatile rules that govern the application. System evolution consists of adding and removing contracts (i.e., changing the rules) between components. The interactions specified by the contract are superposed on the functionality provided by the components, without the programmer having to modify them and without components being aware of contracts. New contracts may be added anytime (even runtime).

This demonstration presents our Coordination Development Environment (CDE), written in Java, that supports the development of Java applications with coordination contracts. It allows to write contracts, and to register Java classes (components) for coordination. The tool is then used to generate the Java code for adapting those components and for implementing the contract semantics based on a micro-architecture. The CDE also includes an animation tool, in which the runtime behavior of contracts and their participants can be observed using sequence diagrams, thus allowing testing of the deployed application.

Convention Ctr — Room 19

2 Information Integration and Visualization (IIV) with Java and XML

Kurt Derr, *Idaho National Engineering and Environmental Laboratory*
 Kenneth Rohde, *Idaho National Engineering and Environmental Laboratory*

Information systems are typically tailored to specific domains, such as management, financial, geographic, project management, inventory control, military planning, etc. This makes it difficult for a single application to view related information from disparate sources. For example, information from different applications can only be viewed with that application. IIV solves the problem of viewing related information from disparate sources. It stitches together islands of information at the level of the desktop, with no user programming required. An extensive repertoire of visualizations and data manipulation tools is available. IVBQ is a client-server application written in Java.

This demonstration will show the integration and visualization of related information from multiple sources - XML, live network connections, relational databases, files, and GIS sources.

DEMONSTRATIONS

Tuesday and Wednesday – 10:30 am - 11:15 am (cont.)

Convention Ctr — Room 20

3 Round-trip Objects in Smalltalk

Daniel Antion, *American Nuclear Insurers*

The demonstrated system was developed by American Nuclear Insurers to deploy an emergency claims system in response to emergency situations at commercial nuclear power plants. A single host is responsible for claim validation and persistence. Remote nodes can be individual users or host computers. The system communicates by sending Smalltalk objects over TCP/IP in a host/remote mode, mixed mode (some remote hosts and some remote users) or in a multi-tier configuration. Smalltalk's support for creating and manipulating object based transactions and the availability of a socket framework was a key reason for selecting VisualAge. Also important was the high degree of support for object reuse and the ease of testing during development.

The demo includes a discussion of the transaction processing, the underlying communication framework (built on top of TotallyObject's - Socket Set), the reasons for building the system in Smalltalk, the testing methods and the test results.

Convention Ctr — Room 21

4 Asset Locator - an Enterprise Development Resource Management System

Avi Yaeli, *IBM Haifa Research Staff Member*

Gabi Zodik, *Manager, Programming Languages and Environments*

Dept., *IBM Haifa Research Lab*

Iftach Ragoler, *IBM Haifa Research Staff Member*

Alex Akilov, *IBM Haifa Research Staff Member*

Enterprise Development Resource Management (EDRM) is a new paradigm for the management of development resources (e.g. code, documentation, design documents, test suites) in the enterprise. EDRM leverages the valuable information that exists in development resources and systems to provide developers with search, reuse, collaboration and impact analysis capabilities. Asset Locator - a low cost/maintenance scalable solution is a first step towards achieving EDRM. Asset Locator uses a set of autonomous crawlers to crawl into the enterprise repositories/systems and discover new development resources. A set of domain specific analyzers (e.g., Java, C++, HTML, etc.) analyze the discovered resources and then populate a central repository. This repository serves as the base for global analysis (e.g. impact analysis) as well as the rest of the EDRM capabilities that Asset Locator provides.

This demonstration will show how to exploit the textual information and semantic features that reside in OO resources (e.g. inheritance relationship in Java) to support automatic categorization of OO resources into predefined domain taxonomies, semantic navigation of the enterprise repository and improved component/code reuse via a semantic search server.

Tuesday and Wednesday – 11:15 am - 12:00 pm

Convention Ctr — Room 18

5 ComposeJ: Aspect-Oriented Programming with Composition Filters and Java

Lodewijk Bergmans, *University of Twente, The Netherlands*

Mehmet Aksit, *University of Twente, The Netherlands*

Richard van de Stadt, *University of Twente, The Netherlands*

Hans Wichman, *TRIMM bv, The Netherlands*

Patricio Salinas, *University of Twente, The Netherlands*

Luigi Savarese, *University of Twente, The Netherlands*

Composition Filters address a number of modeling obstacles that conventional object-oriented techniques cannot address or can only solve with poorly maintainable designs. Aspect-oriented programming (AOP) deals with features crosscutting multiple methods and/or classes. These features typically results in tangled code that mixes and/or duplicates code that deals with different issues. The Composition Filters approach to AOP integrates aspects and classes, retains strong encapsulation, and supports composability. ComposeJ is a tool that takes Java classes and (separate) composition filters specifications, and transforms the Java classes by inlining so that they implement the behavior as specified in the composition filters specification.

The demonstration will introduce a modeling problem involving crosscutting behavior, and explain why it cannot be implemented in Java effectively. A solution will be shown by extending Java classes with a composition filters specification. The GUI of ComposeJ is used to generate Java files that implement the composition filters specification. The generated code will show that this code is a tangled implementation of the crosscutting behavior.

Convention Ctr — Room 19

6 AspectJ™: the Language and Support Tools

Erik Hilsdale, *Xerox PARC*

Gregor Kiczales, *University of British Columbia*

Aspect-oriented programming (AOP) is a technique for improving separation of concerns in software design and implementation. AOP works by providing explicit mechanisms for capturing the structure of crosscutting concerns. When implemented in a non-aspect-oriented fashion, the code for these concerns typically becomes spread out across entire programs. AspectJ controls such code-tangling and makes the underlying concerns more apparent, making programs easier to develop and maintain. AspectJ is a seamless aspect-oriented extension to Java™. It can be used to cleanly modularize the crosscutting structure of concerns such as exception handling, multi-object protocols, synchronization, performance optimizations, and resource sharing.

This demonstration illustrates what the AspectJ language can do and it shows the tools that support developing programs with this language. AspectJ is freely available at <http://www.aspectj.org>

DEMONSTRATIONS

Tuesday and Wednesday – 11:15 am - 12:00 pm (cont.)

Convention Ctr — Room 20

7 Comparing Two Approaches for Mapping Objects to Relational Databases

Arash Farzin, *Nebras Informatics Co.*

Navid Khosravi, *Nebras Informatics Co.*

Sasan Dashtinezhad, *Nebras Informatics Co.*

Nebras Persistence Service is a product that can be used to add persistency to object-oriented systems. It provides a tool that automatically adds the needed persistency methods and attributes to persistent classes in Rational Rose models. A run-time service facilitates the persistency between the Application Layer and Storage Layer. The persistency is achieved by mapping attributes to relational database tables. Different strategies can be followed with this mapping.

The demonstration will show the two mapping approaches and they will be compared for their benefits and drawbacks in different circumstances.

Convention Ctr — Room 21

8 GME: A Reflective Environment for Domain-Specific Modeling

Akos Ledeczki, *Institute for Software Integrated Systems, Vanderbilt University*

The Generic Modeling Environment (GME 2000) is a metaprogrammable graphical editor supporting the design, analysis and synthesis of complex, software-intensive systems in diverse engineering fields. GME 2000 has a component based architecture using MS COM technology and is implemented in C++. The Core component exposes the domain-specific language specifications, provides access to the models and publishes its modification events. All the other components (GUI, browser, OCL constraint manager, software generators, etc.) are built around the Core. Model persistence is supported via standard database technology and XML import/export functionality.

The demonstration will focus on an example domain, an integrated simulation framework for embedded systems. The UML and OCL based metamodels specifying the domain-specific visual modeling language will be shown. From these metamodels, the domain-specific environment is generated automatically. An example application will be demonstrated; an Automatic Target Recognition system, including its complex models and the automatically generated Matlab simulation and C implementation.

Lunch break

Tuesday and Wednesday – 1:30 pm - 2:15 pm

Convention Ctr — Room 18

9 NetBeans Metrics Module

Thomas Ball, *SunLabs, Sun Microsystems, Inc.*

Software metrics are often used by project managers to help understand a project's design, complexity, and robustness. Unfortunately, much of this information is fed back to the project's engineers when it is too late in a release cycle to be of much use.

This demonstration shows a NetBeans (Forte for Java) module that adds software metrics properties to NetBeans' Explorer. This allows metrics to be easily monitored while classes are designed and developed. Sun Labs is investigating whether this early feedback will improve Java applications and reduce their development and maintenance costs.

Convention Ctr — Room 19

10 Tengger Support of Design-through-Code [UML through Java] for Feature-Oriented Separation of Concerns

William Harrison, *IBM T.J. Watson Research*

Charles Barton, *IBM T.J. Watson Research*

Mukund Raghavachari, *IBM T.J. Watson Research*

Tools for composition-based software building, whether based on Aspect-Oriented Programming or more general approaches to Advanced Separation of Concerns, are generally focused on the weaving or composition of code in OO languages. Tengger is a tool that supports the expression of the design as separated concerns, generating Java code from sets of UML class diagrams. Each concern is expressed by its own UML class diagrams and by Java implementations of the UML-declared methods. The UML design is at a high-level, not specifically tailored to Java's language constructs. Tengger composes the UML designs and produces Java code implementing the result. Hyper/J composes the generated Java code with the developer's code to produce a complete application.

This demonstration will present the concepts behind feature-based design and development as supported by Tengger. A small application is carried through the design using Rational Rose, the code-generation using Tengger, and the code composition with Hyper/J. The focus will be on the design-through-code process, on the Java coding style used in the developer-written code, and on the ways in which Tengger and Hyper/J fit into the process to yield the final application.

DEMONSTRATIONS

Tuesday and Wednesday – 1:30 pm - 2:15 pm (cont.)

Convention Ctr — Room 20

11 Explicit Programming: Improving the Design Vocabulary of your Program

Avi Bryant, *University of British Columbia*

Andrew Catton, *University of British Columbia*

Kris De Volder, *University of British Columbia*

Gail Murphy, *University of British Columbia*

Object-oriented systems are frequently built around conventions and design patterns that can only be indirectly captured in source code. Explicit programming provides a solution to extend the vocabulary of existing, general purpose languages, allowing developers to encode design decisions in a concrete, encapsulated, and reusable way.

ELIDE supports explicit programming in Java and allows developers to introduce new, parameterized modifiers into the Java language, at the class, field, method, and block levels. Each of these modifiers triggers a series of transformations on the program source code. The transformations for a modifier are defined by the developer in a Java class, which operates on a simple representation of the original source code.

The demonstration will show how to use the tool to greatly simplify the usage of existing frameworks such as JUnit and standards such as JavaBeans, by introducing test-specific and bean-specific modifiers.; ELIDE will be used to refactor repetitive code and design structures into custom transformations through lightweight, iterative application of the tool.

Convention Ctr — Room 21

12 Demonstration of a UML-Based Architecture Description Language

David Boyle, *ALLTEL Communications Inc.*

Sandeep Gupta, *ALLTEL Information Services*

Floyd Berus, *ALLTEL Information Services*

Charles Letner, *ALLTEL Information Services*

The notations and tools in widespread use for the purpose of designing and documenting enterprise IT architecture focus primarily on application architecture. As a result they do not adequately provide for the technical management of an enterprise comprised of many systems. An Architecture Description Language (ADL) provides a mechanism for modeling enterprises from a logical level, through system function, integration, hosting, and networking. The goal is to provide an integrated set of models that take the viewer from a business process down to the specific devices on which the software resides that addresses the business concern. This ADL notation is separated into Views and Layers. These Views and Layers represent various abstractions of both the business and the technical domains.

This demonstration will show ADL and how to extend two of the popular UML-based modeling tools and create a CASE environment that supports enterprise architecture design.

Tuesday and Wednesday – 2:15 - 3:00 pm

Convention Ctr — Room 18

13 Integrating Refactoring Support into a Java IDE

Dirk Baeumer, *Object Technology International*

Erich Gamma, *Object Technology International*

Adam Kiezun, *Object Technology International*

Most existing refactoring tools are add-ons to existing Java development environments. The Eclipse Java tooling is a new open source development environment that comes with integrated refactoring support for Java. Refactoring support is built on top of an infrastructure that was designed with a focus on refactoring. A database containing structural information about Java projects provides means for efficient searching and reference updates. Refactoring is seamlessly integrated into the UI and available from all views. And last, the environment is open for developers to extend the set of available refactorings.

The demonstration shows various refactorings and their integration into the Eclipse Java Tooling. An in-depth overview of the technical aspects of the refactoring tool is given by presenting the implementation of a particular refactoring.

Convention Ctr — Room 19

14 Web Testing Using an Object-Relational Database System

Robert Cook, *Yamacraw Professor, Dept. Math and CS, Georgia Southern University*

The demonstration illustrates how the XDB object-relational database system implements a web student testing server. An example web site is <http://bcook.cs.gasou.edu/eclass>. A unique aspect of the testing server is its extensibility and the test format, which is executable. As a result, question parameters and answers are computed on-the-fly. The Answer-Vector (AV) database object supports the flexible grading of tests, which might have a wide variety of question types. The AV encodes each question's format (multiple choice, numeric answer, etc.) and a possible list of answers in format in which answers are graded by performing a case-insensitive comparison to a list containing the correct answer and some common misspellings of the correct answer.

The demonstration will include the dynamic generation of computed tests as well as a review of different question and answer specifications. The goal is to foster a discussion of the applicability of computer-based testing to a problem domain that ranges from true/false questions to essays.

DEMONSTRATIONS

Tuesday and Wednesday – 2:15 - 3:00 pm (cont.)

Convention Ctr — Room 20

15 Traversing the `SemiPositionableStream`: Millions of Objects in a Scrolling List

James Foster, *Systems Architect, Banner Health System*

A scrolling list is generally impractical for displaying more than a few hundred items, particularly if the items are in a multi-user database. While databases typically provide an interface to access sorted records as a stream (e.g., with a cursor), they generally do not provide: (1) the information needed to display the scroll bar (or thumb) that identifies the relative position in the list; (2) an interface to jump to a relative position in the list; or (3) an interface to move backwards. This demonstration will discuss `SemiPositionableStream`, an extension to Brokat's `GemStone/S` Object Server that identifies an approximate position in a list, allows approximate positioning, and moves both forward and backward. By filling the visible portion of the scrolling list and adjusting the scroll bar to reflect the position of those items in the underlying list, the scrolling list can appear to present any number of items. The database techniques described will be applicable to non-user-interface situations and to other database implementations as well.

Convention Ctr — Room 21

16 An OO Client Server System for Concept Mapping and Lexical Navigation

James Cooper, *IBM T J Watson Research Center, jwcnmr@watson.ibm.com*

The purpose of the system is the discovery of related concepts that might never have been obvious from reading any single group of articles. The system can be used for query refinement and focus of overly general queries. The system works as follows. A server manages a search engine and index, document text mining and a database which holds document metadata, including linguistic relations between discovered multiword terms. A client program displays "concept maps," or lexical relations between terms using a graphical layout system. This program is based on JSP and Java and communicates to the server with SOAP over HTTP.

The system will be demonstrated on a set of prescription drug data, showing what can be learned by mining these documents.

Tuesday and Wednesday – 3:30 pm - 4:15 pm

Convention Ctr — Room 18

17 ArchJava: Connecting Software Architecture to Implementation

Jonathan Aldrich, *University of Washington*

Craig Chambers, *University of Washington*

David Notkin, *University of Washington*

ArchJava is a small extension of Java that integrates software architecture smoothly into Java code. Software architecture describes the structure of a system, and is useful for design, program understanding, and formal analysis. However, existing systems' implementations are developed separately from their architecture and therefore may not conform to that architecture, causing confusion, violating architectural properties, and inhibiting software evolution. ArchJava is unique in that it keeps code and architecture coherent without restricting common programming idioms.

This demonstration will show how to apply ArchJava to create both static and dynamically changing software architectures. It will demonstrate the downloadable ArchJava compiler and tools, and explain how ArchJava's type system keeps implementation code and architecture coherent. ArchJava will be applied to two moderate-size (12-15,000 line) applications. The demo will include a brief comparison to related technologies and will actively search feedback from the audience.

Convention Ctr — Room 19

18 CommonRules : A Framework For Rapid Development of Object Oriented Applications Using Business Logic Modules.

Benjamin Grosf, *Sloan School of Management, MIT*

Jeff Kephart, *IBM T.J. Watson Research Center*

Even with modern OO languages like Java and C++, it still requires skilled programmers to create an application. Java beans technology provides a way to connect objects together but is limited in expressing the logical relationships. Modifications and maintenance will also be a time consuming and difficult, especially when original programmers are not around. CommonRules introduces a platform and methodology allowing rapid application building based on small business logic modules expressed as sets of logic based, human language like rules. Larger applications can be constructed by merging small modules of business logic dynamically. The framework completely hides the programming details and exposes only the business logic as human language like syntax.

The demonstration will show how to build an application using the CommonRules framework and modify the application subsequently to illustrate the various features mentioned above. It will also show how to construct an application by merging small pieces of business logic modules dynamically and resolve conflicts among the small business logic modules.

DEMONSTRATIONS

Tuesday and Wednesday – 3:30 pm - 4:15 pm (cont.)

Convention Ctr — Room 20

19 Supporting Distributed Extreme Programming

Frank Maurer, *Department of Computer Science, University of Calgary*
Sebastien Martel, *Department of Computer Science, University of Calgary*

Extreme programming (XP) is arguably improving the productivity of small, co-located software development teams. However, there is the XP constraint of co-location. This can be overcome by introducing a process-support environment (called MILOS) that helps software development teams to maintain XP practices in a distributed setting. MILOS supports project coordination, process support, object versioning, skill model, information routing, team communication, pair programming, resource pools, long term organizational learning and more. To achieve this, MILOS was built using EJB and various other OO components.

The demonstration will consist of an example scenario of simplified extreme programming development cycle using the MILOS system.

Convention Ctr — Room 21

20 Personal Information Agent

Dominik Kuroпка, *University of Muenster (Germany)*

Information overflow is one of the greatest challenges for information focused professions today. Personal Information Agent (PI-Agent) is an adaptive agent-based information filtering system which automatically scans news-sources for presumably important messages. Different types of information sources are possible (e.g. web-pages, e-mail, newsgroups). PI-Agent works for one user and has ideas about what is important for its user. The user's past evaluations are used for learning. The PI-Agent prototype is implemented in Java. Persistency of agent's knowledge base is assured by a relational database system (PostgreSQL). For communication with the user a web-based user interface is implemented using an Apache webserver and Java-Servlets.

The audience will experience how the agent works in cooperation with its user. Furthermore the system architecture, the functionality and the implementation details are presented.

NOTES:

DEMONSTRATIONS

POSTERS

POSTERS

Chair: Peri Tarr,
IBM T. J. Watson Research Center

Poster Session: Monday, 5:30 pm - 7:30 pm, Tampa Convention Center

Poster Display: Tuesday-Friday, Exhibit Hall

The poster session is an informal and highly interactive session that gives OOPSLA attendees the opportunity to engage one another in discussions about relevant, ongoing work and critical issues in key areas, and also to learn about work in areas with which they wish to become familiar. Researchers and practitioners describe their work in progress or to elaborate on work presented in other conference forums, while all OOPSLA participants can obtain rapid, low-cost introductions to interesting work and technologies in object-oriented software engineering and provide input and feedback directly to the authors. Posters are displayed throughout the conference, so they also provide attendees with the opportunity to learn about interesting work at their leisure, and they enable interactions to occur throughout the conference. Posters cover the same interest areas as the Technical Papers and Practitioner Reports.

The OOPSLA 2001 Posters program begins with a special session at the Welcome Reception on Monday evening, where all posters will be on display and the authors will be present to meet with attendees and discuss their work. For the remainder of the conference, the posters will be displayed in the Exhibit Hall, so that OOPSLA participants can view them at their convenience, and 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, and the poster session will be organized to facilitate such interactions, and to assist attendees in finding information about topics that interest them.

1 Domain-Specific Visual Languages

Juha-Pekka Tolvanen, *Jyväskylä University*

Jeff Gray, *Vanderbilt University/ISIS*

Steven Kelly, *MetaCase Consulting*

Kalle Lyytinen, *Case Western Reserve University*

An upward shift in abstraction leads to a corresponding increase in productivity. In the past, this has occurred when programming languages have evolved towards a higher level of abstraction. Today, domain-specific visual languages provide a viable solution for continuing to raise the level of abstraction beyond coding.

2 A Framework for Performance Monitoring and Modeling of Enterprise Java™ Beans Applications

Adrian Mos, *Dublin City University*

John Murphy, *Dublin City University*

We present a methodology that helps developers and system integrators understand and potentially correct the performance issues of an EJB-based system at an object-oriented level. Using this methodology, they will also be able to predict the behavior of their system when different user loads are applied.

3 Flow- and Context-insensitive Points-to Analyses for Java™: Extensions and Evaluation

Donglin Liang, *Georgia Institute of Technology*
 Maikel Pennings, *Georgia Institute of Technology*
 Mary Jean Harrold, *Georgia Institute of Technology*

This poster presents extensions to Steensgaard's and Andersen's algorithms to handle Java™ features. The poster also presents the empirical studies that evaluate the effectiveness of handling Java™ features using alternative approaches and the impact of the points-to information provided by these two algorithms on client analyses that use the information.

4 Why Java™ is not Suitable for Object-Oriented Frameworks

Dragos Manolescu, *Applied Reasoning Systems Corporation*
 and the *University of Kansas*
 Adrian Kunzle, *Skillgames*

Many business applications involve Java™ and object-oriented frameworks. Several characteristics of Java™ conflict with some key features of frameworks. These conflicts force the creation of “work-arounds” by developers. We show several examples that illustrate the tensions that exist between Java™ and object-oriented frameworks, and discuss how we solved them.

5 Workshop on UML Profile for Enterprise Distributed Object Computing

Fred Cummins, *EDS bluesphere*
 Arne Berre, *SINTEF, Distributed Information Systems*
 Cory Casanave, *Data Access Technologies*
 Hiroshi Miyazaki, *Fujitsu*
 Yoshi Nagase, *Technologic Arts Inc./Consortium for Business Object Promotion*
 David Zenie, *Iona Technologies*
 Sandy Tyndale-Biscoe, *Open-IT, Ltd.*

OMG has a tentative specification for the modeling of recursive components and processes for distributed computing. The purpose of this workshop is to discuss how this UML profile might be refined, extended and incorporated in development tools, as well as to discuss related work that could leverage this profile.

6 Architectural Patterns for Usability

Len Bass, *Software Engineering Institute/Carnegie Mellon University*
 Bonnie E. John, *Software Engineering Institute/Carnegie Mellon University*

Facets of usability that require architectural support such as cancellation, undo, and progress bars are identified. For each facet, an architectural pattern is described that supports the achievement of the facet. Facets of usability that require architectural support are difficult to add after completing the initial design of a system and, hence, it is critical to identify these facets prior to initial system design.

7 Must Java™ Development Be So Slow?

Albrecht Wöß, *J. Kepler University*

Starting a new VM each time an application is executed forces numerous classes to be loaded multiple times. This overhead significantly slows down the development of Java™ software. So don't do it! We plan to eliminate these bottlenecks with an open Java™ environment where only one VM hosts all applications and classes are loaded only once.

POSTERS

8 Results of the Educators' Symposium: Looking for Abstractions in the Real World

Rebecca Wirfs-Brock, *Wirfs-Brock Associates*
Alan McKean, *Wirfs-Brock Associates*
Jutta Eckstein, *Objects in Action*

This poster presents some results of the Educators' Symposium activity session on object-oriented modeling: Looking for Abstractions in a Concrete World: Candidates, Responsibilities, and Collaborations. CRC has traditionally stood for "Classes-Responsibilities-Collaborators." But classes are too concrete. Jumping to classes too soon results in a lifeless design.

9 Specification and Verification of Component-Based Systems Workshop

Gary T. Leavens, *Iowa State University*
Dimitra Giannakopoulou, *NASA Ames Research Center*
Murali Sitaraman, *Clemson University*

This poster summarizes results from the workshop on Specification and Verification of Component-Based Systems. The workshop's goal is to explore foundations for applying formal methods to component-based systems. The results and future work sections of the poster will be prepared during the workshop.

10 Optimization of Planar Gradient Coil Systems for a Mobile Magnetic Resonance Device by Genetic Algorithms Using Object-Oriented Design Techniques

Hartmut Popella, *RWTH Aachen*
Gerhard Henneberger, *RWTH Aachen*

This poster presents an object-oriented software tool for the optimization of a planar surface gradient coil system for magnetic resonance imaging (MRI) performed by genetic algorithms. The application of UML notation is well suited to the development of a numerical computation tool for the construction of gradient coil systems.

11 Objects, XML and Databases

Akmal Chaudhri, *IBM Informix Labs*
Awais Rashid, *Lancaster University*

The OOPSLA 2001 Workshop on Objects, XML and Databases will explore various issues in the integration and effective use of these technologies. This poster summarizes and highlights solutions to a set of issues that will be extracted from the workshop presentations and discussions.

12 Performance Monitoring of Large Global Distributed Systems

Doris Ressimann, *De Montfort University*
Amelia Platt, *De Montfort University*
Steve Rumsby, *De Montfort University*

Technological advances, particularly in object-based software design and communication networks, have laid the foundations for building more sophisticated and flexible distributed systems. Distributed execution of applications demands that the performance of executing applications be monitored. This poster presents two architectures that support performance monitoring.

13 MultiJava: Open Classes and Multiple Dispatch for Java™

Curtis Clifton, *Iowa State University*
 Gary Leavens, *Iowa State University*
 Craig Chambers, *University of Washington*
 Todd Millstein, *University of Washington*

MultiJava is a backward-compatible extension to Java™ that includes open classes and symmetric multiple dispatch. It is the first full-scale programming language to support these features while retaining modular static type checking and compilation. The implementation of mjc, a MultiJava compiler, validates the language design.

14 Automation of Component Communication in Java™

Alexander Sakharov, *Verizon*

This poster presents a generic solution to the problem of automating the implementation of component communication in Java™. It applies to J2EE components as well. The users specify component communications directly in Java™. A communication adapter is automatically generated from the specifications. This solution takes advantage of component hierarchies.

15 MOO: A Programming Environment that Promotes Feelings of “Being There” or Presence

John Towell, *Carroll College*
 Elizabeth Towell, *Carroll College*

A MOO is a networked, object-oriented, text-based virtual reality. Connected users control the behavior of avatars that interact with the environment. Most people who experience this environment report feelings of “being there” or presence, which has been utilized as a pedagogical tool to teach object-oriented concepts.

16 Evaluation of the Runtime Performance of Control Flow Structures for Dynamic Dispatch in Java™

Olivier Zendra, *INRIA / McGill University*
 Karel Driesen, *INRIA / McGill University*
 Feng Qian, *McGill University*
 Laurie Hendren, *McGill University*

We present an ongoing study of control flow structures in Java™. We run a number of benchmarks on various JVM and hardware platforms, to characterize the performance of these structures when used to simulate dynamic dispatch implementations. Both execution patterns and control structures have a surprisingly large impact on performance.

17 Interdependence of Software Evolution and Development Process Evolution in Agile Methodologies

Christian Wege, *University of Tuebingen*

A software development process can become more successful by analyzing the produced artifacts and feeding the results back into the process. This research will produce techniques for analyzing the artifacts, show how to interpret them and list a collection of experiences made in real world projects.

POSTERS

18 Supporting Distributed Extreme Programming

Frank Maurer, *University of Calgary*
Sebastien Martel, *University of Calgary*

Extreme programming is arguably improving the productivity of small, co-located software development teams. We introduce an approach that overcomes the XP constraint of co-location by introducing a process-support environment, called MILOS, which helps software development teams to maintain XP practices in a distributed setting.

19 The Eclipse Platform Plugin Architecture

Jeff McAffer, *Object Technology International*

The Open Source Eclipse Platform (<http://www.eclipse.org>) has been designed for building integrated development environments (IDEs). In order to focus the generic functionality on something specific, the Eclipse Platform is equipped with a mechanism for discovering, integrating, and running third-party plugins and to host their tool-specific UI.

20 Introducing Patterns (or Any New Idea) into Organizations

Linda Rising, *Independent consultant*
Mary Lynn Manns, *University of North Carolina at Asheville*
Alan O'Callaghan, *De Montfort University*

Many people who have tried to introduce patterns (or any new idea) into organizations have found it to be difficult. This poster presents a pattern language to capture solutions to these challenges. A thumbnail of each pattern will be displayed, as well as ways the patterns have been used.

21 Domain-Specific Pattern Languages

Michael Kircher, *Siemens AG*
Prashant Jain, *Siemens AG*
Kirthika Parameswaran, *Telecordia Technologies*
Douglas Schmidt, *University of California, Irvine*
Angelo Corsaro, *Washington University, St. Louis*

The goal of this poster is to capture the results of two workshops, namely “The Three-Tier Architecture Pattern Language” and “Towards Patterns and Pattern Languages for OO Distributed Real-time and Embedded Systems.” While the domains of the two workshops are quite different, they share a common theme of capturing patterns and pattern languages by identifying the underlying forces.

22 Pedagogies and Tools for Object Oriented Teaching and Learning

Jürgen Börstler, *Umea University*
Isabel Michiels, *Free University of Brussels*

Many OO teaching approaches focus on the control structures part of a specific OO language. Even “object first” approaches do sometimes confuse students by starting out with non-typical examples/exceptions from the rule. We summarize results from workshops on issues in teaching and learning object orientation held at OOPSLA and ECOOP.

23 An Object Oriented Approach for Developing Finite Element SolversDirk van Riesen, *Aachen Institute of Technology (RWTH)*Gerhard Henneberger, *Aachen Institute of Technology (RWTH)*

An object-oriented approach to finite element calculations in electromagnetic, structural dynamics and thermal applications is presented. Special attention is given to the easy implementation of new element types and new formulations. The object-oriented design makes it possible to link in external libraries for specific tasks, like MTL/ITL for the equation system.

24 Transmigration of Object Identity: The Programming Language GilgulPascal Costanza, *University of Bonn*

Gilgul is an extension of Java™ that strictly separates the notions of reference and comparison that are traditionally subsumed in the concept of object identity. This allows for the introduction of new operations that open up new degrees of flexibility by providing a mechanism for unanticipated, dynamic software adaptation.

25 FCL Checker: Detecting Structural Errors in Framework Based DevelopmentDaqing Hou, *University of Alberta*H. James Hoover, *University of Alberta*

Object-oriented frameworks are hard to learn and use. The capability to automatically detect errors occurring at the boundary between frameworks and applications is crucial to mitigate the problem. We introduce the notion of framework constraints and the language FCL to formally specify and check them.

26 Transactions Meet MOM—System Support for Integrating Distributed Object Transactions and Messaging in Java™ and MQ EnvironmentsStefan Tai, *IBM Research*Thomas Mikalsen, *IBM Research*Isabelle Rouvellou, *IBM Research*Stanley Sutton, *IBM Research*

The Dependency-Spheres project explores concepts and middleware system support for distributed transaction processing across object and messaging components. The objective is to enrich standard object middleware and messaging middleware to provide for an increased level of reliability for their use in combination in enterprise systems.

27 Reducing Proof Burden in Object-Oriented VerificationFrancis Tang, *University of Edinburgh*

Formal verification of programs is generally accepted to be laborious and time consuming. The use of verification condition generators can significantly reduce the amount of work required. We demonstrate such an approach for object-oriented programs, using Euclid's algorithm as an example.

POSTERS

28 Java™ Component Development in Jiazzi

Sean McDirmid, *University of Utah*

Matthew Flatt, *University of Utah*

Wilson Hsieh, *University of Utah*

This poster shows how Java™ developers can use Jiazzi to enhance their Java™ code with constructs for large-scale binary components. We show examples that use small source files and development flows to illustrate how to use Jiazzi.

29 Multiple Executable-Filed Java™ Virtual Machine

Satoshi Numata, *Osaka Electro-Communication University*

Hiroataka Uoi, *Osaka Electro-Communication University*

Java™ programs contain safe parts, which are compiled into Java™ machine code, and unsafe parts, which are compiled into native machine code. We propose a new Java™ Virtual Machine composition, which comprises two executable files: one executes Java™ machine code, and the other executes native machine code. We believe this completely separates unsafe parts from safe parts, and it also improves safety and efficiency.

30 A Coordination Methodology and Technology for Agile Businesses

Luís Andrade, *ATX Software SA*

José Luiz Fiadeiro, *Univ. de Lisboa*

João Gouveia, *Oblog Software SA*

Georgios Koutsoukos, *Oblog Software SA*

Michel Wermelinger, *Univ. Nova de Lisboa*

Coordination contracts are a modeling primitive that facilitates the development of systems subject to frequently changing business requirements. This is achieved through non-intrusive superposition of adaptors and connectors, corresponding to volatile business rules, on the components that implement core, stable, services. The poster addresses both methodological and technological support.

31 Concern Space Modeling in Cosmos

Stanley Sutton Jr., *IBM Thomas J. Watson Research Center*

Isabelle Rouvellou, *IBM Thomas J. Watson Research Center*

Cosmos is a schema for modeling software concerns across the life cycle. It defines a metamodel, including concern types, relationships, and predicates, for modeling multidimensional concern spaces. Cosmos allows concerns to be modeled independently of development formalisms, tools, and methods, and it complements and supports advanced separation of concerns technologies.

32 STOOP: The Sable Toolkit for Object-Oriented Profiling

Rhodes Brown, *McGill University*

John Jorgensen, *McGill University*

Qin Wang, *McGill University*

Karel Driesen, *McGill University*

Laurie Hendren, *McGill University*

Clark Verbrugge, *McGill University*

Many tools are available to profile and visualize Java™ programs. However, those we have encountered are limited by the fixed set of data they collect. Frustrated by this limitation, we developed STOOP: the Sable Toolkit for Object-Oriented Profiling, a framework for creating tools to collect and visualize arbitrary profile data.

33 Handling Crosscutting Constraints in Domain-Specific Modeling

Jeff Gray, *Vanderbilt/ISIS*
 Ted Bapty, *Vanderbilt/ISIS*
 Sandeep Neema, *Vanderbilt/ISIS*
 James Tuck, *Vanderbilt/ISIS*

Domain-specific models for embedded systems often contain constraints that aid in stipulating design criteria. These constraints, however, are typically scattered across a model hierarchy in such a manner that it is difficult to reason about the effect and purpose of each constraint. This poster describes an approach for providing better separation of concerns with respect to constraints.

34 Results of the Workshop: Human Issues on Agile Processes

Jens Coldewey, *Coldewey Consulting*
 Jutta Eckstein, *Objects in Action*
 Pete McBreen, *McBreen Consulting*
 Alastair Handley, *Pragmatic Software Consulting Services Ltd.*

Many of the values stated by the “Manifesto for Agile Software Development” are Human Issues in Agile Processes. These issues affect team- as well as client management and team collaboration. The poster presents the results of the corresponding workshop, whose goal is to identify and illuminate these human issues.

35 Enterprise-Scale Java™ Object-To-Relational Persistence Strategies Compared

Changrong Ji, *The St. Paul Companies*
 Casey Phipps, *The St. Paul Companies*
 Alan Weiss, *The St. Paul Companies*
 Wen Ji, *Independent Professional Services*

Enterprise-scale Java™ object-to-relational persistence strategies are briefly outlined and compared. Domain-integrated persistence frameworks are compared to distributed persistence services, and inheritance vs. metadata approaches are contrasted. We also include a discussion of how EJB, JDO and O/R mapping tools such as TOPLink fit into each strategy.

36 Beyond Design: Patterns (mis)used

Christa Schwanninger, *Siemens AG*
 Elisa Baniassad, *University of British Columbia*
 Vera Seidel, *Siemens AG*
 Gail Murphy, *University of British Columbia*

This poster visualizes the results of the OOPSLA workshop called “Beyond Design: Patterns (mis)used”. This workshop is intended to collect attempts to use and perform research on aspects of current patterns beyond the act of transferring design experience. This could include research areas such as reverse engineering programming languages.

37 Model Driven Analysis & Design of XML Schemas

David Carlson, *Ontogenics Corp.*

The concept of model driven architecture (MDA) has become a focal point for the Object Management Group. This poster describes two approaches, using two different UML profiles, for refining abstract, platform-independent models (PIMs) into platform-specific models (PSMs) that can be used to generate XML Schemas.

POSTERS

38 Design Aspects for Describing Frameworks

Federico Balaguer, *University of Illinois at Urbana-Champaign*

This poster presents an extension to UML for describing design aspects of frameworks. Aspects are documented by applying a UML Profile called “Framework Description” to class diagrams. Design Aspects of frameworks are useful for reasoning about extension and instantiation scenarios, as well as for designing the applications that rely on them.

39 Visualizing Indirect Branch Hot Spots in Object-Oriented Programs

Matthew Holly, *McGill University*

Karel Driesen, *McGill University*

We demonstrate four visualizations of indirect branch instructions corresponding to switch statements and virtual function calls in object-oriented programs. Spatial and Temporal hot spot visualizations highlight code locality. Footprints show dynamic program size while prediction profiles visualize the regularity of a program phase.

40 An Object-Oriented Framework for Distributed Numerical Computations

Roxana Diaconescu, *Norwegian University of Science and Technology*

Writing concurrent distributed memory applications requires skills beyond the background of a researcher experimenting with scientific problems. We propose a component framework to capture the concurrency infrastructure for dynamically distributed numerical applications. We focus on the problem of concurrent Finite Element Method (FEM) solution of Partial Differential Equations (PDEs) for general (unstructured) meshes.

41 Autonomous Points in Component Composition

Vladimir Mencl, *Charles University, Prague*

Current technologies provide only primitive support for managing configuration of applications composed of software components. Advanced facilities are needed, namely, for specifying the points in a component hierarchy suitable for initiating an update or acquisition from an independent vendor. Autonomous points proposed in this poster reflect these requirements.

42 ArchJava: Connecting Software Architecture to Implementation

Jonathan Aldrich, *University of Washington*

Craig Chambers, *University of Washington*

David Notkin, *University of Washington*

ArchJava is a small extension of Java™ that integrates software architecture smoothly into Java™ code. Our approach enables reliable reasoning about system structure, because ArchJava ensures that the actual code corresponds to the conceptual architecture. This poster describes the ArchJava language, the downloadable tools, and our initial practical experience.

43 Comanche, Rapid Web Development for the Rest of Us

Bolot Kerimbaev, *Georgia Institute of Technology*

Our goal is to enable rapid development of web applications by non-professional programmers. Rapid development is achieved through a simplified model of building web applications, availability of familiar development tools, and automation of some tasks. Performance tuning ensures that dynamic applications retain good performance.

44 Design Support for Aspect-Oriented Software Development

Christina von Flach G. Chavez, *Pontifical Catholic University of Rio de Janeiro (PUC-Rio)*

Carlos J. P. de Lucena, *PUC-Rio*

We propose a design model for aspect-oriented software development, which incorporates the main features of aspect-oriented programming and is language- and process-independent. A set of design principles for aspect-oriented modeling is under investigation. A representative case study and a development scenario illustrate the suitability of the proposed ideas.

45 Concepts Reuse for Requirements Specifications

Walaal-Eldeen Mohamed Bakry, *Middlesex University*

Successful specification reuse in software development hinges on finding a mechanism that links new requirements to old ones. This research proposes Concepts Reuse as a new approach to specification reuse. It provides a classification scheme based on generic patterns that aid the matching process between old and new requirements.

46 OOSPICE – Bringing Objects and Components to Process Improvement and Capability Determination

Mark Woodman, *Middlesex University*

Freiderich Stallinger, *Kepler University Linz*

Alec Dorling, *University of Borås*

OOSPICE is an EU/Australian project to extend the SPICE approach to software process improvement and capability determination to cover object-oriented and component-based development. Major deliverables include: a unified CBD process metamodel, a tool-supported CBD assessment methodology, a CBD method and tool, and extensions to the ISO/IEC 15504 process assessment standard.

47 Separation of Concerns in Multi-Agent Software Engineering

Viviane Silva, *PUC-Rio*

Otávio Silva, *PUC-Rio*

Alessandro Garcia, *PUC-Rio*

Christina Chavez, *PUC-Rio*

Carlos Lucena, *PUC-Rio*

Agent technology has been revisited as a complementary approach to the object paradigm. Although objects and agents have many similarities, the introduction of agents in the object model poses new problems. In this context, we present our approach for dealing with the intricacies of developing agent systems using recent advances of separation of concerns techniques.

48 Raising the Level of Abstraction of Design Models

Joern Bettin, *Equinox Software Architects*

We have used a model-driven software development approach and template-based code generation techniques to build a complex Java™ application within the electricity industry. Our approach allows us to significantly raise the level of abstraction of design models, results in highly compact representations of application architecture and design, and highlights weaknesses of the UML and current UML tools.

POSTERS

49 A Reusable Design for Building Dynamically Programmable and Workflow-enabled Object-Oriented Software

Reza Razavi, *University of Paris 6 (LIP6)*

“Expert-programmable” software is software that is designed to allow non-programmer domain experts to define desired classes by defining at runtime new types of objects, their structure and behavior. There is currently no standardized design for creating such software. The author proposes a solution supporting dynamic composition of dynamically defined, workflow-enabled services.

50 EasyMock: Dynamic Generation of Mock Objects

Tammo Freese, *OFFIS*

This poster presents a Java™ library called EasyMock. EasyMock provides dynamically generated mock objects for interfaces by using Java™ proxies. This simplifies unit testing by allowing the generation and usage of mock objects directly within the test code.

51 Third Workshop on Best-Practices for the Design and Implementation of Business Rules

Ali Arsanjani, *IBM*

Joe Yoder, *Joe Yoder Enterprises*

Alan Abrams, *Cambridge University*

Reza Razavi, *Université Pierre et Marie Curie*

Jeff Oakes, *The Hartford*

Keith Levi, *Maharishi University of Management*

Businesses implement their information systems upon the pillars of business rules. Yet, writing scalable, robust, and adaptable business rules for business objects in today’s e-business and n-tier architectures is a challenge! This workshop will categorize and capture, in pattern format, best and successful practices in the design and implementation of business rules within the context of business objects and components.

52 Retest: A Regression Testing Tool for Java™ Software

Alessandro Orso, *Georgia Institute of Technology*

Donglin Liang, *Georgia Institute of Technology*

This poster presents a technique that effectively selects, from the test suite used to test the original version of a Java™ program or subsystem, all test cases that may reveal a fault in the modified version of the software. It also presents a system that implements the technique and empirical studies that demonstrate its effectiveness.

53 The Jalapeño Research Virtual Machine for Java™

Bowen Alpern, *IBM Research*
 Matthew Arnold, *IBM Research*
 C. R. Attanasio, *IBM Research*
 Maria Butrico, *IBM Research*
 Jong-Deok Choi, *IBM Research*
 Anthony Cocchi, *IBM Research*
 Julian Dolby, *IBM Research*
 Stephen Fink, *IBM Research*
 David Grove, *IBM Research*
 Michael Hind, *IBM Research*
 Mark Mergen, *IBM Research*
 Ton Ngao, *IBM Research*
 Igor Pechtchanski, *IBM Research*
 Vivek Sarkar, *IBM Research*
 Stephen E. Smith, *IBM Research*
 Peter Sweeney, *IBM Research*
 Martin Trapp, *IBM Research*

Building an infrastructure to explore Java™ design and implementation issues exceeds the resources of most academic research projects. This poster provides an overview of Jalapeño, a state-of-the-art research virtual machine for Java™ infrastructure, written in Java™ at IBM Research. Source for Jalapeño (available under academic license) has been acquired by at least a dozen universities.

54 The Aspect Mining Tool - Support for Concern Mining

Jan Hannemann, *University of British Columbia*
 Gregor Kiczales, *University of British Columbia*

Refactoring legacy systems to improve modularity is difficult since the current modularization may have left some concerns unextracted. To identify the complete concern code despite tangling and scattering, developers need the right information about the system. We propose combining lexical and structural queries to identify concern extents.

55 Demeter Aspects

Karl Lieberherr, *Northeastern University*
 David Lorenz, *Northeastern University*
 Doug Orleans, *Northeastern University*
 Johan Ovlinger, *Northeastern University*
 Mitchell Wand, *Northeastern University*
 Pengcheng Wu, *Northeastern University*

The Demeter group summarizes its activities in aspect-oriented software development (AOSD). Aspectual Collaborations are an extension of adaptive plug-and-play components with method replacement to modularize and parameterize aspects. Predicate Dispatch Extension adds around methods and other extensions to predicate dispatching to provide a flexible AOSD system.

POSTERS

56 Tool Support for Managing Crosscutting Concerns in Existing Artifacts

Elisa Baniassad, *University of British Columbia (UBC)*

Martin Robillard, *UBC*

Albert Lai, *UBC*

Gail Murphy, *UBC*

Software developers spend a lot of time handling crosscutting concerns. Tool support can help a developer find, understand, and manage these aspects. This poster describes three tools we have built to help address these needs for both source-level and design-level artifacts.

57 Separation of Distribution Concerns in Distributed Java™ Programming

Michiaki Tatsubori, *University of Tsukuba*

We propose an aspect-oriented distributed programming tool, with which programmers can specify the distribution aspect of program simply and separately from a non-distributed Java™ program, to enhance the modularity of a program. The aspect-weaver of this tool is a bytecode translator implemented as a customized class loader in Java™.

58 Assessing the Quality of Object-Oriented Designs

Ralf Reifing, *University of Stuttgart*

Design quality is vital for reducing software cost. Unfortunately, it is unclear what design quality really is. This work wants to clarify the notion of design quality and make it measurable. A quality model is created that can be used for design assessment, comparing design alternatives, and design improvement.

59 A Multithreaded Concurrent Generational Garbage Collector for Java

Chia-Tien Dan Lo, *Illinois Institute of Technology*

J. Morris Chang, *Illinois Institute of Technology*

A new multithreaded concurrent generational garbage collector based on mark-sweep with the assistance of reference counting is proposed and implemented. The scheme takes advantage of multiple processors in SMP systems and the merits of threads. Furthermore, it reduces garbage collection pauses and enhances garbage collection efficiency. Measurement results are studied.

60 On the Syllogistic Structure of Object-Oriented Programming

Derek Rayside, *University of Waterloo*

Kostas Kontogiannis, *University of Waterloo*

This poster demonstrates a prototype implementation of a novel program browser for understanding polymorphic calls in Java programs. The browser's design was first presented at ICSE 2001 and is inspired by Aristotle's syllogism. The prototype is built on IBM's new WebSphere Studio Workbench (aka Eclipse) technology.

61 Integration of Independently-Developed Object-Oriented Designs

Adam Batenin, *University of Bath*

Programming technology should facilitate the independent development of object-oriented designs and their subsequent composition to create complete programs. We show that even in an environment where a limited independence is granted, relatively simple programs are difficult to integrate using present technologies. We propose techniques to address these issues.

62 An OCL Query-Based Debugger for C++

Chanika Hobatr, *Clemson University*

Development using object technology can make debugging a daunting task due to the number of generated objects and the complexity of their interrelationships. We propose a debugging tool that permits the expression of complex relationships to be formulated easily and evaluated efficiently. OQBD is a query-based debugger for C++ programs using queries formulated in the Object Constraint Language.

63 Hyper/J™: Supporting Decomposition, Integration, and Evolution of Java™ Software

Harold Ossher, *IBM Thomas J. Watson Research Center*

Peri Tarr, *IBM Thomas J. Watson Research Center*

Vincent Kruskal, *IBM Thomas J. Watson Research Center*

Hyper/J™ supports a new approach to constructing, integrating and evolving standard Java™ software called Multidimensional Separation of Concerns. Developers can decompose and organize software according to multiple, arbitrary criteria (concerns) simultaneously—even after the software has been implemented—and integrate the pieces into larger-scale components and systems. Hyper/J works on standard Java class files and is freely available.

EDUCATORS' SYMPOSIUM

Celebrating the 10th anniversary of the Educators' Symposium—the unique forum for trainers and educators.

Chair: Jutta Eckstein, *Objects in Action*

The Educators' Symposium is for industry and academic professionals who have a vested interest in object technology training and education. This one-day symposium is a unique forum for trainers and educators to discuss their needs and ideas for incorporating object technology into training plans and courses under consideration of the human factor. We will celebrate this year's symposium by reflecting on what we achieved in the last decade and also focusing on the challenges we face in the near future.

The Educators' Symposium facilitates the swapping of ideas in a number of ways, including featured talks by professionals on the cutting edge of OO education, paper presentations, activity sessions, posters, demonstrations and other opportunities to exchange course materials. All attendees are invited to actively participate in the symposium by bringing any course related material they would like to share (exercises, teaching and learning tools, effective evaluation questions and methods, URLs, descriptions of course needs, things you want to brag about or advertise, etc.).

8:30 am - 9:40 am

Marriott Hotel — Salon C-D Program

Welcome and DesignFest Introduction

Jutta Eckstein,
Objects in Action

Ralph Johnson,
University of Illinois at Urbana-Champaign

Keynote

Object Technology Education Today

Linda Northrop,
Software Engineering Institute

Over the last ten years, object technologists have spawned major developments that have changed our implementations and our approach. Use cases, standardized middleware, Java, UML, aspect-oriented programming, and design patterns have influenced the way we build systems and have now become part of the technical fiber we routinely use. What about object technology education? Has it kept pace? Is the focus properly fixed? Together we will examine the ten year journey and the challenges that remain.

9:40 am - 10:00 am

Panel**Educators' Symposium: The First Ten Years**

Mary Lynn Manns,
University of North Carolina at Asheville

Ed Gehringer,
North Carolina State University

Joe Bergin,
Pace University

Rick Mercer,
University of Arizona

This year marks the 10th consecutive OOPSLA Educators' Symposium. In the beginning, Educators' Symposia were organized rather like technical conferences, with invited talks and contributed papers. More recently, they have evolved to a participatory format, with poster and demonstration sessions, and design exercises. The panelists, who have attended all ten symposia, will reminisce on these symposia by considering the topics, the opinions, the debates, the people, and the hype. We will connect with happenings in other parts of the OOPSLA conference and the trends of OO education in industry and academia.

10:00 am - 10:30 am

Break and Sneak-through the DesignFest

10:30 am - 11:30 am

Paper Presentations: Rethinking — How do we teach and what?**Extreme Programming in an Introductory Course Taught using the Java Programming Language**

Daniel Steinberg,
Dim Sum Thinking Inc.

Daniel Palmer,
John Carroll University

It seems natural to introduce Extreme Programming (XP) to an upper level course in software engineering. This methodology is well suited to working on projects for real clients that have restrictive time constraints for completing a project that satisfies the client's most pressing needs. In this paper, we'll discuss the benefits of introducing at least some of the practices of XP much earlier in the curriculum.

On the Conflict between Teaching Software Engineering and Teaching Computer Science

Michael Whitelaw,
Charles Sturt University, Australia

There is no clear distinction in the computing field between the conservatism of a (Software) Engineer and the risk-taking curiosity of a (Computer) Scientist. Reflecting this confusion, most of our courses transmit ambiguous messages to students as to what is permissible and what is not permissible. Some of the differences that the contrasting attitudes take can be seen in different approaches to life-cycle methodologies, code-reuse, tools, and even the very understanding of concepts like object. At the very least, we need to clarify these differences for our students; in the long term we must expect our courses to diverge.

Containers and Iterators: An Example of Constructive Elicitation of Patterns

Arturo Sánchez,
University of North Florida

Design Patterns are currently being used as an integral part of curricula that introduces object-oriented technology from the first course and up. A popular approach to using patterns in introductory computer science courses consists of presenting a catalog to be applied for tackling problems whose complexity level is appropriate for the course in question. Although this is the natural approach for it, goes in tune with the goal of promoting a lingua franca among software designers and programmers, it is not the only viable approach. In this paper we present an example that illustrates a constructive approach to the problem of unveiling three cognitive processes that take place in the discovery of patterns, namely observation of repeatability, lifting (or generalization) of specific solutions to more general ones, and specialization of general solutions to specific ones. The example starts with a very simple and specific iteration mechanism (Java's Enumeration) and covers a series of problems whose solutions converge to the Iterator Pattern. Our example can be used to introduce novices to the concept of patterns, and also to introduce the already initiated pattern mining.

Grasp All, Loose All

Marianna Sipos,
Budapest University of Technology and Economics, Hungary

Learning a new programming paradigm without any practical experience seems to be impossible. So it is not enough to know the OO concepts, but also a language and a framework have to be taught. This paper describes my solution, which gives the students clear concepts, success with small exercises, and the feeling that they are capable of solving bigger problems in project work.

11:30 am - 12:15 pm

Demonstration**DesignFest on Stage**

Ralph Johnson,
University of Illinois at Urbana-Champaign

The OOPSLA DesignFest is about design and creativity. The DesignFest is a free event (for conference registrants) that was created to give OOPSLA attendees the opportunity to learn more about design by doing it. In the DesignFest people work in small groups to solve a particular design problem, bringing to bear their experience and skills in object-oriented design and/or experience working on similar problems. The goal is to learn new techniques from each other and to uncover and articulate the analysis and design patterns that are already used subconsciously. DesignFest is also a great way to get some first-time experience, which means it is an excellent technique for teaching Design!

12:15 pm - 1:15 pm

Marriott Hotel — Meeting Rooms 8-9
Lunch

1:15 pm - 2:00 pm

Invited Talk**Object-Oriented Thinking Is Easy To Learn, But Seldom Taught. Experiences 1965-2001**

Kristen Nygaard,
University of Oslo and the Norwegian Computing Center, Norway

Kristen Nygaard invented object-oriented programming together with Ole-Johan Dahl in the 1960s. The languages Simula I and Simula 67 introduced objects, classes, inheritance, virtual quantities and quasi-parallel (multi-threaded) sequencing. He is of the opinion that the standard pedagogic in teaching object-oriented programming is wrong. His approach, which he has lectured around the world, is that one must start with “sufficiently complex examples” instead of “sufficiently simple examples” in order to teach the pupils the “world view” of object-orientation. Otherwise the pupils will continue programming as before, albeit in an object-oriented language.

2:00 pm - 3:00 pm

Paper Presentation: Teaching Collaboration Skills**Redesigning CS101: A Learning Based Approach**

Alfonso Rodriguez and Ariel Ortiz,
ITESM Campus, Mexico

Nowadays, software developers are required to build better systems in less time. We expect computer professionals to work in teams, learn new technologies, solve tough problems, and exceed all past achievements. But as educators, are we really promoting these skills and values? This paper presents the experiences obtained from redesigning an objects-first CS101 course, in which students were made responsible of their own learning practice. Interesting projects developed in teams and built from self-acquired knowledge are the foundation of our proposed scheme.

Techniques for Active Learning of OO Development

Robert Biddle, James Noble and Ewan Tempero,
Victoria University of Wellington, New Zealand

We describe our use of active learning techniques to teach OO development. We have developed new techniques, adapted from CRC cards, to teach use cases for requirements gathering. We have also adapted CRC cards to teach the principles of OO. Our approach has been tried with large university classes as well as industry groups, programmers as well as business analysts and managers.

Cooperatively Enriching Education: Industrial Projects for Academic Credit

Dr. Edward F. Gehringer,
North Carolina State University

Dave Maeda,
IBM VisualAge for Smalltalk Group

Industry and universities have long worked together on cooperative education (co-op) projects in which a student spends a semester or more working in industry. Could even shorter-term projects be of mutual benefit? During the Spring 2001 semester, NCSU and the IBM Smalltalk Group set up a program where an NCSU faculty member and IBM employees jointly supervised students working on small, well-defined Smalltalk projects for academic credit. Benefits to the students included gaining experience on a real-world software project directly related to their coursework, and making valuable contacts in industry. Benefits to the NCSU Computer Science department included offering its students the ability to work with practicing software developers in a small-group setting. For its part, IBM obtained short-term help on three projects, and acquaintance with several potential job candidates.

3:00 pm - 3:45 pm

Workshop Reports, Poster Session and Break

3:45 pm - 5:15 pm

Activity**Looking for Abstractions in a Concrete World: Candidates, Responsibilities, and Collaborations**

Rebecca Wirfs-Brock and Alan McKean,
Wirfs-Brock Associates

CRC has traditionally stood for “Classes-Responsibilities-Collaborators.” But Classes are too concrete. Jumping to classes too soon results in a lifeless design. We should be thinking more abstractly: in terms of distinct roles that collaborate. Classes imply code; roles imply behavior. Thinking about roles can simplify a design, but it is hard to make the leap from concrete objects to abstract roles. This talk demonstrates how to think about candidate roles and their responsibilities, what the benefits are, and how to implement these abstractions with interfaces and classes.

5:15 pm - 5:30 pm

Wrap-up and Open Microphone

DOCTORAL SYMPOSIUM

Chair: Doug Lea, *SUNY Oswego*

Monday

Marriott Hotel — Meeting Room 10

OOPSLA 2001 will provide a selected group of doctoral students with a forum in which to present their work and to obtain guidance from mentors drawn from university faculty and industry laboratories. The goal of the symposium is to expose the student to external helpful but critical peers before their defense in order to improve upon their thesis work, as well as to give points of advice for job interviews. This year's mentors are Brent Hailpern (IBM), Doug Lea (SUNY Oswego), James Noble (Victoria University of Wellington, NZ), Mary Beth Rosson (Virginia Tech), Richard Gabriel (Sun Microsystems), and Ron Goldman (Stanford University).

As in the past, this year's selected participants were chosen on the basis of the mentors' evaluation of the students' descriptions of their research, and whether the work was sufficiently advanced to have some preliminary results but with sufficient time remaining to benefit from the symposium experience.

The participating doctoral students and the titles of their research work are:

1 Integration of Independently-Developed Object-Oriented Designs

Adam Batenin, *University of Bath*

Programming technology should facilitate the independent development of object-oriented designs and their subsequent composition to create complete programs. We show that even in an environment where a limited independence is granted, relatively simple programs are difficult to integrate using present technologies. We propose techniques to address these issues.

2 Design Support for Aspect-Oriented Software Development

Christina von Flach G. Chavez and Carlos J.P. de Lucena,
Pontifical Catholic University of Rio de Janeiro

We propose a design model for aspect-oriented software development, which incorporates the main features of aspect-oriented programming and is language and process independent. A set of design principles to aspect-oriented modeling is under investigation. A representative case study and a development scenario will illustrate the suitability of the proposed ideas.

3 Transmigration of Object Identity: The Programming Language GILGUL

Pascal Costanza, *University of Bonn*

Gilgul is an extension of Java that strictly separates the notions of reference and comparison that are traditionally subsumed in the concept of object identity. This allows for the introduction of new operations that open up new degrees of flexibility by providing means for unanticipated, dynamic software adaptation.

4 A Multithreaded Concurrent Generational Garbage Collector for Java

Chia-Tien Dan Lo and J. Morris Chang,
Illinois Institute of Technology

A new multithreaded concurrent generational garbage collector based on mark-sweep with the assistance of reference counting is proposed and implemented. The scheme takes advantage of multiple processors in SMP systems and the merits of threads. Furthermore, it reduces garbage collection pauses and enhances garbage collection efficiency. Measurement results are studied.

5 An Object-Oriented Framework for Distributed Numerical ComputationsRoxana Diaconescu, *Norwegian University of Science and Technology*

Writing concurrent distributed memory applications requires skills beyond the background of a researcher experimenting with scientific problems. We propose a component framework to capture the concurrency infrastructure for dynamically distributed numerical applications. We focus on the problem of concurrent Finite Element Method (FEM) solution of Partial Differential Equations (PDEs) for general (unstructured) meshes.

6 A Framework for Creating Aspect WeaversJeff Gray, *Vanderbilt University*

We present two objectives for extending Aspect-Oriented Programming (AOP) research. First, the concept of AOP is investigated at a higher level of abstraction, focusing on the application of aspect-oriented techniques to model-integrated computing. Second, we create a framework that aids in the construction of new aspect weavers. The framework utilizes domain-specific languages and generators to provide variability among weaver instances.

7 An OCL Query-Based Debugger for C++Chanika Hobatr, *Clemson University*

Development using object technology can make debugging a daunting task due to the number of generated objects and the complexity of their interrelationships. We propose a debugging tool that permits the expression of complex relationships to be formulated easily and evaluated efficiently. OQBD is a query-based debugger for C++ programs using queries formulated in the Object Constraint Language.

8 Supporting the Use of Object-Oriented FrameworksDaqing Hou, *University of Alberta*

Object-oriented frameworks are often hard to learn and use. The capability to automatically detect errors occurring at the boundary between frameworks and applications is crucial to mitigate the problem. We introduce the notion of framework constraints and a specification language, FCL (Framework Constraints Language), to formally specify them. Framework constraints are rules that frameworks impose on the code of framework-based applications.

9 Assessing the Quality of Object-Oriented DesignsRalf Reiing, *University of Stuttgart*

Design quality is vital for reducing software cost. Unfortunately, it is unclear what design quality really is. This work wants to clarify the notion of design quality and make it measurable. A quality model is created that can be used for design assessment, comparing design alternatives, and design improvement.

10 Separation of Distribution Concerns in Distributed Java ProgrammingMichiaki Tatsubori, *University of Tsukuba*

We propose an aspect-oriented distributed programming tool, with which programmers can specify the distribution aspect of program simply and separately from non-distributed Java program, for enhancing the modularity of program. The aspect-weaver of this tool is a bytecode translator implemented as a customized class loader in Java.

DOCTORAL SYMPOSIUM, STUDENT VOLUNTEERS, BIRDS OF A FEATHER SESSION

11 Interdependence of Software Evolution and Development Process Evolution in Agile Methodologies

Christian Wege, *University of Tuebingen*

A software development process can become more successful by analyzing the produced artifacts and feeding the results back into the process. This research will produce techniques for analyzing the artifacts, show how to interpret them and list a collection of experiences made in real world projects.

STUDENT VOLUNTEERS

Chair: Dirk Siebert, *Leipzig University*, sv@oopsla.acm.org

The student volunteers program is an opportunity for students from around the world to associate with the top people in object-oriented technologies and software development. In return for about ten hours of their time, student volunteers receive complimentary registration and other benefits. In the past, job assignments have included assisting with tutorials, and panels, checking badges at doors, helping with traffic flow, and general go-for assistance to keep the conference running smoothly.

Student volunteers need to be enrolled in a full-time undergraduate, or graduate program at the time of the conference.

More detailed information is available on <http://oopsla.acm.org/~sv>.

BIRDS OF A FEATHER SESSIONS (BOFs)

Birds of a Feather Sessions are informal gatherings that provide a forum for discussion of a particular topic, tool, product, etc. Several rooms will be available for BOFs each evening from Sunday through Wednesday. Rooms may be reserved by any Conference attendee on a first-come, first-served basis by signing up at the Information Booth at the Conference. There will be no advance reservations for these rooms prior to the Conference.

BOF organizers are responsible for providing descriptive information for their session, which OOPSLA will post in the OOPSLA Courtyard. BOFs require no registration or submission for attendance. All OOPSLA attendees may attend any BOF session(s).

DESIGNFEST

DESIGNFEST

(including AnalysisFest and CodeFest)

Chair: Torsten Layda,
SWX Swiss Exchange, designfest@oopsla.acm.org

The OOPSLA DesignFest is about design and creativity. The DesignFest is a free event (for conference registrants) that was created to give OOPSLA attendees the opportunity to learn more about design by doing it. DesignFest is an increasingly popular event at OOPSLA and many participants choose to return year after year. If you don't have much knowledge about design, this is probably a great way to get some first-time experience.

DesignFest is not about passively sitting and listening to experts talk about design. DesignFest is about sharpening your design skills by rolling up your sleeves and working on a real problem with others in the field. You can expect to interact with everyone from total beginners to textbook authors! DesignFest participants work in small groups to solve a particular design problem, bringing to bear your experience and skills in object-oriented design and/or experience working on similar problems. The goal is to learn new techniques from each other and to uncover and articulate the analysis and design patterns that we already use subconsciously.

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.

In recent years, the CodeFest teams have become a regular part of DesignFest. These teams of student programmers will implement some of the designs created during the DesignFest. In this way the DesignFest participants will be able to see how good their designs really are. During the conference, you'll be able to see the CodeFest students hard at work, and you are welcome to ask them questions. On Sunday and Monday, they will be located in a prominent place, close to the Convention Center Waterfront, and on the remaining days you can find them in the OOPSLA Courtyard.

Starting in 1999, DesignFest added a sub-program called AnalysisFest. In these sessions, open to DesignFest participants, professionals form groups to study carefully statements of need and requirements documents. Some of these are the same documents used in DesignFest, but others are not. Each group must come up with precise specifications of the domain of the problem they are given, without any regard to solution (design). AnalysisFest is not a tutorial on object-oriented analysis; it is an opportunity for analysts to sharpen and measure their skills by interacting with their peers. Appropriate analysis of a system's problem domain increases the understanding of requirements and leads to better, more robust design solutions. Participants in the DesignFest can be in an Analysis group, a Design group, or a combined group.

Design teams are provided with a timetable, process hints, and a list of deliverables. During the session, teams will get the opportunity to improve their designs by periodically comparing their work with that of other teams solving the same problem independently.

Each team will be asked to produce a poster or two that summarizes their work. These posters will be presented at the social event on the final day of the conference where participants can review the work of other teams, and view the results from the CodeFest teams. Each team has a moderator and a recorder. The recorder is responsible for reporting about what the group has achieved in design and what they have learned. The moderator is responsible for making sure that the team makes progress, without leading them in any particular direction.

For each problem we provide a detailed description of the domain and a requirements specification. For certain problems, a domain expert who acts as the customer is available for a short period during the design sessions.

The following problems have already been selected for this year's DesignFest:

- Materials Transport — Dump Truck Scheduling in Surface Mining
- A Reverse Buying System
- FlexTier — A Flexible Extensible Service Tier
- Managing Enrollment Policies for Insurance Plans
- VelcoWare — A Warehouse Manager
- DVD+RW Recorder
- Securities Trade Matching
- The Viking — A Direct Marketing System (a repeat of a popular description from previous years)

You are invited to take a tour of all the DesignFests and CodeFests at <http://designfest.acm.org>.

The DesignFest Committee appears in the Conference Committees section of the Program.

DESIGNFEST

DESIGNFEST AT A GLANCE

Note that all DesignFest and AnalysisFest sessions are separate sessions, lasting half a day or a full day, while the CodeFest is a continuous event.

Sunday

8:30 am - 5:00 pm	Full-day DesignFest and AnalysisFest	Marriott Hotel — Meeting Room 8-9
12:00 pm - 5:00 pm	CodeFest	Convention Ctr — Waterfront

Monday

8:30 am - 5:00 pm	CodeFest	Convention Ctr — Waterfront
8:30 am - 12:30 pm	Half-day DesignFest at Educators' Symposium	Marriott Hotel — in front of Salon C-D

Tuesday

10:00 am - 5:00 pm	CodeFest	Convention Ctr — OOPSLA Courtyard
10:30 am - 5:00 pm	Full-day DesignFest / optional AnalysisFest	Marriott Hotel — Meeting Rooms 3, 4, 5, 6, 7, & 13
1:30 pm - 5:00 pm	Half-day DesignFest / optional AnalysisFest	

Wednesday

10:00 am - 5:00 pm	CodeFest	Convention Ctr — OOPSLA Courtyard
1:30 pm - 5:00 pm	Half-day DesignFest / optional AnalysisFest	Marriott Hotel — Meeting Rooms 3, 4, 5, 6, 7, & 13

Thursday

10:00 am - 3:00 pm	CodeFest	Convention Ctr — OOPSLA Courtyard
3:30 pm - 5:00 pm	DesignFest/AnalysisFest/CodeFest Wrap-Up (joint venture with the ice-cream social) Presentation of final analyses and designs, and demos of final code.	Convention Ctr — Ballroom D

CAMP SMALLTALK

Camp Smalltalk combines the power of Smalltalk with the energy of open source. We will set up Camp in the exhibition hall. Stop by if you want to join in, to see Smalltalk in action, or just to meet an old friend. For further information, see <http://camp.smalltalk.org>.

EXHIBITS

EXHIBITS

For an updated Exhibits Floor Plan, the latest list of OOPSLA Exhibitors with booth numbers and a description of their products and services, consult your OOPSLA 2001 Exhibits Directory!

As the heart of every OOPSLA Conference, the Exhibits provide an informal venue for interaction between Conference attendees and the software developers, publishers, technology researchers, framework architects, personal electronic device suppliers, training consultants and recruiters who offer the latest technology and information to the object-oriented community.

Like a bustling village square, the Exhibits are a place for 2500 object-oriented citizens to consult with exhibiting companies, meet personally with OOPSLA speakers one-on-one, consider the latest Poster presentations, browse at special activity locations like Camp Smalltalk, and still find time for lunch or refreshments.

The OOPSLA 2001 Exhibits will be held in the West Hall of the Tampa Convention Center, right behind the Conference registration area.

- The “OOPSLA Courtyard” on the Exhibits floor will provide food and beverage concessions; message boards; Camp Smalltalk; and an informal area for one-on-one interaction with featured conference speakers and authors.
- Conference refreshment breaks will take place in the Exhibits.
- The popular Publishers’ Pavilion, where attendees can browse for the latest periodicals and books, will be in the Exhibits.
- **New this year! Geek Alley!** Where the latest technology in personal computing devices and peripherals can be purchased directly from the exhibiting company.
- OOPSLA Posters will again be integrated with the exhibit booths.
- The Exhibits Reception – open to all Conference and Exhibits participants – will be held Tuesday from 5:00-7:00 pm on the Exhibits floor!

Exhibits Hours: Tampa Convention Center — West Hall

Tuesday, October 16

10:00 am – 1:30 pm (Note: Exhibits are closed from 1:30-3:00 pm)

3:00 pm – 7:00 pm

Exhibits Reception – 5:00-7:00 pm

Wednesday, October 17

10:00 am – 5:30 pm

Thursday, October 18

10:00 am – 2:00 pm

SPECIAL EVENTS

OOPSLA 2001 provides the opportunity to mingle with the recognized leaders in object technology in a more casual setting through a number of social activities designed just for that purpose. In addition, the OOPSLA Courtyard, in the center of the Exhibit Hall, will offer food and beverage concessions, message boards, the OOPSLA CodeFest, and one-on-one interaction with Conference speakers and authors.

Name badges will be required to attend events.

Tutorial and Workshop Reception

*Sunday Evening,
October 14, 5:30 pm – 7:30 pm
Marriot Hotel — Grand Ballroom*

The reception is open to anyone who attends (or leads) at least one Tutorial or Workshop. This is an excellent forum to share information on the day's activities.

OOPSLA 2001 Welcome Reception

*Monday Evening,
October 15, 5:30 pm – 7:30 pm
Convention Center — Ballrooms B-C*

All full (and Monday one-day) conference registrants are invited to attend the Welcome Reception. OOPSLA are encouraged to view the OOPSLA 2001 Poster Session during the Welcome Reception.

Newcomers' Orientation

*Monday Evening after Welcome Reception,
October 15,
7:30 pm – 8:00 pm
Convention Center — Ballroom D*

A special session has been arranged for OOPSLA newcomers. If you have never been to an OOPSLA before, this is a unique opportunity to meet other first-time OOPSLA participants early in the conference and to learn about the facets of OOPSLA and how to maximize your OOPSLA experience.

Exhibits Reception

*Tuesday Evening,
October 16, 5:00 pm – 7:00 pm
Exhibit Hall*

Located in the Exhibit Hall, this reception is open to all full and Tuesday one-day conference registrants. Stop by for chance to view the exhibits and visit Geek Alley.

Special Event "Under the OCEAN"

*Wednesday Evening,
October 17, 7:00 pm – 10:00 pm
The Florida Aquarium*

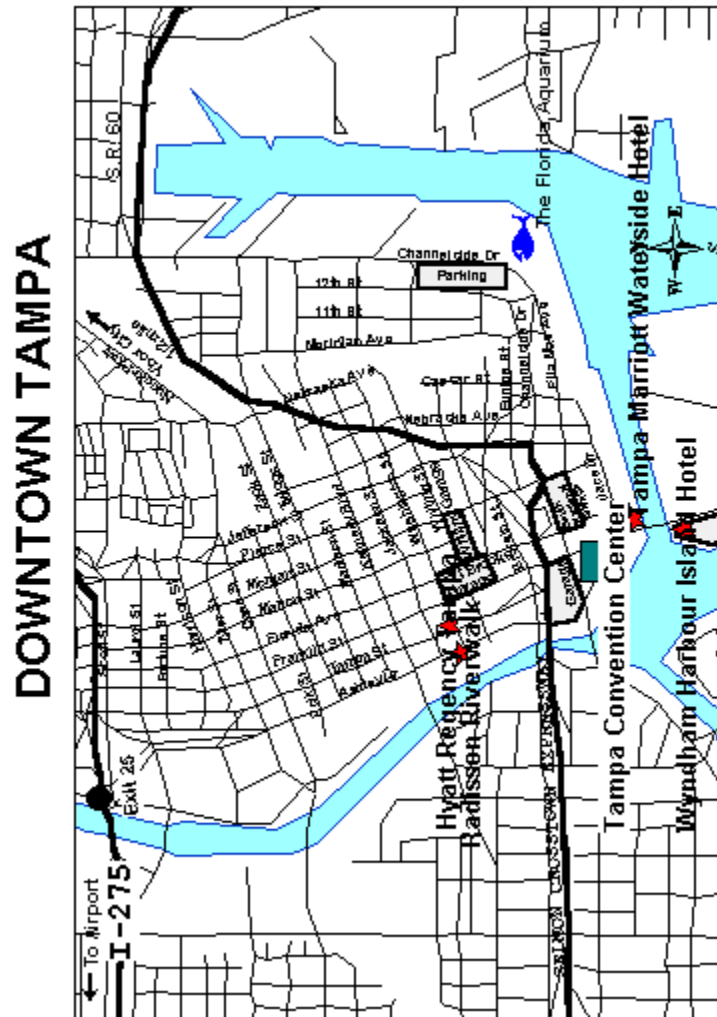
Are alligators sneaky?
What are epiphytes?
Can an oyster change sex?
What goes on under a bridge?
Do fish sleep?
Why are they called stingrays?

To answer these questions and more...join us for the OOPSLA Special event at the Florida Aquarium. The Aquarium has galleries exploring wetlands, bays and beaches, coral reefs, and offshore mysteries. The newest and largest addition just opening is Sea Hunt, the unveiling of our largest addition ever! Sea Hunt featuring ocean predators from around the world! See a giant Pacific octopus! Expanded shark collection! Wolf eels! more!

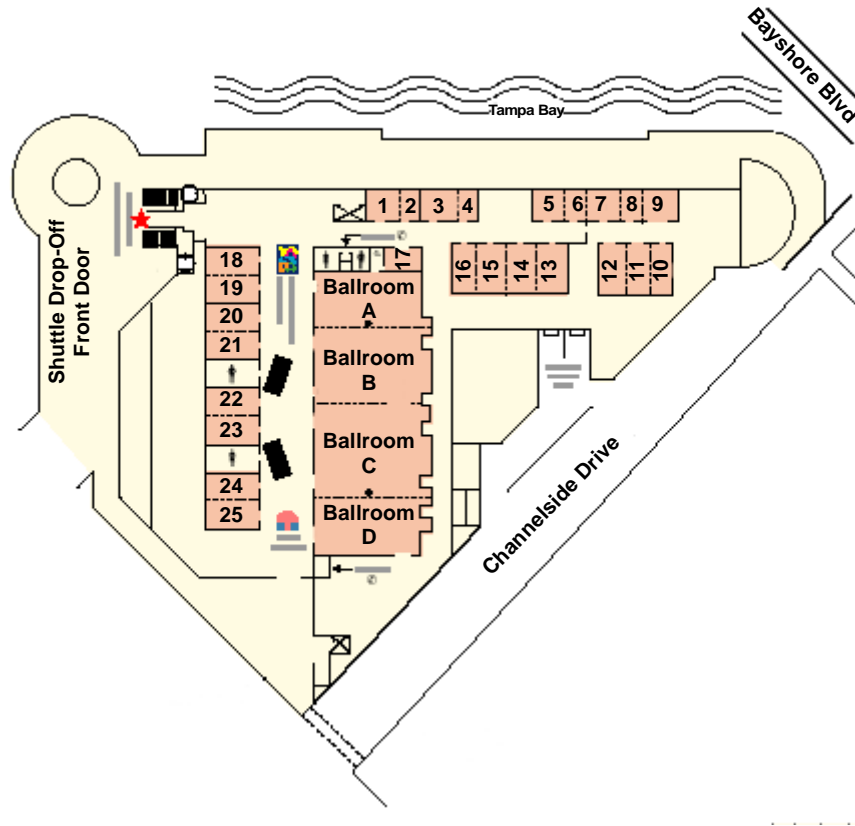
We will have a reception menu and beverages in addition to our own wildlife with dancing outside while enjoying the warm Florida evening.

INFORMATION AND MAPS

Hotel Locations

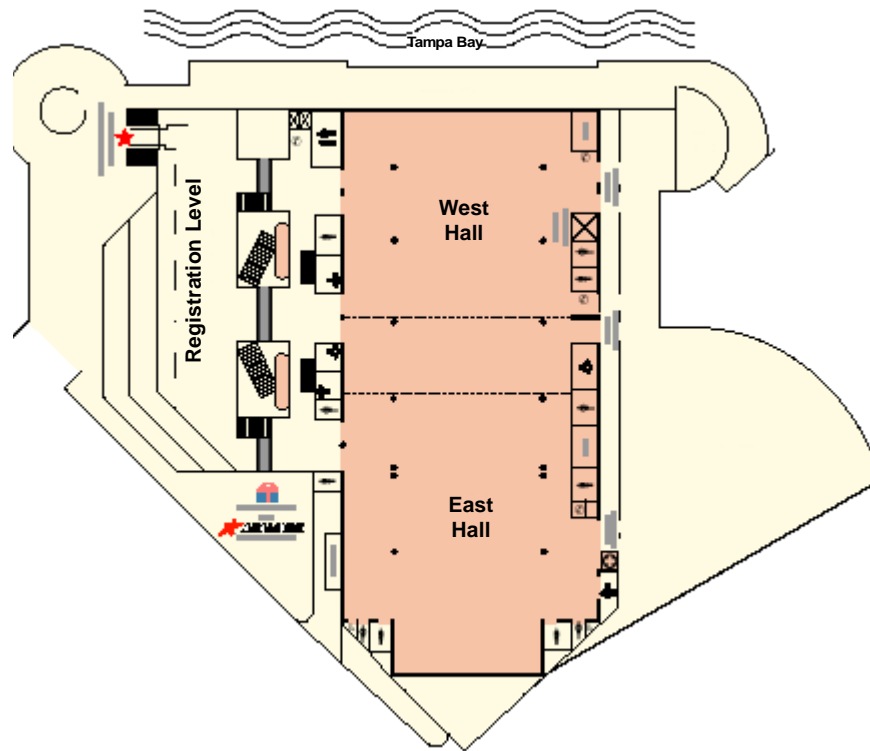


Tampa Convention Center

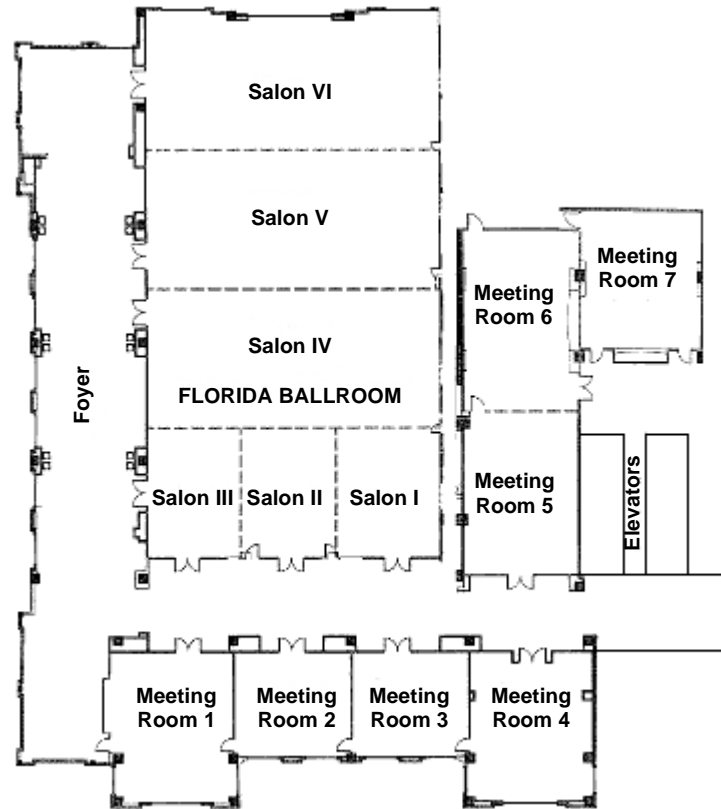


INFORMATION AND MAPS

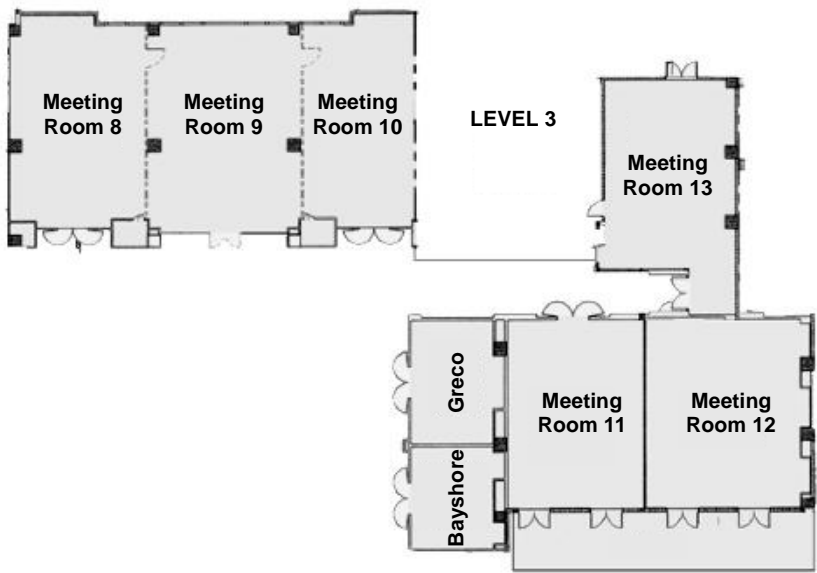
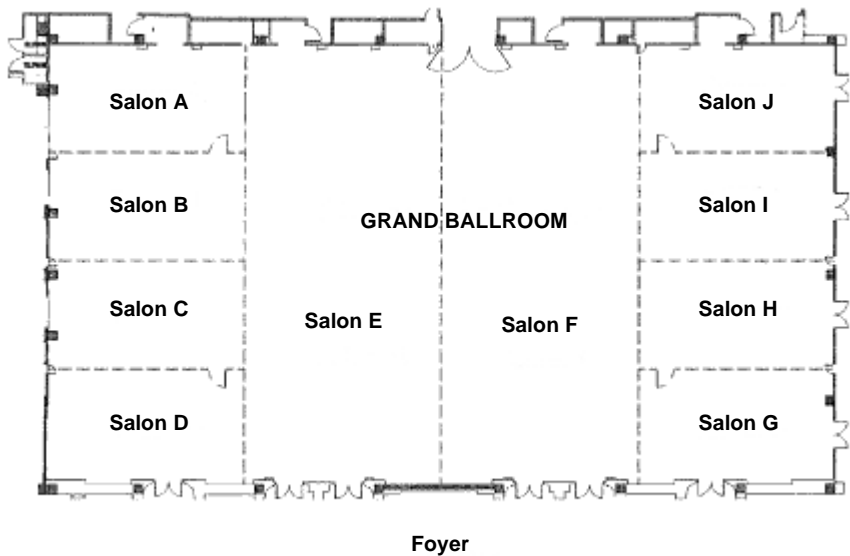
Tampa Convention Center



Marriott Waterside Hotel



Marriott Waterside Hotel



TAMPA FACTS

Tampa enjoys a temperate, semi-tropical climate marked by an average winter temperature of 62.5 degrees Fahrenheit and an average summer temperature of 81.4 degrees Fahrenheit. Year-round average rainfall totals 46.7 inches.

Tampa is located on Florida's west coast. Its central location makes it easy to get to the beaches and all the attractions throughout central Florida. Orlando is located just 84 miles east of the Tampa area. Hillsborough County is connected to Pinellas County and the beaches by the Courtney Campbell Causeway, Howard Frankland and Gandy bridges.

Hillsborough Bay and Tampa Bay border Tampa. Tampa Bay, Florida's largest open-water estuary, stretches 398 square miles at high tide. The county's two major rivers, the Hillsborough and the Alafia, empty into Hillsborough Bay. Land elevation in the county ranges from sea level to nearly 179 feet above sea level.

Tampa is Florida's third most populous city (following Miami and Jacksonville); Hillsborough is Florida's fourth most populous county.

Tampa	293,920
Hillsborough	998,948
Tampa Bay Area	2.5 million

Hillsborough County encompasses 1,048 square miles of land and 24 square miles of inland water area. Incorporated cities are Tampa, Temple Terrace and Plant City. Major unincorporated areas are Brandon, Ruskin, Apollo Beach, Dover, Valrico, Sun City Center and Ybor (EE-bore) City.

Once a small Indian fishing village, which later became known as the "Cigar City," Tampa Bay is now one of Florida's most invigorating travel destinations. In addition to its sandy beaches and beautiful sunsets, the Bay Area shines with entertainment and world-class attractions. Tampa Bay is home to rich history, opulent architecture and diverse Visitors can feel the pulse which beats in the streets of Ybor City -Tampa's nightlife capital and the steel drums and majestic scenes on the Serengeti Plain of Busch Gardens.

TAMPA FACTS

Major Visitor Attractions

- Busch Gardens - 4 million + (estimated)
- Historic Ybor City - 2.6 million
- Ice Palace arena - 1.5 million
- Cypress Gardens - 900,000 (estimated)
- Tampa Bay Performing Arts Center - 634,000+
- Lowry Park Zoo - 613,000+
- Museum of Science & Industry - 605,000+
- The Florida Aquarium - 591,000+
- Tampa Convention Center - 461,700+
- Adventure Island - 500,000+
- Roger's Christmas House - 350,000+
- Salvador Dali Museum - 225,000
- Florida International Museum - 200,000
- Bok Tower Gardens - 170,000
- Fantasy of Flight - 100,000
- Tampa Museum of Art - 81,000+
- Henry B. Plant Museum - 50,000 (1998)
- Canoe Escape - 18,000 (1998)

ACM / SIGPLAN / SIGSOFT



ACM, the Association for Computing Machinery, is a major force in advancing the skills

and knowledge of Information Technology (IT) professionals and students worldwide. ACM serves as an umbrella organization offering its 80,000 members a variety of forums in order to fulfill members' needs — the delivery of cutting-edge technical information, the transfer of ideas from theory into practice, and opportunities for information exchange. Providing high quality products and services — world-class journals and magazines; dynamic special interest groups; numerous “main event” conferences; tutorials; workshops; local special interest groups and chapters; and electronic forums - ACM is the resource for lifelong learning in the rapidly changing IT field. ACM also offers its members (both individual and institutional) the ACM Digital Library, an online resource with conference proceedings, journals and magazine archives combined with a state-of-the-art search engine.

SIGPLAN, ACM's Special Interest Group on Programming Languages explores programming language 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 (including OOPSLA), regular columns and technical correspondence. This SIG offers an additional newsletter (for FORTAN), on a subscription only basis. **SIGPLAN** sponsors four major annual conferences: the OOPSLA conference in 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 (ICFP).

SIGSOFT focuses on issues relating to all aspects of software engineering, providing a forum for computing professionals from industry, government and academia to examine principles, practices, education, and new research results in software engineering. **SIGSOFT** sponsors the Foundations of Software Engineering conference and co-sponsors the International Conference on Software Engineering. **SIGSOFT** also sponsors a variety of one-time and on-going workshops that bring practitioners, researchers, and educators together to discuss and debate timely issues. **SIGSOFT** publishes a bimonthly newsletter, “Software Engineering Notes,” which includes articles submitted by members as well as the popular forum “Risks to the Public,” which describes software safety mishaps and concerns.

Members of ACM, SIGPLAN, and SIGSOFT receive a discount when registering for OOPSLA.

Contact ACM to join!

Voice:

800-342-6626: US & Canada
+1-212-626-0500: NY & Global

Fax:

+1-212-944-1318

E-mail:

acmhelp@acm.org

Web:

<http://www.acm.org/>
<http://www.acm.org/sigplan/>
<http://www.acm.org/sigsoft/>

Committees

OOPSLA 2001 COMMITTEES

Conference Committee

Conference Chair:

Linda Northrop,
Software Engineering Institute

Program Chair:

John Vlissides,
IBM T.J. Watson Research Center

A/V Chair:

Tetsuhei Nakashima,
ITESM Campus Guadalajara

Communications Chair:

Laura Hill,
Sun Microsystems

Demonstrations Chair:

Peter Kriens,
aQute B.V.

DesignFest Chair:

Torsten Layda,
SWX Swiss Exchange

Doctoral Symposium Chair:

Doug Lea,
SUNY Oswego

Educators' Symposium Chair:

Jutta Eckstein,
Objects in Action, Germany

Internet Technology Chair:

Ken Bauer,
ITESM Campus Guadalajara

Panels Chair:

Ron Crocker,
Motorola, Inc.

Posters Chair:

Peri Tarr,
IBM T.J. Watson Research Center

Practitioner Reports Chair:

Jim Doble,
Tavve Software Company

Student Volunteers Chair:

Dirk Siebert,
Leipzig University, Germany

Treasurer:

Mamdouh Ibrahim,
IBM Global Services

Tutorials Chair:

Craig Larman,
Valtech

Workshop Chair:

Peter F. Sweeney,
IBM T.J. Watson Research Center

OOPSLA 2002 Chair:

Mamdouh Ibrahim,
IBM Global Services

Steering Committee Chair:

Brent Hailpern,
IBM T.J. Watson Research Center

Exhibits:

Jim Johnson,
Meetings & More, Inc

Liaison with ACM:

Donna Baglio,
ACM

Operations:

Nancy Wilson & Colleen Puderbaugh
Meeting Strategies Worldwide

Registration:

Carole Mann,
Registration Systems Lab

CyberChair Operation and Support:

Richard van de Stadt,
Borbala Online Conference Services

Typesetting:

Erin Sunahara,
Sunahara Consulting

Webmaster:

Steve Watson,
Watson Sites

Program Committee

John Barton,
Hewlett Packard Laboratories

Robert Binder,
RBSC Corporation

Toby Bloom,
Clinsoft Corporation

Craig Chambers,
University of Washington

Krzysztof Czarnecki,
DaimlerChrysler Research and Technology

Jim Doble,
Tavve Software Company

Brian Foote,
The Refactory, Inc.

Dennis Gannon,
Indiana University

Giovanna Guerrini,
University of Genova

Gerti Kappel,
Johannes Kepler University of Linz

Joe Kiniry,
California Institute of Technology

Jørgen Lindskov Knudsen,
Mjølnær Informatics

Doug Lea,
SUNY Oswego

Mary Loomis,
Commerce One

Satoshi Matsuoka,
Tokyo Institute of Technology

Gail Murphy,
University of British Columbia

Oscar Nierstrasz,
University of Berne

James Noble,
Victoria University of Wellington

Bran Selic,
Rational Software Canada

Peter F. Sweeney,
IBM T.J. Watson Research Center

Clemens Szyperski,
Microsoft Research

Rebecca Wirfs-Brock,
Wirfs-Brock Associates

Tutorial Committee

Craig Larman,
Valtech (Chair)

Sherman Alpert,
IBM T. J. Watson Research Center

Ron Crocker, *Motorola, Inc.*

Martin Fowler, *Thoughtworks*

Laura Hill, *Sun Microsystems*

Doctoral Symposium Committee

Doug Lea,
SUNY Oswego (Chair)

Brent Hailpern,
IBM T.J. Watson Research Center

James Noble,
Victoria University of Wellington, NZ

Mary Beth Rosson,
Virginia Tech

Richard Gabriel
Sun Microsystems

Ron Goldman
Stanford University

Practitioner Reports Committee

Jim Doble,
Tavve Software Company (Chair)

Nathaniel Talbott,
RoleModel Software, Inc.

Jamel Marzouki,
Motorola Labs

Scott A. Hissam,
Software Engineering Institute

Jeff Canna,
RoleModel Software, Inc.

Gerard Meszaros,
ClearStream Consulting

Scott E. Preece,
Motorola, Inc.

Granville Miller,
TogetherSoft

Ron Buskey,
Motorola, Inc.

Chris Carpenter,
RoleModel Software, Inc.

Committees

Educators' Symposium

Jutta Eckstein,
Objects in Action, Germany (Chair)

Dennis Mancl,
Lucent Technology, Bell Labs

Eugene Wallingford,
University of Northern Iowa

Helen Sharp,
City University London, UK

Joseph Bergin,
Pace University, New York

Markus Voelter,
Mathema AG, Germany

Nicolai Josuttis,
Solutions in Time, Germany

Robert Duvall,
University of Arizona, AZ

Workshop Committee

Peter F. Sweeney,
IBM T.J. Watson Research Center (Chair)

Peter Kriens,
aQuote

Darin Swanson,
Object Technology Inc.

Peri Tarr,
IBM T.J. Watson Research Center

Poster Committee

Peri Tarr,
IBM T.J. Watson Research Center (Chair)

Peter F. Sweeney,
IBM T.J. Watson Research Center

Doug Lea,
SUNY Oswego

Student Volunteer Captains

Maria Elena Chávez Echeagaray,
ITESM, Campus Guadalajara

Christopher Dutchyn,
University of Alberta

DesignFest Committee

Torsten Layda,
SWX Swiss Exchange (Chair)

Rob van den Berg,
Siemens VDO

Michael Feathers,
Object Mentor, Inc.

Gail E. Harris,
Instantiated Software Inc.

Jim Heliotis,
Rochester Institute of Technology

Alan O'Callaghan,
De Montfort University

OOPSLA Steering Committee

Brent Hailpern,
IBM T. J. Watson Research Center (Chair)

Mary Beth Rosson,
Virginia Tech

Linda Northrop,
Software Engineering Institute

Mamdouh Ibrahim,
IBM Global Services

Ron Crocker,
Motorola, Inc.

Lori Pollock,
University of Delaware

Kathryn S. McKinley,
University of Massachusetts, Amherst

Donna Baglio,
Director, ACM Office of SIG Services

OOPSLA 2002

17th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications

November 4-8, 2002
Seattle, Washington, USA

On behalf of the OOPSLA 2002 Conference Committee, I would like to invite you to contribute to and actively participate in next year's OOPSLA. The conference will be held in Seattle, Washington, November 4-8, 2002.

OOPSLA is the premier forum for bringing together practitioners, researchers, and students to share their ideas and experiences in a broad range of disciplines woven with the common thread of object technology. Whether you are an OOPSLA veteran or attending for the first time, you probably realize by now that this conference is a unique annual event. It is a well-integrated collage of activities, including:

- Outstanding invited speakers
- Carefully refereed technical papers that will be archived in the ACM Digital Library
- Real world experiences in the form of practitioner reports
- Exciting panels
- Comprehensive tutorial program
- Topic-focused workshops
- Symposia specifically designed for educators and for doctoral students
- Late-breaking demonstrations
- Interactive posters session
- Birds-of-a-feather sessions for informal discussions, and
- Plenty of social opportunities for mingling and professional networking

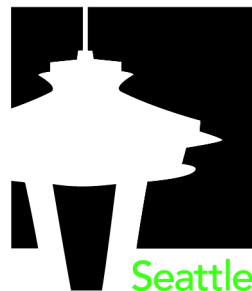
Such a variety of activities make OOPSLA the conference of choice for a broad range of professionals: from recognized academics to undergraduate students, from industrial researchers to technical developers and managers, and from technology gurus to new users ready to explore this exciting field.

You may not realize that becoming an active participant in OOPSLA is much easier than you think. All it takes is to identify the events that match your interest and the status of your work, read the submission requirements carefully, and submit your paper or proposal using the OOPSLA on-line submission system. Also, the committee is interested in your ideas for new venues and activities that help OOPSLA continue to evolve to better serve its community. If you have ideas for one-day symposia, student activities, or other activities, please submit a proposal to the Conference Chair for consideration.

I strongly encourage you to become a contributor to OOPSLA 2002 and take part in shaping the future of the conference for years to come. For more information about actively participating in OOPSLA 2002, and for submission deadlines and details, please visit the OOPSLA 2002 web page:

<http://oopsla.acm.org/OOPSLA2002> or contact the OOPSLA 2002 Office

OOPSLA 2002



OOPSLA 2002 Important Dates

March 22, 2002 — Submission deadline for Technical Papers, Practitioner Reports, Tutorials, Panels, Workshops, Educators' Symposium, and DesignFest problems.

July 9, 2002 — Submission deadline for Posters, Demonstrations, Doctoral Symposium, and Student Volunteers.

Conference Chair: Mamdouh Ibrahim, *IBM Global Services*

Program Chair: Satoshi Matsuoka, *Tokyo Institute of Technology*

Seattle

OOPSLA 2002 Office — 6220 NE Glisan — Portland, OR 97213

Voice: +1-503-252-5709 — Fax: +1-503-261-0964 — E-mail: oopsla02@acm.org