ICSE 2003
25th International Conference on Software Engineering

http://www.icse-conferences.org/2003/

The Hilton Portland Hotel, Portland, Oregon
May 3-10, 2003

Silver Anniversary Advance Program

Welcome to ICSE 2003, Software Engineering Week in Portland!

A Message from the Chairs

Managing the complexity of modern software systems is without question a grand challenge for software engineering. It is therefore fitting that the theme for the silver anniversary of the International Conference on Software Engineering is: "Scaling New Heights." World leaders in software engineering research, practice, and education will gather this May, inspired by the view of Mount Hood, to present and discuss the most recent advances, trends, and concerns in this ever expanding and critical field.

In the tradition of past meetings, ICSE 2003, the premier conference for software engineering, will feature the latest inventions, achievements, and experiences in software engineering research and practice, and will give researchers, practitioners, and educators the opportunity to present, discuss, and learn. The Software Engineering Week, May 3—10, 2003, consists of the main ICSE conference and an interesting array of tutorials, workshops, and related events associated or co-located with ICSE. The conference venue is the Hilton Hotel, centered in the heart of Portland's entertainment and cultural district, with access to performing arts, shopping, museums, coffee houses, microbreweries, and numerous restaurants, all within three blocks.

Each morning of the ICSE 2003 conference starts with a topical keynote address by an outstanding speaker. These talks kick off an exciting program that includes 42 technical papers, 16 experience reports, 11 education papers, and a variety of mini-tutorials, panels, and formal research demonstrations. Some notable highlights of this program are: three mini-tutorials on new and promising software engineering technologies in the Frontiers of Software Practice (FoSP) track; a mini-tutorial on how to write a good research paper in software engineering; and participation from the automotive community in a session on automotive software engineering. Throughout the conference, there are also exhibits, posters, and informal research demonstrations, as well as an onsite newsletter describing memorable moments, humor, and fascinating facts. Finally, the conference features several breaks, lunches, and receptions providing opportunities to meet and mingle with old and new friends.

Prior to and immediately following the main ICSE 2003 program, there are 17 tutorials (full day and half day) on a variety of topics and 15 workshops that offer a forum for interaction. There are also three special events: the Pioneers Symposium, the New Software Engineering Faculty Symposium, and the Doctoral Symposium. Finally, the week also includes five co-located workshops and events: the Workshop on Software Configuration Management (SCM-11), the International SPIN Workshop on Model Checking of Software (SPIN 2003), the Workshop on Software Process Simulation Modeling (ProSim 2003), the International Workshop on Program Comprehension (IWPC 2003), and the Summit on Software Engineering Education.

We hope that you will also have time, perhaps before or after this busy Software Engineering week, to explore the "Rose City" and the many recreational opportunities of the Pacific Northeast.

We cordially invite you to participate in ICSE 2003. Please join us in a memorable week of presentations, discussions, demonstrations, and learning.

Lori A. Clarke
General Chair

Laura K. Dillon
Program Co-Chair

Walter Tichy
Program Co-Chair

Table of Contents

Keynote Speakers

Components That You Can Trust — *Tuesday, May 6*

Bertrand Meyer, ETH, Zürich, and ISE Santa Barbara, USA

Component-based development, one of the most promising paths of progress for the world of software engineering, is fraught with risks if it isn't accompanied by a constant concern for quality. Components of demonstrably high quality may, on the other hand, bring a critical contribution to the improvement of both software products and the software process. This presentation will address the challenge of building "trusted components" whose quality can be guaranteed. It will discuss both the "low road" of certifying components built with current technologies and the "high road" of proving component properties.

Bertrand Meyer is Professor of Software Engineering at the ETH (Swiss Federal Institute of Technology) and Scientific Advisor of ISE, the company he co-founded in 1985. He is the author of a number of books including "Object-Oriented Software Construction, 2nd edition," "Eiffel: The Language" and "Reusable Software." He has been involved in the design of numerous libraries and tools applying the principles of "Design by Contract" and object technology.

Must There Be So Few? Including Women in CS — *Wednesday, May 7*

Joanne McGrath Cohoon, Department of Leadership, Foundations and Policy, University of Virginia, USA

Women's participation in undergraduate computing is low and likely to continue declining. This situation is not due to intractable gender differences, however. It has been demonstrated that academic computing departments can effectively recruit and retain female students. Dr. Cohoon will describe the current state of affairs and discuss how and why departments can act to reverse this trend.

Joanne McGrath Cohoon is a sociologist who studies higher education, gender, and technology. She earned her BA in Philosophy from Ramapo College of New Jersey; her MA in Student Personnel Administration from Teacher's College, Columbia University; and her Ph.D. in Sociology from the University of Virginia in 2000. Dr. Cohoon has held professional positions in higher education as a researcher, administrator, and instructor at a women's college, a survey research center, a center for public service, and a continuing education program. She is currently a Research Assistant Professor in the Curry School of Education at the University of Virginia. Her research has been funded by the Alfred P. Sloan Foundation and the National Science Foundation. She is a member of the ACM, SIGCSE, and sociological and higher education professional organizations.

Relating Software Engineering and Information Security — *Thursday, May 8*

Eugene Spafford, Purdue University, USA

There are many connections between software engineering(SE) and information security(infosec). Some are obvious, such as the process of detecting software faults, and some are more subtle, such as definition and capture of privacy requirements. In both infosec and SE there are complex challenges of how best to balance cost, design, technology, and time to market: Too often, good practices are skipped because of cost or time. Meanwhile, failures in both areas can lead to everything from minor inconvenience to catastrophic failures and compromises. In this talk, I intend to explain some of the connections I see between SE and infosec. In particular, I hope to illustrate how some of the challenges -- and advances -- in infosec have a

basis in SE. Some of these suggest high-leverage areas of research, while others provide insight about why we will continue to experience security problems in widely deployed software. For instance, is there truth to the contention that open source software is more secure than proprietary source? Along the way, I will connect Las Vegas, the PDP-11, Roman chariots, and a common security flaw as one illustration of how unintended consequences shape both security and software development.

Eugene H. Spafford is a professor of Computer Sciences at Purdue University, a professor of Philosophy (courtesy appointment), and is Director of the Center for Education and Research in Information Assurance and Security (CERIAS). Spaf's research career has included work in information security, software engineering, distributed systems, and professional ethics. Dr. Spafford is a Fellow of the ACM, Fellow of the AAAS, Fellow of the IEEE, and is a charter recipient of the Computer Society's Golden Core award. He was the year 2000 recipient of the NIST/NCSC National Computer Systems Security Award, generally regarded as the field's most significant honor in information security research. In 2001, he was elected to the ISSA Hall of Fame, and he was awarded the William Hugh Murray medal of the NCISSE for his contributions to research and education in infosec. Among his many activities, Spaf is co-chair of the ACM's U.S. Public Policy Committee and of its Advisory Committee on Computer Security and Privacy, is a member of the Board of Directors of the Computing Research Association, and is a member of the US Air Force Scientific Advisory Board.

Frontiers of Software Practice (FoSP)

FoSP talks are mini-tutorials that provide an overview of a topic at the frontier of software practice. The FoSP topics were selected based on their interest to a wide range of participants. We are very excited to have the following speakers give us an overview of three frontier topics that software researchers and practitioners grapple with on a daily basis.

Component Technology - What, Where, and How? *Tuesday, May 6*

Dr. Clemens Szyperski, Microsoft Corporation

Software components, if used properly, offer many software engineering benefits. Yet, they also pose many challenges starting from quality assurance and ranging to architectural embedding and composability. In addition, the recent movement towards services as well as the established world of objects cause many to wonder what purpose components might have. This talk offers an end-to-end overview of what components should do, where they should be used, and how this can be achieved.

Clemens Szyperski is a Software Architect with Microsoft—which he joined in 1999—and affiliated with Microsoft Research, both in Redmond, WA. He is also an adjunct professor at Queensland University of Technology in Brisbane, Australia. He received his PhD in computer science from the Swiss Federal Institute of Technology (ETH) in Zurich, Switzerland under Prof. Niklaus Wirth. Clemens is the award-winning author of "Component Software - Beyond Object-Oriented Programming", which is recently being released in its second edition. He is also the co-author of "Software Ecosystem - An Indispensable Industry and Technology", to appear later this year. In addition, he has published several other books and many articles, and he frequently presents at international events. Clemens has worked as an academic, researcher, and practitioner in the areas of programming languages, component technologies, software systems, and software architecture. He is a co-founder of Oberon Microsystems in Zurich, specializing in component technology and software architecture. He has served as a consultant to large corporations and serves as an assessor and reviewer for domestic, federal, and international funding agencies and for learned journals across the globe. He has been a member of program and organizing committees of numerous events, including many of the most prestigious conferences in his discipline areas.

Patterns, Frameworks, and Middleware: Their Synergistic Relationship

Dr. Douglas Schmidt, Vanderbilt University, DARPA **Wednesday, May 7**

Historically, the knowledge required to develop mission-critical software systems has largely existed in programming folklore, the heads of experienced researchers and developers, or buried deep within complex source code. Today's methods and tools for software modeling help somewhat by capturing *how* a system is designed. They only automate limited portions of software development, however, and do not articulate *why* a system is designed in a particular way, which may complicate software evolution and optimization.

Patterns, frameworks, and middleware are increasingly popular techniques for addressing the challenges outlined above. Patterns codify design expertise that provides time-proven solutions to commonly occurring software problems that arise in particular contexts. Frameworks provide (1) a reusable architecture—based on patterns—for a family of related applications and (2) an integrated set of collaborating components that implement concrete realizations of the architecture. Middleware is a class of software that leverages patterns and frameworks to increase systematic reuse significantly by bridging the gap between the end-to-end functional requirements of applications and the underlying operating systems and network protocol stacks.

The relationship between patterns, frameworks, and middleware is highly synergistic. For example, patterns help guide framework creation and use, thereby reducing development effort and training costs. In turn, frameworks can be used to develop middleware, whose interfaces then provide a simpler facade for the complex internal component structure of the frameworks. This talk describes the synergy between patterns, frameworks, and middleware and illustrates how they have been applied successfully in many production mission-critical software systems.

Dr. Douglas C. Schmidt is a Full Professor in the Electrical Engineering and Computer Science Department at Vanderbilt University. His research focuses on patterns, optimization techniques, and empirical analyses of object-oriented frameworks that facilitate the development of high-performance, real-time distributed object computing middleware on parallel platforms running over high-speed networks and embedded system interconnects. In addition to his academic research, Dr. Schmidt has over fifteen years of experience developing object-oriented middleware, in particular ACE and TAO, which are widely-used open-source middleware frameworks that implement patterns for high-performance and real-time systems. Dr. Schmidt also serves as a program manager in the DARPA Information Exploitation Office (IXO), where he leads the United States effort on distributed object computing middleware research.

CyberSecurity: What are Best Practices? **Thursday, May 8**

Dr. Richard Kemmerer, University of California at Santa Barbara

As more business activities are being automated and an increasing number of computers are being used to store vital and sensitive information, the need for secure computer systems becomes apparent. This need is even more apparent as systems and applications are being distributed and access is via an insecure network. Secure systems and networks can be obtained only through systematic development; they cannot be achieved through haphazard seat-of-the-pants methods.

The pervasive use of computer and network technologies in all walks of life has turned cybersecurity issues into national security issues. The Internet has become critical for governments, companies, financial institutions, and millions of everyday users. Networks of computers support a multitude of activities whose loss would all but cripple these organizations. Protecting these critical infrastructures is a difficult task.

This talk introduces some known threats to cybersecurity, categorizes the threats, and analyzes protection mechanisms and techniques for countering the threats. Approaches to prevent, detect, and respond to cyber attacks will also be discussed.

Richard A. Kemmerer is a Professor and past Chair of the Department of Computer Science at the University of California, Santa Barbara. He is a Fellow of the IEEE Computer Society, a Fellow of the Association for Computing Machinery, and past Editor-in-Chief of IEEE Transactions on Software Engineering. Dr. Kemmerer has chaired or served on many program committees and was the program co-chair of the 20th International Conference on Software Engineering (ICSE'98). He has

served as a member of the National Academy of Science's Committee on Computer Security in the DOE, the System Security Study Committee, the Committee for Review of the Oversight Mechanisms for Space Shuttle Flight Software Processes, the Committee on Maintaining Privacy and Security in Health Care Applications of the National Information Infrastructure, and the Committee on the Review of Programs for Command, Control, Communication, Computers, and Intelligence (C4I) in the Department of Defense. He has also served as a member of the National Computer Security Center's Formal Verification Working Group and was a member of the NIST's Computer and Telecommunications Security Council. Dr. Kemmerer is also the past Chair of the IEEE Technical Committee on Security and Privacy and a past member of the Advisory Board for the ACM's Special Interest Group on Security, Audit, and Control. He has written numerous papers on the subjects of computer security, formal specification and verification, software testing, programming languages, and software complexity measures. He is the author of the book "Formal Specification and Verification of an Operating System Security Kernel" and a co-author of "Computers at Risk: Safe Computing in the Information Age." He has been a Principal Investigator on numerous government and private sector sponsored projects and leads the Reliable Software Group at UCSB. Under his direction the Reliable Software Group has addressed the need for better languages and tools for designing, building, validating, and securing software systems.

Doctoral Symposium — Monday, May 5

Dick Hamlet, Portland State University, USA
Mary Lou Soffa, University of Pittsburgh, USA

The Doctoral Symposium is a forum for Ph.D. students to publicly discuss their research goals, methods, and results at an early stage in their research. The Symposium aims to provide useful guidance for completion of the dissertation research and for initiation of a research career by experienced faculty members in a supportive setting. The Symposium and ICSE also provide an opportunity for student participants to interact with established researchers and others in the wider software engineering community. Interested students must apply; see the conference web page for application details.

New Software Engineering Faculty Symposium (NSEFS 03)

Leon J. Osterweil, University of Massachusetts, USA — **Monday, May 5**

The challenge of becoming a new Assistant Professor at a research university is one that most people accept with enthusiasm and energy, but also with some trepidation. One's tenure track appointment is a signal accomplishment and the culmination of many years of hard work. But, simultaneously, it signals the start of another round of hard work and challenges that are new and often unfamiliar. Among these new challenges are establishing an independent research program, learning how to teach, learning how to mentor, and deciding how to balance career and personal life. For a new software engineering faculty member there are additional challenges such as obtaining financial support, supervising the development of software systems, and dealing with skeptical faculty colleagues from other disciplines. The purpose of NSEFS 03 is to help new software engineering faculty members to feel more comfortable and confident in dealing with these many challenges. NSEFS 03 will feature presentations by leading academic software engineering researchers who will provide advice and guidance based upon their personal experiences and insights into the contemporary academic software engineering community. There will also be ample time for informal and small group interactions with the presenters and other attendees. NSEFS 03 is designed to be a service to those who are within two years of (either before or after) their initial tenure track academic appointment, although the symposium may be of significant interest and value to others. Enrollment is limited.

Pioneers Symposium — Sunday, May 4

Larry Votta, Sun
Stuart Faulk, University of Oregon

The Pioneers Symposium will provide a perspective on and bridge between the first 25 ICSE's and the next 25. Pioneers in Software Design, Process, Economics, Empiricism, and Safety Critical Systems will share their views in presentations and discussions on where we have been, where the field is going, and what software engineers should know. The Symposium provides the opportunity for participants, particularly students and new faculty, to learn from and interact with some of the field's most influential pioneers. A reception after the talks will encourage interaction between established mentors in the field and software engineering's next generation. Speakers will include Barry Boehm, Michael Jackson, Nancy Leveson, and David Parnas. Enrollment is limited.

Managing Commitments and Risks: Challenges in Agile Development: A Co-operative Learning Experience — Tuesday, May 6

Jyrki Kontio, Helsinki University of Technology, Finland
Magnus Höglund, TietoEnator, Espoo, Finland
Jan Rydén, TietoEnator, Espoo, Finland
David Raffo, Portland State University, USA

This co-operative learning event is entertaining and eye-opening. In it, participants experience human and organizational dimensions of software development. The purpose is to highlight and explore the challenges in commitment management and risk management in agile software development. In addition to providing concise briefings on risk management, commitment management, and agile software development, this event will use teamwork and role-playing so that all participants experience and discover the new insights and challenges of modern software management. Many practical challenges in software engineering are not limited to technological issues. Managerial issues, communications, personnel relationships, and skills management often have a substantial impact on a software project's success. This event explores and studies these issues of software management from three specific perspectives: how commitments are made and managed, how risks are managed, and what specific challenges face the agile software development context. The event aims to share best practices in the agile development area; identify challenges and solutions; help participants establish contacts with each other; and explore new ways of learning, using the unconventional format. After the event, the participants will have a greater understanding and appreciation of these challenges through reflection on and sharing of their own experiences. The event is targeted mainly for software project and program managers, software engineers, and researchers in software engineering. Enrollment is limited.

ICSE 2003 Awards Wednesday, May 7

ICSE 2003 is proud to be the venue for presentation of awards to people who have made outstanding contributions to the field of software engineering.

SIGSOFT Distinguished Service, Outstanding Research Award and ACM Fellows

Awards presented by A. Wolf

IEEE Fellows

Awards presented by K. Reed

Recognition Award from Journal of Systems and Software to Robert Glass

Award presented by D. Card

Most Influential Paper Award from ICSE'15 to Bashar Nuseibeh, Jeff Kramer, and Anthony Finkelstein

Award presented by D. Perry

Workshops

To maintain an informal environment and encourage discussion, workshops have limited attendance. Once the enrollment cap is reached, enrollment will be closed, so those interested in attending a workshop should register early.

W1: SELMAS'03 - 2nd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems Saturday, May 3 - Sunday, May 4

Jose Alberto R. P. Sardinha, PUC-Rio - Brazil
Alessandro Garcia, PUC-Rio - Brazil
Carlos José P. de Lucena, Pontifocia Universidade Catalica (PUC-Rio)
Alexander Romanovsky, University of Newcastle
Donald Cowan, University of Waterloo
Paulo S.C. Alencar, University of Waterloo
Jaelson F. B. Castro, Universidade Federal de Pernambuco - UFPE

Advances in networking technology have revitalized the investigation of agent technology as a promising paradigm to engineer complex distributed software systems. Nowadays, agent technology has been applied in a wide range of application domains, including e-commerce, human-computer interfaces, telecommunications, and concurrent engineering. Software agents can be viewed as complex objects with an attitude. Like objects, agents provide a specific set of services for their users. However, software agents are inherently more complex abstractions, so that development of multi-agent systems (MASs) poses other challenges to software engineering. A single agent is driven by beliefs, goals, plans, and various behavioral properties, such as autonomy, adaptation, interaction, collaboration, learning and mobility. Each of these features adds complexity to system modeling, design and implementation. In addition, a large-scale MAS encompasses multiple types of agents, each of them having distinct agency properties, and it needs to satisfy multiple stringent requirements, such as reliability, security, adaptability, interoperability, scalability, maintainability, and reusability. Many existing agent-oriented solutions are far from ideal; in practice, they are often built in an ad-hoc manner and are error-prone, not scalable, and not generally applicable to large MAS. The workshop will cover a wide range of topics of software engineering for large-scale multi-agent systems, from theoretical foundations to empirical studies.

W2: EDSER-5 - 5th International Workshop on Economics-Driven Software Engineering Research: The Search for Value in Engineering Decisions Saturday, May 3 - Sunday, May 4

Jyrki Kontio, Helsinki University of Technology, Finland
Shawn Butler, Carnegie Mellon University

The EDSER-5 workshop aims at improving the state-of-the-art and state-of-practice in economics and business driven software engineering in order to ensure that the engineering decisions made during software development fully address the relevant economic and business issues. While these technical engineering decisions form the foundation of software engineering, these decisions should be guided by the objective of adding value to stakeholders. The workshop will give an up-to-date view of the recent research in this area and it will also act as a forum where practitioners can exchange views with leading researchers.

W3: CBSE6 - 6th Workshop on Component-Based Software Engineering Automated Reasoning and Prediction Saturday, May 3 - Sunday, May 4

Ivica Crnkovic, Mälardalens University
Heinz Schmidt, Centre for Distributed Systems and Software Engineering, Monash University
Judith Stafford, Tufts University
Kurt C. Wallnau, Software Engineering Institute, Carnegie Mellon University

Component-based technologies and processes have been deployed in many organizations and fields over the past several years. Measurable gains in design flexibility and development productivity have been demonstrated where software component technology has been combined with software architecture and product line practices. However, modeling, reasoning about, and predicting component and system properties remains challenging in theory and in practice. CBSE6 builds on previous workshops in the ICSE/CBSE series, this year thematically centered on automated composition theories. Composition theories support reasoning about, and predicting, the runtime properties of assemblies of components. In this workshop emphasis is placed on composition theories that are well founded theoretically, automated by tools, and/or supported by evaluation. Both empirical and formal theories of composition are of interest. Issues related to composition theory and practice include: determining what properties are of interest in a given domain, how to model these properties, how to reason about and with property models, how to measure component and assembly properties, how to verify the measurements and predictions, and how to communicate property values and composition theories to component users. Resolving these issues requires collaborative work of researchers in software engineering, computer science, and other disciplines. The primary goal of CBSE6 is to achieve better understanding of the state of the art in automated compositional reasoning and prediction. While emphasizing state of the art, the workshop aims at bridging theory and practice.

W4: Bridging the Gaps Between Software Engineering and Human-Computer Interaction Saturday, May 3 - Sunday, May 4

Rick Kazman, Software Engineering Institute, Carnegie Mellon University
Len Bass, Software Engineering Institute, Carnegie Mellon University
Jan Bosch, University of Groningen, Netherlands

Almost half of the software in systems being developed today and over a third to fifty percent of the effort expended throughout the software life cycle relate to the systems' user interfaces. For this reason, problems and methods from the field of human-computer interaction (HCI) have tremendous impact on the overall process of software engineering (SE), and vice versa. Yet despite these powerful reasons to practice and apply effective SE and HCI methods, there still exist major gaps of understanding both between suggested practice and actual practice and between the best practices of the two fields. The respective curricula seldom reference each other, and do not teach the respective communities how to interact with one another. The architectures, processes, methods and vocabulary being used in each community are often foreign to the other community. As a result, product quality is not as high as it could be, and (avoidable) re-work is frequently necessary. This workshop will bring together practitioners and academics in the two fields in an attempt to enumerate and understand these gaps between SE and HCI, with an eventual goal of proposing practical ways—shared processes, shared architectures, shared notations, etc.—to bridge these gaps. The workshop will focus on: increasing awareness of the issues in the world at large; designing joint or related curricula; creating unified tools, methods, and processes; and influencing regulations and/or

conventions of practice. The tangible results of the workshop will be a practical program of education, research, and public relations focused on changing the way that people think about these two fields, and the way that the fields are actually practiced.

W5: First Workshop on Requirements Engineering in Open Systems

Saturday, May 3 - Sunday, May 4

Stephen Fickas, University of Oregon
Robert J. Hall, AT&T Labs Research, Florham Park, NJ

This workshop will bring together researchers and practitioners in requirements engineering, component-based design (including Enterprise Application Integration (EAI)), verification and validation, and related fields to discuss the challenges of designing and using open systems in which requirements are ephemeral and user-relative and in which it is difficult or impossible to know the behaviors of all the parts of the system. The goals for the workshop are (1) to improve awareness and understanding of how open systems create novel problems for requirements engineering and (2) to begin to explore potential solutions. To help focus the discussion, we have selected some open system scenarios and we encourage each presentation to discuss how its ideas address or relate to the problems illustrated in the scenarios. The format of the presentations will include extra time for audience discussion of each presentation, hopefully allowing the group both to better understand each set of ideas and to relate them to other presentations and to the workshop themes.

W6: Workshop on Software Variability Management

Saturday, May 3

Peter Knauber, Mannheim University of Applied Sciences, Mannheim, Germany
Jan Bosch, University of Groningen, Groningen, Netherlands

In a variety of approaches to software development, software artifacts are used in multiple contexts or for various purposes. The differences lead to so-called variation points in the software artifact. During recent years, the amount of variability that has to be supported by a software artifact is growing considerably and the management of variability is developing as a main challenge during development, usage, and evolution of software artifacts. So far, variability management is recognized as a crosscutting concept that has a key role in various areas but it is poorly understood as an issue in its own right. In different facets, variability management is part of many recent development approaches, including but not limited to object-oriented frameworks, design patterns, domain-oriented languages, generative programming, generic components, domain and requirements analysis, and software product families (also called software product lines). Successful management of variability in software artifacts leads to more customizable software products that are in turn likely to result in higher market success: in the information systems domain, the products are more easily adaptable to the needs of different user groups; in the embedded systems domain, the software can be more easily configured to work with different hardware and environmental constraints. This workshop will explicitly address the management of variability in software artifacts from its different perspectives, bringing together representatives from different academic and industrial communities to share their experiences and ideas.

W7: Workshop on Software Architectures for Dependable Systems

Saturday, May 3

Rogério de Lemos, University of Kent at Canterbury, UK
Cristina Gacek, University of Newcastle upon Tyne, UK
Alexander Romanovsky, University of Newcastle upon Tyne, UK

Architectural representations of systems have been shown to be effective in assisting the understanding of broader system concerns by abstracting away details of the system. Dependability has become an important aspect of computer systems since everyday life increasingly depends on software. Although there is a large body of research in dependability, architectural level reasoning about dependability is only just emerging as an important theme in software engineering. The complexity of emerging applications and the trend of building trustworthy systems from existing, untrustworthy components are urging that dependability concerns be considered at the architectural level. This workshop will bring together members of the communities of software architectures and dependability to discuss the state of research and practice when dealing with dependability issues at the architectural level.

W8: 2nd International Workshop on Scenarios and State Machines: Models, Algorithms, and Tools — Saturday, May 3

Sebastian Uchitel, Imperial College, UK
Francis Bordeleau, University of Carleton, Canada
Alexander Egyed, Teknowledge Corporation, US
Martin Glinz, University of Zurich, Switzerland
Jeff Kramer, Imperial College, UK
Ingolf Krüger, University of California at San Diego, USA
Axel van Lamsweerde, U. Louvain, Belgium
Stefan Leue, University of Freiburg, Germany
Wilhelm Schäfer, University of Paderborn, Germany
Tarja Systä, University of Tampere, Finland
Jon Whittle, NASA Ames, USA
Albert Zündorf, University of Braunschweig, Germany

Behavior modeling plays an important role in the engineering of software-based systems; it is the basis for systematic approaches to requirements capture, specification, design, simulation, code generation, testing, and verification. A range of notations, techniques and tools supporting behavior modeling for these development tasks exists. Two complementary approaches for modeling behavior have proven useful in practice: state- and scenario-based modeling. UML statecharts have become popular as a technique for describing the intended behavior of class instances in object-oriented systems. State-based formalisms are also widely used for modeling distributed and real-time systems, in particular because the corresponding models can be rigorously analyzed using model checking. Practitioners also use scenario-based notations and tools extensively; here, the focus of concern shifts from the complete behavior specification for individual components or objects to the (partial) specification of component collaboration. Exploring the relation between scenarios and state machines can lead to new areas of research and to tools that can exploit the best of both worlds. This second workshop on scenarios and state machines has been motivated by the very successful first workshop on this topic at ICSE'02.

W9: 3rd Workshop on Open Source Software Engineering — Saturday, May 3

Joseph Feller, University College Cork, Ireland
Brian Fitzgerald, University of Limerick, Ireland
Scott A. Hissam, Software Engineering Institute, USA
Karim Lakhani, MIT Sloan School of Management, USA

Building on the success of the 1st and 2nd Workshops on Open Source Software Engineering (ICSE'01 and ICSE'02), the 3rd Workshop on Open Source Software Engineering will bring together researchers and practitioners for the purpose of discussing the diverse array of techniques, as well as supporting tools and social/organizational contexts, that can be observed in the domain of open source software (OSS). Despite the constantly growing body of research on OSS, our knowledge of the open source family of software engineering processes is far from complete, a point raised repeatedly by the participants of the 2nd workshop in this series. To address this knowledge gap, this workshop invites participants to bring to the conversation empirical descriptions and contribute to informed discussions of the key tools, techniques, and contexts currently used by OSS development communities. Discussions will address the processes that are involved in critical software engineering activities, such as requirements gathering, analysis and design, leadership and decision making; coordination and collaboration; configuration management; testing and quality assurance; release management; debugging; documentation and support; translation and localization.

W10: Software Engineering for High Assurance Systems: Synergies between Process, Product, and Profiling

Friday, May 9 - Saturday, May 10

Martin Feather, Jet Propulsion Lab, USA
Constance Heitmeyer, Naval Research Lab, USA
Nancy Mead, Software Engineering Institute, Carnegie Mellon University, USA
Allen Nikora, Jet Propulsion Lab, USA

This two-day workshop will bring together researchers and practitioners interested in the development of an engineering method for constructing and evaluating software for high assurance systems. A high assurance system is a system where compelling evidence is required that the system delivers its services in a manner satisfying certain critical properties, such as security, safety, fault-tolerance, and survivability. Examples of high assurance systems include safety-critical medical systems, control systems for nuclear plants, and aerospace systems. The workshop participants will explore the opportunities for, and benefits of, synergies between different themes, each addressing aspects of the problem of high assurance software development. These themes are: Process, Product, and Profiling. The purpose of this workshop is to find synergies between the themes and discover where crossover work can lead to advances that might otherwise go unexplored.

W11: ACSE 2003 - 3rd International Workshop on Adoption-Centric Software Engineering

Friday, May 9

Robert Balzer, Teknowledge Corporation, USA
Jens Jahnke, University of Victoria, Canada
Marin Litoiu, IBM Canada Ltd., Canada
Hausi A. Müller, University of Victoria, Canada
Dennis B. Smith, Carnegie Mellon Software Engineering Institute, USA
Margaret-Anne Storey, University of Victoria, Canada
Scott R. Tilley, Florida Institute of Technology, USA
Ken Wong, University of Alberta, Canada

Understanding adoption of software engineering tools and practices is critical for the software and information technology sectors, which are continually challenged to increase their productivity. This workshop will bring together researchers and practitioners who investigate innovative solutions to software engineering adoption issues. The key objective of this workshop is to explore approaches where software engineering tools and practices are implemented as extensions of Commercial Off The Shelf Software (COTS) products and middleware technologies that work in conjunction with software engineering tools as well as mined components. The workshop aims to advance the understanding and evaluation of adoption of software engineering tools and practices.

W12: RAMSS - Workshop on Remote Analysis and Measurement of Software Systems

Friday, May 9

Alessandro Orso, Georgia Institute of Technology, USA
Adam A. Porter, University of Maryland, USA

The manner in which software is produced and used is changing radically. Not long ago, software systems had only a few users and ran on a limited number of mostly disconnected computers. Today the situation is quite different. Nowadays the number of software systems, computers, and users has dramatically increased. Moreover, most computers are connected through the Internet. This workshop will bring together researchers and practitioners interested in exploring how the characteristics of today's computing environment (e.g., high connectivity, substantial computing power for the average user, higher demand for and expectation of frequent software updates) can be leveraged to improve software quality and performance. In particular, the workshop will discuss how software engineers can shift substantial portions of their analysis and measurement activities to actual user environments, so as to leverage in-the-field computational power, human resources, and user data to investigate the behavior of systems after deployment and to improve quality and performance of the systems.

W13: International Workshop on Global Software Development — Friday, May 9

Daniela Damian, University of Victoria, BC Canada
Filippo Lanubile, University of Bari, Italy
Heather L. Oppenheimer, Lucent Technologies, NJ USA

Increased globalization of software development creates software engineering challenges due to the impact of temporal, geographical and cultural differences, and requires development of techniques and technologies to address these issues. The goal of this workshop is to provide an opportunity for researchers and industry practitioners to explore both the state-of-the art and the state-of-the-practice in global software development. The workshop is a continuation of the last five ICSE workshops on the same topic. Our workshop at ICSE last year, after changing the title from the technology-focused "Software Engineering over the Internet" to the more general "Global Software Development," generated increased interest from software practitioners and fruitful discussions between industry and academia. This year at ICSE`03, the workshop will continue and draw upon this interaction between academia and industry in addressing the issues of Global Software Development.

W14: WODA 2003 - Workshop on Dynamic Analysis — Friday, May 9

Jonathan E. Cook, New Mexico State University
Michael D. Ernst, Massachusetts Institute of Technology

This workshop will focus on sharing ideas on and brainstorming new approaches to effective dynamic analysis. It will cover a topical spectrum, possibly including enabling technologies; framework and common tool support; event type definition, classification, and specification; symbolic and theoretically exact reasoning techniques; statistical and probabilistic reasoning techniques; research foundations; relationships to static analysis; relationships to testing; and other potential topics.

W15: STRAW '03 - 2nd International Workshop on SofTware Requirements to Architectures — Friday, May 9

Daniel M. Berry, University of Waterloo, Canada
Roel Wieringa, University of Twente, Netherlands
Rick Kazman, Software Engineering Institute, CMU, USA

There is a clear relationship between requirements engineering and architecture design in software engineering. However, for the most part, the two disciplines have evolved independently from each other, and promising areas of mutual interest remain to be explored. The Second International Workshop on Software Requirements and Architectures (STRAW '03) will bring together researchers from the requirements engineering and architecture communities to exchange views and results that are of mutual interest, and to discuss topics for further research.

Tutorials

F1: Practical Software Measurement **Sunday, May 4**

David Card, Software Productivity Consortium, USA

F2: Lessons Learned from Managing E-business project **Sunday, May 4**

John Backett, Boston University

F3: Documenting Software Architectures: Views and Beyond **Sunday, May 4**

Paul Clements, Software Engineering Institute, USA
David Garlan, Carnegie Melon University, USA
Reed Little, Senior Member of Technical Staff, Software Engineering Institute, USA
Robert Nord, Siemens Corporate Research, USA
Judith Stafford, Tufts University, USA

F4: Computing System Dependability **Sunday, May 4**

John Knight, University of Virginia, USA

F5: Goal-Oriented Requirements Engineering: From System Objectives to UML Models to Precise Software Specifications **Sunday, May 4**

Axel van Lamsweerde, University of Louvain, Belgium

F6: Mastering Design Patterns **Sunday, May 4**

Craig Larman, Valtech, USA

F7: Value-Based Software Engineering (VBE) **Monday, May 5**

Barry Boehm, University of Southern California, USA
Kevin Sullivan, University of Virginia, USA

F8: Usage-Centered Software Engineering: An Agile Approach to Integrating Users, Usability, and User Interfaces into Software Engineering Practice **Monday, May 5**

Larry Constantine, Constantine and Lockwood, Ltd., USA
Lucy Lockwood, Constantine and Lockwood, Ltd., USA

F9: Internet Security and Intrusion Detection **Monday, May 5**

Richard Kemmerer, University of California at Santa Barbara, USA
Giovanni Vigna, University of California at Santa Barbara, USA

F10: Pattern-Oriented Distributed System Architectures **Monday, May 5**

Doug Schmidt, Vanderbilt University, USA

F11: Industrial-Strength Software Product-Line Engineering **Monday, May 5**

John Klein, Avaya CRM, USA
Barry Price, Avaya Labs, USA
David Weiss, Avaya Labs, USA

H1: Feature-Oriented Programming **morning, Monday, May 5**

Don Batory, University of Texas, USA

H2: An Overview of UML 2.0 **morning, Monday, May 5**

Bran Selic, Rational Software Canada

H3: Best Practices for Implementing CMM-Based Software Process Improvement

Bill Curtis, TeraQuest, USA **morning, Monday, May 5**

H4: Designing Software Architectures for Usability **afternoon, Monday, May 5**

Jan Bosch, University of Groningen, The Netherlands
Natalia Juristo, Technical University of Madrid, Spain

H5: Quality of Service Engineering with UML, CORBA, and .NET **afternoon, Monday, May 5**

Kurt Geihs, Berlin University of Technology, Germany
Torben Weis, Berlin University of Technology, Germany

H6: The Agile Unified Process Distilled **afternoon, Monday, May 5**

Craig Larman, Valtech, USA

ICSE 2003 Posters and Demos

Poster presenters will be available to discuss their posters during coffee breaks as well as at the conference reception.

Demonstrations of the latest research prototypes will be given at selected times during the conference. Attendees can hear formal presentations of the prototypes or visit with the presenters at scheduled demonstration times..

Saturday Program At-A-Glance — May 3, 2003

Workshops

all day	**W1**: SELMAS'03 2nd Int'l. Workshop on Software Engineering for Large-Scale Multi-Agent Systems (Day 1 of 2)	**W2:** EDSER-5 5th Int'l. Workshop on Economics-Driven Software Engineering Research (Day 1 of 2)	**W3:** CBSE6 6th Workshop on Component-Based Software Engineering (Day 1 of 2)	**W4:** Bridging the Gaps Between Software Engineering and Human-Computer Interaction (Day 1 of 2)	**W5:** First Workshop on Require-ments Engineering in Open Systems (Day 1 of 2)	**W6:** Workshop on Software Variability Management	**W7:** Workshop on Software Architectures for Dependable Systems	**W8:** 2nd Int'l. Workshop on Scenarios and State Machines	**W9:** 3rd Workshop on Open Source Software Engineering

Other Events

all day	**C1**: ProSim (Day 2 of 3, off-site). First day was Friday, May 2.

Sunday Program At-A-Glance — May 4, 2003

Workshops and Other Events

morning	**W1**: SELMAS'03 (Day 2 of 2)	**W2:** EDSER-5 (Day 2 of 2)	**W3:** CBSE (Day 2 of 2)	**W4:** SE and HCI (Day 2 of 2)	**W5:** Requirements Engineering in Open Systems (Day 2 of 2)	**C1**: ProSim (off-site, Day 3 of 3)
afternoon		Pioneers Symposium				
evening		Pioneers Symp. Reception				

Tutorials

all day	**F1**: Practical Software Measurement. *D. Card*	**F2**: Lessons Learned from Managing E-business Projects. *J. Brackett*	**F3**: Documenting Software Architectures. *P. Clements, D. Garlan, R. Little, R. Nord, J. Stafford*	**F4**: Computing Systems Dependability. *J. Knight*	**F5**: Software Requirements Engineering. *A. van Lamsveerde*	**F6**: Mastering Design Patterns. *C. Larman*

Legend: W=Workshop, F=Full-day tutorial, H=Half-day tutorial, C=Co-located event, T=Technical track, E=Education track, X=Experience report track.

Tutorials

morning	**F7**: Value-Based Software Engineering. *B. Boehm, K. Sullivan*	**F8**: Usage-Centered Software Engineering. *L. Constantine, L. Lockwood*	**F9**: Internet Security. *R. Kemmerer, G. Vigna*	**F10**: Pattern-Oriented Distributed System Architectures. *D. Schmidt*	**F11**: Industrial-Strength Software Product-Line Engineering. *J. Klein, B. Price, D. Weiss*	**H1**: Feature-Oriented Programming. *D. Batory*	**H2**: An Overview of UML 2.0. *B. Selic*	**H3**: CMM Best Practices. *B. Curtis*
afternoon						**H4**: Designing Software Architectures for Usability. *J. Bosch*	**H5**: Quality of Service with UML, CORBA, and .NET. *K. Geihs, T. Weis*	**H6**: Agile, Unified Process. *C. Larman*

Other Events

all day	Doctoral Symposium	NSEFS'03 - New Faculty Symposium	**C2**: Summit on Software Engineering Education
evening			**ICSE** Steering Committee Meeting

Legend: W=Workshop, F=Full-day tutorial, H=Half-day tutorial, C=Co-located event, T=Technical track, E=Education track, X=Experience report track.

Tuesday Program At-A-Glance — May 6, 2003

Session 1: Welcome and Keynote
Components That You Can Trust, *Bertrand Meyer, ETH, Zürich, and ISE Santa Barbara, USA*

Coffee Break

Session 2.1T: Component Technologies
- Pluggable Reflection: Decoupling Meta-Interface and Implementation. *D. Lorenz, J. Vlissides*
- Component Rank: Relative Significance Rank for Software Component Search. *K. Inoue, R. Yokomori, H. Fujiwara, T. Yamamoto, M. Matsushita, S. Kusumoto*
- Vienna Component Framework: Enabling Composition Across Component Models. *J. Oberleitner, T. Gschwind, M. Jazayeri*

Session 2.2T: Testing I
- Constructing Test Suites for Interaction Testing. *M. Cohen, P. Gibbons, W. Mugridge, C. Colbourn*
- Improving Web Application Testing with User Session Data. *S. Elbaum, S. Karre, G. Rothermel*
- Improving Test Suites via Behavioral Abstraction. *M. Harder, J. Mellen, M. Ernst*

Session 2.3E: Extreme Programming
- Introducing Software Engineering by means of Extreme Programming. *L. Bendix, G. Hedin, B. Magnusson*
- Agile Processes in Software Engineering -- An Educational Perspective. *J. Schneider, L. Johnstoni*
- Panel: eXtreme Programming: Helpful or Harmful?
  Chair: *H. Saiedian*

Session 2.4X: Case Studies
- Fault-tolerance in a Distributed Management System: a Case Study. *R. Smeikal, K. Goeschka*
- The Deployer's Problem: Configuring Application Servers for Performance and Reliability. *M. Raghavachari, D. Reimer, R. Johnson*
- From Scenarios to Code: An Air Traffic Control Case Study. *J. Whittle, R. Kwan, J. Saboo*

Lunch

Session 3.1T: Empirical Studies I
- Using Benchmarking to Advance Research: A Challenge to Software Engineering. *S. Sim, S. Easterbrook, R. Holt*
- An Empirical Study of an Informal Knowledge Repository in a Medium-Sized Software Consulting Company. *T. Dingsoyr, E. Royrvik*
- End-User Software Engineering with Assertions in the Spreadsheet Paradigm. *M. Burnett, C. Cook, O. Pendse, G. Rothermel, J. Summet, C. Wallace*

Session 3.2: FoSP
Component Technology - What, Where, and How? *C. Szyperski*

Session 3.3E: Undergrad. Education
- The Impact of Pair Programming on Student Performance and Pursuit of Computer Science Related Majors. *H. Bullock, J. Fernald, C. McDowell, L. Werner*
- Positive Experiences with an Open Project Assignment in an Introductory Programming Course. *G. Sindre, S. Line, O. Valvag*
- Problems and Programmers: An Educational Software Engineering Card Game. *A. Baker, E. Oh, A. van der Hoek*

Session 3.4X: Automotive Software Engineering
- Software Technology in an Automotive Company - Major Challenges. *K. Grimm*
- Panel: Automotive Software Engineering.
  Chair*: M. Broy*

Coffee Break

Session 4.1T: Design Recovery and Documentation
- On the Uniformity of Software Evolution Patterns. *E. Barry, C. Kemerer, S. Slaughter*
- Design Recovery of Interactive Graphical Applications. *K. Chan, Z. Lian, A. Michail*
- Recovering Documentation to Source Code Traceability Links using Latent Semantic Indexing. *A. Marcus, J. Maletic*

Session 4.2T: Formal Methods I
- Computer-Assisted Assume/Guarantee Reasoning With VeriSoft. *J. Dingel*
- Architecture, Design, Implementation. *A. Eden, R. Kazman*
- Cadena: An Integrated Development, Analysis, and Verification Environment for Component-Based Systems. *J. Hatcliff, W. Deng, M. Dwyer, G. Jung, V. Ranganath*

Session 4.3: Panel
Empirical Validation: What, Why, When, Where.
Chair: *R. Walker*

Session 4.4X: Process Analysis and Improvement
- Experiences on Defining and Evaluating an Adapted Review Process. *T. Barth, T. Gantner*
- Towards Systematic Recycling of Systems Requirements. *N. Heumesser, F. Houdek*
- Tricks and Traps of Initiating a Product Line Concept in Existing Products. *C. Ebert, M. Smouts*

SIGSOFT/TCSE General Meetings

Birds of a Feather: IEEE-CS/ACM Computing Curricula for Software Engineering. *R. LeBlanc*

Agile (Cooperative learning experience)

<table>
<tr><td colspan="4">Session 5: Keynote
Must There Be So Few? Including Women in CS, Joanne McGrath Cohoon, Department of Leadership, Foundations and Policy, University of Virginia, USA</td></tr>
<tr><td colspan="4">Coffee Break</td></tr>
<tr><td>Session 6.1T: Software Design
- DADO: Enhancing Middleware to Support Cross-Cutting Features in Distributed, Heterogeneous Systems. E. Wohlstadter, S. Jackson, P. Devanbu
- Scaling Step-Wise Refinement. D. Batory, J. Sarvela, A. Rauschmayer
- Sound Methods and Effective Tools for Engineering Modeling and Analysis. D. Coppit, K. Sullivan</td><td>Session 6.2T: Testing II
- Fragment Class Analysis for Testing of Polymorphism in Java Software. A. Rountev, A. Milanova, B. Ryder
- A Framework for Component Deployment Testing. A. Bertolino, A. Polini
- Data Flow Testing as Model Checking. H. Hong, S. Cha, I. Lee, O. Sokolsky, H. Ural</td><td>Session 6.3: Panel
Undergraduate SE Degree: Pros and Cons.
Chair: D. Perry</td><td>Session 6.4: Mini-tutorial
Writing Good Software Engineering Research Papers. M. Shaw</td></tr>
<tr><td colspan="4">Lunch</td></tr>
<tr><td>Session 7.1T: Software Process
- New Directions on Agile Methods: A Comparative Analysis. P. Abrahamsson, J. Warsta, M. Siponen, J. Ronkainen
- Agile Authoring of Software Documentation Using RaPiD7. R. Kylmakoski
- Using Process Technology to Control and Coordinate Software Adaptation. G. Valetto, G. Kaiser</td><td>Session 7.2: FoSP
Patterns, Frameworks, and Middleware: Their Synergistic Relationship, D. Schmidt</td><td>Session 7.3T: Empirical Studies II
- Understanding and Predicting Effort in Software Projects. A. Mockus, D. Weiss, P. Zhang
- Cost Estimation for Web Applications. M. Ruhe, R. Jeffery, I. Wieczorek
- Evaluating the Quality of Information Models: Empirical Testing of a Conceptual Model Quality Framework. D. Moody, G. Sindre, T. Brasethvik, A. Solvberg</td><td>Session 7.4: Demos</td></tr>
<tr><td colspan="4">Coffee Break</td></tr>
<tr><td colspan="4">Session 8: Awards</td></tr>
<tr><td colspan="4">Reception: Relax, enjoy hors d’oeuvres, and discuss posters with their creators</td></tr>
</table>

Legend: W=Workshop, F=Full-day tutorial, H=Half-day tutorial, C=Co-located event, T=Technical track, E=Education track, X=Experience report track.

Thursday Program At-A-Glance — May 8, 2003

Session 9: Keynote
Relating Software Engineering and Information Security, *Eugene Spafford, Purdue University, USA*

Coffee Break

Session 10.1T: Program Analysis
- Whole Program Path-Based Dynamic Impact Analysis. *J. Law, G. Rothermel*
- Precise Dynamic Slicing Algorithms. *X. Zhang, R. Gupta, Y. Zhang*
- An Empirical Study of Predicate Dependence Levels and Trends. *D. Binkley, M. Harman*

Session 10.2T: Software Architecture
- Comparison of Two Component Frameworks: The FIPA-Compliant Multi-Agent System and the Web-Centric J2EE Platform. *M. Casagni, M. Lyell*
- Design Pattern Rationale Graphs: Linking Design to Source. *E. Baniassad, G. Murphy, C. Schwanninger*
- A Component Architecture for an Extensible, Highly Integrated Context-Aware Computing Infrastructure. *W. Griswold, R. Boyer, S. Brown, T. Truong*

Session 10.3E: Course Delivery and Evaluation
- Evaluating Individual Contribution Toward Group Software Engineering Projects. *J. Hayes, T. Lethbridge, D. Port*
- On The Supervision and Assessment of Part-Time Postgraduate Software Engineering Projects. *A. Simpson, A. Martin, J. Gibbons, J. Davies, S. McKeever*
- Using a Web-Based Project Process Throughout the Software Engineering Curriculum. *D. Bagert, S. Mengel*

Session 10.4X: Process and Tools
- About the Development of a Point of Sale System: an Experience Report. *S. Berner*
- Effective Experience Repositories for Software Engineering. *K. Schneider and J. von Hunnius*
- The Software Engineering Impacts of Cultural Factors on Multi-cultural Software Development Teams. *G. Borchers*

Lunch

Session 11.1T: Formal Methods II
- A Compositional Formalization of Connector Wrappers. *B. Spitznagel, D. Garlan*
- Modular Verification of Software Components in C. *S. Chaki, E. Clarke, A. Groce, S. Jha, H. Veith*
- Architectural Interaction Diagrams: AIDs for System Modeling. *A. Ray, R. Cleaveland*

Session 11.2T: Software Understanding
- Hipikat: Recommending Pertinent Software Artifacts. *D. Cubranic, G. Murphy*
- Toward an Understanding of the Motivation of Open Source Software Developers. *Y. Ye, K. Kishida*
- Tools for Understanding the Behavior of Telecommunication Systems. *A. Marburger, B. Westfechtel*

Session 11.3E: Process and Methodology
- Beyond the Personal Software Process: Metrics Collection and Analysis for the Differently Disciplined. *P. Johnson, H. Kou, J. Agustin, C. Chan, C. Moore, J. Miglani*
- Teaching Contract Programming Concepts to Future Software Engineers. *D. Ross*
- Empowering Software Engineers in Human-Centered Design. *A. Seffah*

Session 11.4X: Architecture
- Why Can't They Create Architecture Models Like "Developer X"? An Experience Report. *G. Fairbanks*
- The Co-Evolution of a Hype and a Software Architecture: Experience of Component-Producing Large-Scale EJB Early Adopters. *L. Prechelt and Daniel Hutzel*
- Quantifying the Value of Architecture Design Decisions: Lessons from the Field. *M. Moore, R. Kazman, Mark Klein, J. Asundi*

Coffee Break

Session 12.1T: Consistency Management and Quality Assurance
- Palantir: Raising Awareness among Configuration Management Workspaces. *A. Sarma, Z. Noroozi, A. van der Hoek*
- Consistency Management with Repair Actions. *C. Nentwich, W. Emmerich, A. Finkelstein*
- Automated Support for Classifying Software Failure Reports. *A. Podgurski, D. Leon, P. Francis, W. Masri, M. Minch, J. Sun*

Session 12.2: FoSP
CyberSecurity: What are Best Practices?
R. Kemmerer

Session 12.3: Panel
Modules for the New Millenium.
Chair: *P. Devanbu*

Session 12.4X: Testing and Fault Correction
- Assessing Test-Driven Development at IBM. *E. Maximilien, L. Williams*
- An Analysis of the Fault Correction Process in a Large-Scale SDL Production Model. *W. Zage, D. Zage*
- Requirements Discovery During the Testing of Safety-Critical Software. *R. Lutz, C. Mikulski*

Closing Ceremonies

Friday Program At-A-Glance — May 9, 2003

Workshops

all day	**W10**: Software Engineering for High Assurance Systems (Day 1 of 2)	**W11**: ACSE 2003 3rd Int'l. Workshop on Adoption-Centric Software Engineering	**W12**: RAMSS Workshop on Remote Analysis and Measurement of Software Systems	**W13**: Int'l. Workshop on Global Software Development	**W14**: WODA 2003 Workshop on Dynamic Analysis	**W15**: STRAW'03 2nd Int'l. Workshop on SofTware Requirements to Architectures

Other Events

all day	**C3**: SPIN (Day 1 of 2)	**C4:** SCM (Day 1 of 2)

Saturday Program At-A-Glance — May 10, 2003

Workshops

all day	**W10**: Software Engineering for High Assurance Systems (Day 2 of 2)

Other Events

all day	**C3**: SPIN (Day 2 of 2)	**C4:** SCM (Day 2 of 2)	**C5**: IWPC (Day 1 of 2). Continues on Sunday, May 11.

Legend: W=Workshop, F=Full-day tutorial, H=Half-day tutorial, C=Co-located event, T=Technical track, E=Education track, X=Experience report track.

Venue and Registration Information

Conference Hotel

The conference venue is the Hilton Portland & Executive Tower, centered in the midst of Portland's vibrant downtown area. Located in the heart of the entertainment and cultural district, the Hilton provides access to performing arts, shopping, museums, coffee houses, microbreweries, and numerous restaurants, all within three blocks.

City of Portland

The "Rose City's" culture simmers in coffeehouses, Native American art galleries, ubiquitous bookstores, musical and theatrical events, and lively brewpubs. Classical and modern art of all kinds is currently displayed at the Portland Art Museum. Portland is a wonderful city for walkers. The city's streets, which feature statues, fountains and half-size city blocks, prompted Portland's selection in 1998 as one of America's best walking towns by Walking Magazine. For visitors covering a larger portion of the city, public transportation is both accessible and convenient. Light rail trains (MAX), the Central City Streetcar and an easy-to-navigate bus system all offer free service within downtown Portland. Crowning the city's skyline is Mount Hood, the tallest peak in Oregon's Cascade Mountain Range. Within the metro area are 37,000 acres of parks and green spaces that include fabulous rose gardens and the brand new Classical Chinese Garden. For hikers and mountain bikers, the city is proud to host the nation's largest urban wilderness, the nearly 5,000-acre Forest Park.

Adjacent to Portland itself are the outstanding recreational opportunities of the Pacific Northwest. Windswept beaches, verdant forests, and snow-capped peaks give way to sweeping rangelands, towering rock formations, and dramatic river valleys in this diverse land. ICSE 2003 is planning to make a number of local excursions and tours available, such as visiting Mt. Saint Helens, whale watching on the coast, a winery tour in the Oregon countryside, and a Native American cultural tour on an Indian reservation.

Reception

The reception is a time for attendees to mingle and relax together. Extra tickets will be available for purchase.

WOW! The ICSE Newsletter

Continuing a longstanding ICSE conferencing tradition, our onsite newsletter ("Window On the World" or WOW) will be published every morning of the conference. WOW will include a medley of previews of the day's events, reviews of yesterday's happenings, interviews with key people and award winners, pointers to local attractions and entertainment, interesting technical and non-technical news from beyond the borders of our sheltered hotel, including weather, humor, poems, contests, and maybe even some editorials.

If you're intrigued to perhaps join the WOW staff (2-3 hours nightly, for 2 or 3 or 4 days), options include production and copy editing in addition to producing content, send mail to Hal Hart, Hal.Hart@ACM.ORG, to discuss openings and "benefits." Gain a sense of adding value to ICSE attendance by helping make WOW a remembrance taken home by conferees, shared with others, perhaps even a conference highlight.

Registration Information

On-line registration is available on the conference web page: http://www.icse-conferences.org/2003/
The deadline for the early registration discount is April 12.

Student Volunteers

The student volunteer program is an opportunity for students from around the world to associate with the top researchers and practitioners in software engineering. In return for about 20 hours of their time, student volunteers receive a complimentary registration that includes all conference meals. In the past, volunteers helped with registration and assisted the organizers with running the conference smoothly. Student volunteers must be enrolled in a Ph.D., Masters, or full-time undergraduate program at the time of the conference. The application form can be found on the conference web page. The deadline is the same as the early registration deadline, April 12.

ICSE 2003 Program At-A-Glance

<table>
<tr><th>Sat.
May 3</th><th>Sun.
May 4</th><th>Mon.
May 5</th><th>Tue.
May 6</th><th>Wed.
May 7</th><th>Thu.
May 8</th><th>Fri.
May 9</th><th>Sat.
May 10</th></tr>
<tr><td colspan="2">Workshops W1-W5</td><td>Doctoral Symp.</td><td colspan="3" rowspan="2">Technical Program</td><td colspan="2">Workshop W10</td></tr>
<tr><td>Workshops W6-W9</td><td>Pioneers Symp.</td><td>New Faculty Symp.</td><td colspan="2">Workshops W11-W15</td><td>IWPC (cont. on Sun.)</td></tr>
<tr><td colspan="2">ProSim (cont. from Fri.)</td><td>Summit on SE Educ.</td><td>SigSoft/TCSE meetings</td><td>Awards</td><td>Closing Ceremonies</td><td colspan="2">SPIN</td></tr>
<tr><td></td><td>Tutorials F1-F6</td><td>Tutorials F7-F11, H1-H6</td><td>Curricula for SE</td><td>Reception</td><td rowspan="2"></td><td colspan="2">SCM</td></tr>
<tr><td colspan="3"></td><td>Agile Learning</td><td>Posters</td><td colspan="3"></td></tr>
</table>

Sponsors and Donors

Sponsoring Organizations

IEEE

IEEE Computer Society Technical Council on Software Engineering

ACM

ACM SIGSOFT

Corporate Donors

Platinum Level Donor:

Northrop Grumman Space Technology

Gold Level Donors:

Sun microsystems

Nokia Connecting People

BMW

Bronze Level Donor:

DaimlerChrysler