Welcome to ICSE 2003, Software Engineering Week in Portland!

A MESSAGE FROM THE CHAIRS

Welcome to the silver anniversary of the International Conference on Software Engineering! Managing the complexity of modern software systems is without question a grand challenge for software engineering. It is therefore fitting that, inspired by the view of Mount Hood, the theme for this year’s meeting is:Scaling New Heights. ICSE 2003 brings together world leaders in software engineering research, practice, and education to present and discuss the most recent advances, trends, and concerns in this ever expanding and critical field.

Software Engineering is a vibrant and growing discipline. This year there was a significant increase in submissions in all areas, with 324 technical papers, 52 education papers, and 61 experience reports submitted. All submissions were rigorously reviewed by multiple experts on the ICSE 2003 committees (a minimum of three reviewers for technical papers), leading to the acceptance of 42 technical papers, 16 experience reports, and 11 education papers. The result is an impressive program.

The ICSE Software Engineering Week, May 3 to 10, 2003, consists of the main ICSE conference and an interesting array of tutorials, workshops, and related events associated or co-located with ICSE. Each morning of the ICSE 2003 conference starts with a topical keynote address by an outstanding speaker. Other notable highlights of the 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 research demonstrations, as well as a morning newsletter describing memorable moments from the previous day, 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 13 tutorials on a variety of topics and 14 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 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.

ICSE 2003 is being held in the Portland Hilton Hotel, which resides 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. We hope you seize the opportunity, perhaps before or after this busy Software Engineering week, to explore the “Rose City” and the many recreational opportunities of the Pacific Northwest.

A conference as diverse as ICSE requires the efforts of a number of volunteers. Thank you, members of the ICSE 2003 program committee, organizing committee, and all sub-committees! We also wish to acknowledge the support of our donors and sponsors. Finally, thank you for joining us at this the silver anniversary of ICSE. We anticipate that a memorable week of presentations, discussions, and demonstrations lies ahead. Enjoy!

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

Lori A. Clarke
General Chair
Laura K. Dillon
Program Co-Chair
Walter Tichy
Program Co-Chair
### Table of Contents

- Conference Committees .................................................. inside front cover
- Welcome Message ............................................................... 1
- Keynote Speakers ............................................................... 3
- Frontiers of Software Practice (FoSP) ................................... 4

#### Program

<table>
<thead>
<tr>
<th>Date</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSE Overall Program</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Community Meetings at ICSE</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Saturday May 3</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Sunday May 4</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Monday May 5</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Tuesday May 6</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Wednesday May 7</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Thursday May 8</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Friday May 9</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Saturday May 10</td>
<td>........................................................................</td>
</tr>
<tr>
<td>Sunday May 11</td>
<td>........................................................................</td>
</tr>
</tbody>
</table>

#### ICSE Events, Workshops, Tutorials, Panels, Demos and Posters

- Doctoral Symposium ....................................................... 17
- New Software Engineering Faculty Symposium (NSEFS 03) .......... 17
- Pioneers Symposium .......................................................... 17
- Agile Development: A Cooperative Learning Experience .......... 18
- Workshops ......................................................................... 18
- Tutorials ........................................................................... 23
- Panels ............................................................................... 28
- Demonstrations and Posters ............................................... 29

#### Venue and Additional Information

- City of Portland, Reception, WOW! Newsletter, Student Volunteers .......................................................... 32
- ICSE 2004 and ICSE 2005 Announcements ............................... inside back cover
- Hotel Maps ........................................................................ back cover
- Sponsors and Donors ............................................................. back cover

#### Other Events Co-located with ICSE 2003

- **ProSim’03:** Workshop: Software Process Simulation Modeling (held off-site at Portland State University), May 3-4
- **SSEE II:** 2nd International Summit on Software Engineering Education, May 5
- **SPIN 2003:** The 10th International SPIN Workshop on Model Checking of Software, May 9-10
- **SCM-11:** 11th International Workshop on Software Configuration Management, May 9-10
- **IWPC 2003:** 11th International Workshop on Program Comprehension, May 10-11
The Grand Challenge of Trusted Components

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

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

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.

continued next page
Keynote Speakers

continued from page 3

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.

Component Technology - What, Where, and How? Tuesday, May 6
Clemens Szyperski, Microsoft Corporation, USA

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 Ph.D. 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 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 Wednesday, May 7
Douglas Schmidt, Vanderbilt University, DARPA, USA

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. Schmidt is a 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 frameworks that implement patterns
for high-performance, real-time systems. Dr. Schmidt also
serves as a program manager in the DARPA Information
Exploitation Office (IXO), where he leads the effort on
distributed object computing middleware research.

CyberSecurity: What are Best Practices?
Richard Kemmerer, University of California at Santa Barbara, USA

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 and of the ACM, and past Editor-in-Chief
of IEEE Transactions on Software Engineering. Dr. Kemmerer
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.
ICSE 2003 Overall Program At-A-Glance

<table>
<thead>
<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>
</thead>
<tbody>
<tr>
<td>Workshops W1-W4</td>
<td>Doctoral Symp.</td>
<td>Technical Program</td>
<td>Workshop W10</td>
<td>Workshops W6-W9</td>
<td>Pioneers Symp.</td>
<td>New Faculty Symp.</td>
<td>Workshops W11-W15</td>
</tr>
<tr>
<td>Workshops W6-W9</td>
<td>Tutorials F3-F6</td>
<td>Tutorials F7-F8, F10-F11, H1-H6</td>
<td>Computing Curricula for SE (CCSE)</td>
<td>Demos</td>
<td>Awards</td>
<td>Closing Ceremonies</td>
<td>SPIN</td>
</tr>
</tbody>
</table>

Community Meetings at ICSE

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Meeting</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday May 4</td>
<td>8am - 6pm</td>
<td>ESEC/FSE PC Meeting</td>
<td>Grand Ballroom II</td>
</tr>
<tr>
<td>Monday May 5</td>
<td>8am - 6pm</td>
<td>ESEC/FSE PC Meeting</td>
<td>Grand Ballroom II</td>
</tr>
<tr>
<td>Monday May 5</td>
<td>6 - 7pm</td>
<td>ICSE ’04 PC Meeting</td>
<td>Parlor A&amp;B</td>
</tr>
<tr>
<td>Monday May 5</td>
<td>7:30 - 10:30pm</td>
<td>ICSE steering committee meeting</td>
<td>Alexander’s</td>
</tr>
<tr>
<td>Tuesday May 6</td>
<td>12 -1:30pm</td>
<td>SIGSOFT Executive Committee Meeting</td>
<td>Senate</td>
</tr>
<tr>
<td>Tuesday May 6</td>
<td>5 - 6pm</td>
<td>SIGSOFT General Meeting</td>
<td>Broadway II-IV</td>
</tr>
<tr>
<td>Tuesday May 6</td>
<td>6 - 7pm</td>
<td>TCSE General Meeting</td>
<td>Broadway II-IV</td>
</tr>
<tr>
<td>Tuesday May 6</td>
<td>6 - 10pm</td>
<td>IFSA</td>
<td>Forum</td>
</tr>
<tr>
<td>Wednesday May 7</td>
<td>12 - 1:30pm</td>
<td>ICM Steering Committee Meeting</td>
<td>Senate</td>
</tr>
<tr>
<td>Wednesday May 7</td>
<td>12 - 1:30pm</td>
<td>ESEC/FSE Steering Committee Meeting</td>
<td>Studio</td>
</tr>
<tr>
<td>Wednesday May 7</td>
<td>6 - 7pm</td>
<td>ISSTA Steering Committee Meeting</td>
<td>Studio</td>
</tr>
<tr>
<td>Wednesday May 7</td>
<td>7 - 9pm</td>
<td>TSE board meeting and dinner</td>
<td>Alexander’s Crown Room</td>
</tr>
<tr>
<td>Friday May 9</td>
<td>8:30-11am</td>
<td>ICSE postmortem</td>
<td>Alexander’s</td>
</tr>
</tbody>
</table>