

ICSE 97

International Conference
on Software Engineering



Pulling Together

ADVANCE PROGRAM



*Sponsored by the ACM Special
Interest Group on Software
Engineering (SIGSOFT) and
IEEE Computer Society —
Technical Council on Software
Engineering (TCSE)*



Boston, Massachusetts USA

May 18–23, 1997

Early registration deadline April 18, 1997

PULLING TOGETHER

ICSE 97 • BOSTON, MASSACHUSETTS, USA • MAY 17-23, 1997



ELECTRONIC VERSIONS

The Advance Program and Call for Participation are available electronically via

- anonymous ftp to www.ics.uci.edu; change directory to pub/ics97/
- World Wide Web:
<http://www.ics.uci.edu/ics97/>

ICSE 97 CONFERENCE OFFICE

For general conference information contact:

ICSE 97

Department of Computer Science
307 Lederle Graduate Research Center
PO Box 34610
University of Massachusetts, Amherst
Amherst, MA 01003-4610 USA

+1-413-545-2475
+1-413-545-3729 (fax)
ics97@cs.umass.edu

THE CREATION, DEPLOYMENT, EVOLUTION, AND MEANING of software and its role in modern society is changing and expanding as a result of new technologies, new applications, and new social factors. The Internet, the World Wide Web, multimedia interfaces, and neighborhood software stores have added new dimensions to traditional issues and topics in software engineering. The International Conference on Software Engineering is changing with the discipline to encompass the new emphases and the broadened sweep of topics and concerns which confront today's software professionals and researchers.

The theme of the 1997 International Conference on Software Engineering is "Pulling Together." Pulling together denotes coordinated action of many individuals in achieving a common goal. It also describes the coming together of many different perspectives, concerns, and abilities to find a common ground and a way of achieving cooperation. Pulling together is fundamentally dynamic in nature, and is often a matter of explicit negotiation and communication. Major changes have been instituted in ICSE 97 to help the software engineering community pull together, in the full sense of that phrase.

ICSE 97 includes a widened range of conference activities, a widened range of participants, and new technical areas. A broadened outlook challenges old beliefs, promotes new ideas and new synergies, and provides for a dynamic, exciting program. New or expanded conference activities include a doctoral symposium, lessons and reports from software engineering organizations, a mentor program, posters, and a commercial exhibit.

A major addition to the conference is a suite of sessions and activities focusing on the interests and needs of the practicing professional. Numerous invited presentations, timely panel topics, experience reports, and an expanded tutorial program are included.

We hope that you will pull together with us, bringing new ideas, new concerns, and new goals, thus helping us to find common ground and reach new objectives.

W. Richards Adrion
ICSE 97 General Chair

Alfonso Fuggetta, Richard N. Taylor, Anthony I. Wasserman
ICSE 97 Program Coordinators

KEYNOTE PRESENTATIONS

ICSE 97 • BOSTON, MASSACHUSETTS, USA • MAY 18-23, 1997

Beyond Software Engineering: Ten Imperatives for the Successful Software Developer at the End of the Decade

by Ed Yourdon

Edward Yourdon, methodologist, author, consultant, and publisher of American Programmer, developed the "Yourdon method" of structured systems analysis and design implemented on most of today's CASE software engineering tools, and is a world expert

in software engineering. Author of 20 textbooks and more than 200 articles, he chairs international conferences on CASE technology, is a Professor at Universidad CAECE in Buenos Aires, and has received numerous honors and awards from universities and societies worldwide.



ED YOURDON

Java and the Evolution of Web Software

by Guy Steele

Guy L. Steele Jr. is a Distinguished Engineer at Sun Microsystems Laboratories in Chelmsford, Massachusetts, and is responsible for research in programming languages, parallel algorithms, implementation strategies, and architectural and software support. He is working with James

Gosling and Bill Joy on the detailed specification of the Java programming language. An ACM Fellow and a Fellow of the AAAI, he was a senior scientist at Thinking Machines Corporation, a member of technical staff at Tartan Laboratories, and an assistant professor at Carnegie-Mellon University.



MARK WEISER

Software Engineering That Matters to People

by Mark Weiser

Dr. Mark Weiser is chief technologist at the Xerox Palo Alto Research Center (parc). Prior to joining parc, Weiser taught at the University of Maryland from 1979 to 1987, where he headed the Computer Science Laboratory. Weiser has started three

companies, and he has written more than 75 technical publications. His Ubiquitous Computing program envisions PCs being replaced with invisible computers embedded in everyday objects. Weiser is the drummer with rock band Severe Tire Damage, the first live band on the Internet.

ICSE SCHEDULE

		8:00	9:00	10:00	11:00	12:00	13:00 (1:00)
SUNDAY	18		Tutorials session 1				
		Tutorials session 2					
MONDAY	19		Tutorials session 4				
		Tutorials session 5					
TUESDAY	20		Opening	1. Keynote	2. Technical Sessions		
WEDNESDAY	21		5. Keynote		6. Technical Sessions		7. T
THURSDAY	22		10. Keynote		11. Technical Sessions		12. T

WORKSHOPS & SYMPOSIA See web pages for details

Process Modelling and Empirical Studies of Software Evolution

<http://louis.ecs.soton.ac.uk/~rh/cfp.html>

Much progress is being made in both the areas of process modelling and software metrics. However, neither of these concepts is complete without the other: processes cannot be improved if no assessment of quality is available, and metrics are useless if they cannot be applied in order to assess the evolution of systems. The goal of this workshop is to stimulate discussion and collaboration between researchers and practitioners in the area of empirical process modelling studies of large-scale systems' development and evolution. We hope to find a number of high-level approaches which will be of interest particularly to those working on both large-scale open systems and on object-oriented systems. **Sunday, May 18**

Living With Inconsistency

<http://www.ics.uci.edu/icse97/workshop/inconsistency.html>

In this workshop, we take a broader view. We are interested in living with inconsistency as part of doing requirements engineering. While some types of inconsistency can be detected and removed early on, other types may not even be noticeable until a system is in production for months or years. **Saturday, May 17**

ISEW Cleanroom

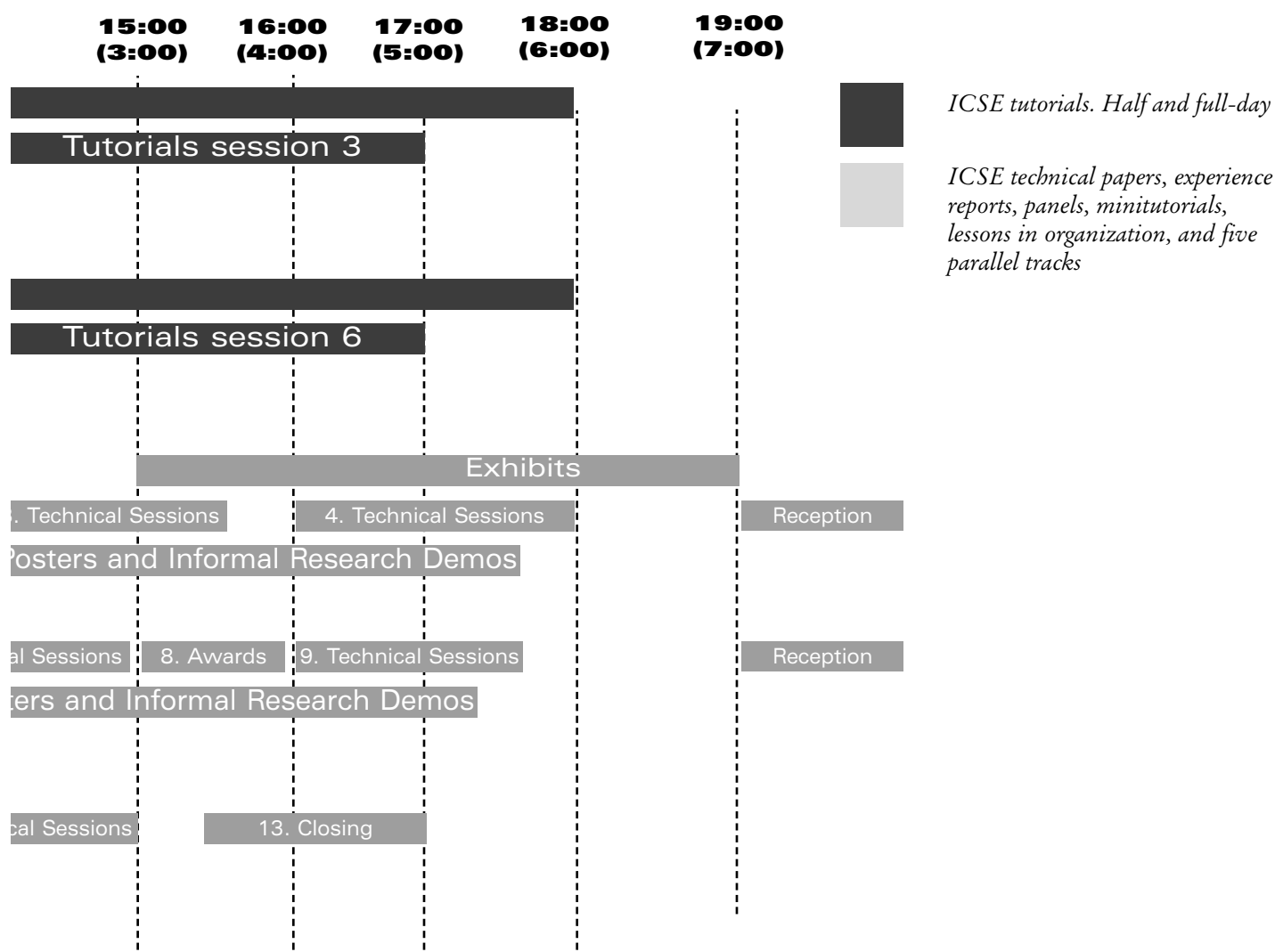
http://www.q-labs.com/isew_icse.html

The workshop will provide an opportunity for people active in Cleanroom, whether as practitioners or researchers to exchange experiences and to discuss problems of common interest. **Saturday, May 17**

Software Engineering for Parallel and Distributed Systems

<http://www.osl.cs.uiuc.edu/PDSE97/welcome.html>

Software applications require the use of explicit parallel programming techniques in order to meet their specifications. Software Engineers must deal with these issues in addition to tackling the more commonly identified problems which occur in all software projects. This workshop is one in a series to provide "one-stop" coverage to the field. **Saturday–Sunday, May 17–18**



Software Engineering on the Web

<http://ricis.cl.uh.edu/SEWWW>

This workshop identifies near and midterm goals of how best to develop and use the world wide web for software engineering. **Monday, May 19**

Software Configuration Management (SCM7)

<http://www.cs.colorado.edu/~andre/scm-7-cfp.html>

SCM is the discipline of managing the evolution of families of software systems. It involves supporting the versioning, composition, and generation of all relevant software items, as well as controlling and supporting related systems. The goal of this workshop is to bring together researchers, vendors and practitioners to discuss the state of the art in SCM as well as challenges for the future. **Sunday–Monday, May 18–19**

Symposium on Software Reusability

<http://www.owego.com/~ssr97/>

The Symposium on Software Reusability is ACM's bi-annual forum for the exchange of ideas, research and development results and experiences in all aspects of software reusability. **Sunday–Monday, May 18–19; tutorials, Saturday, May 17**

Migration Strategies for Legacy Systems

<http://www.infosys.tuwien.ac.at/Research/ICSE97-ws>

Improving the value of legacy systems is of high concern to many organizations nowadays. The modernization of old software helps in gaining control of maintenance costs, in improving system performance, or in moving to a distributed and more efficient environment. **Saturday, May 17**

Software Engineering Education Doctoral Consortium

<http://www.ics.uci.edu/icse97/doctoral.html>

The ICSE 97 Doctoral Consortium is a one-day workshop prior to the regular ICSE technical conference. The goal of the doctoral consortium is to publicly discuss research goals, methods, and results at an early enough stage in Ph.D. research to provide useful guidance in completing the dissertation research and initiation of a research career. The consortium and ICSE will also provide an opportunity for student participants to interact with established researchers and others in the wider software engineering community. **Monday, May 19**

TUTORIALS

COORDINATORS: Mark A. Ardis, AT&T, and Wilhelm Schaefer, Universitaet Paderborn.

See pages 4-5 for tutorial session times

Bill Curtis is Vice President and Chief Scientist with TeraQuest Metrics, Inc., a company that works with client organizations to improve their software development capability. He is a former Director of the Software Process Program in the Software Engineering Institute at Carnegie Mellon University. He works with the SEI, is a founding faculty member of the Software Quality Institute at the University of Texas, and he works with organizations to increase their software development capability.

Timothy D. Korson is currently a senior partner in the training and consulting firm Software Architects, executive director of Comsoft, and director of Southern Adventist University's Software Technology Center. He is a former visiting scientist at the SEI who taught software engineering at Clemson University for 7 years.

John D. Musa, an independent consultant, was recently Technical Manager of Software Reliability Engineering (SRE) at AT&T Bell Laboratories, Murray Hill, NJ. He is a creator of the field of software reliability engineering and is widely recognized as the leader in its practical application. He initiated and spearheaded SRE practice at AT&T and was involved in developing most of the details. Musa is a Fellow of the IEEE, and an international leader in software engineering.

Jim Waldo is a Senior Staff Engineer with JavaSoft, the Sun Microsystems operating company responsible for the Java language and environment, where he leads a group responsible for the distributed computing infrastructure for Java. A long-time advocate of object oriented programming and the architect of one of the first distributed object request brokers, he has written and spoken extensively on object-oriented programming techniques, distributed computing concepts, and techniques for building robust, reusable software.

Jack C. Wileden is a Professor in the Department of Computer Science at the University of Massachusetts at Amherst whose research centers on tools and techniques supporting seamless integration of advanced capabilities into computing systems. *Alan Kaplan* is on the faculty of the Department of Computer Science at Flinders University in Adelaide, South Australia. His research includes tools and techniques supporting software development, O-O databases, and interoperability.

The SEI's Capability Maturity Model and Software Process Improvement

Bill Curtis

1A The growing functionality expected from modern products and systems has resulted in an exponential growth in the software required to run them. During the 1970s and the early 1980s, software developers focused their investments on advanced workstations, languages, and CASE tools to improve their performance. The benefits of these investments were far lower than promised. A software process movement emerged in the mid-1980s when shortcomings in managing development and maintenance processes were recognized as prime inhibitors of growth in software productivity and quality.

A Realistic, Commercially Robust Process for the Development of Object-Oriented Software Systems — Case Study

Tim Korson

1B The successful adoption of object technology requires far more than simply the adoption of an O-O language. Methods for O-O analysis and design must be selected and configured to fit within an overall software development process. This session examines the required elements of a realistic, commercially robust process for the development of object-oriented software systems.

Software-Reliability-Engineered Testing Practice

John D. Musa

1C SRET is testing of software-based systems which employs reliability objectives and profiles of system use to speed testing while ensuring the necessary reliability. It helps us deal with the pressure to get software-based products to market faster while still meeting customer reliability needs. You will learn how to perform the major activities of SRET: defining "necessary" reliability, developing operational profiles, preparing for testing, executing tests, and interpreting failure data. The tutorial uses a simple but realistic example throughout to illustrate the points.

Java: A Language for Software Engineering

Jim Waldo

1D Java, an object-oriented language designed to allow dynamic loading of binary code over a network of heterogeneous machines, includes features to write robust, error-free code and thus presents a platform for serious software engineering. It is a general purpose, object-oriented language with such features as garbage collection, exceptions, strong typing, and a clean separation between classes and interfaces, allowing secure, robust, reliable programs. We will examine the packages that allow Java to be used in traditional distributed systems programming and for more advanced, Java-only environments that support the Remote Method Invocation system.

Software Interoperability

Jack Wielden, Alan Kaplan

2A Software interoperability is fundamental to such topics as development based on components, reuse, and infrastructure for distributed or network-based systems. As a result, a variety of (often partial) approaches to interoperability have appeared, each offering a potential solution to interoperability problems. Yet what these approaches offer, how they compare, and exactly what problems they are solving is generally unclear. Some foundations for understanding and evaluating interoperability problems and proposed approaches for solving those problems.

Distributed Software Architectures

Jeff Kramer, Jeff Magee

2B Software architectures has been identified as a critical design concern when bridging the gap between system requirements and implementation, particularly in large, complex software systems. Software Architecture is the structure of the components of a program or system, their interrelationships, and principles and guidelines governing their design and evolution over time. It provides a clear and well-defined level at which to describe, understand, and analyse system designs.

Effective Use of COTS (Commercial-Off-the-Shelf) Software Components in Long Lived Systems

W. Morven Gentleman

2C This tutorial looks at kinds of COTS software components that can be used in long lived systems, and the technology available for building around them. The potential benefits and risks of this approach to systems are examined. Modifications of conventional development processes are required to focus on where time and cost expenditures occur, and where risks arise.

Rigorous Requirements for Real-Time Systems: Evolution and Application of the SCR Method

Stuart Faulk, Connie Heitmeyer

3A SCR, a practical formal requirements method, can reduce critical errors and decrease costs in industrial development of embedded, high-assurance systems. Requirements errors remain the most intractable and costly problem in embedded software development. Formal methods offer techniques for early error detection and correction but are widely perceived as impractical for large, complex, embedded systems. SCR has systematically addressed the problems of industrial use of formal requirements methods. The resulting methods and tools offer a proven, practical, industrial-strength approach for formal specification and analysis of critical system requirements.

Software and Business Process Technology

Volker Gruhn, Wilhelm Schaefer

3B The state-of-the-art technology in software process and business process technology and its exploitation in an industrial context includes languages, tools, and substrates (database and operating system functionality) being used to model, analyse, and execute business and software processes. The industrial context is based on building, using, and selling a complete workflow environment including tools for modeling, analysing, and executing workflows. This environment called LEU (LION Environment) is used in various industrial sectors like real estate, software process, and insurance.

An Introduction to OMG/CORBA

Wolfgang Emmerich

3C The Common Object Request Broker Architecture (CORBA) standard adopted by the Object Management Group (OMG) is the industry standard middleware for heterogeneous and distributed object-oriented computing. Participants will be provided with an overview of the different constituents of the CORBA standard. Although the main focus of the tutorial is on the standard and its rationales, participants will also be provided with a brief overview of the most important products implementing CORBA.

Jeff Kramer is a Professor at Imperial College, and head of the Distributed Software Engineering research section. His research interests include requirement analysis techniques, design and analysis methods, software construction languages and software development environments. *Jeff Magee* is assistant director of the Department of Computing at Imperial College. His interests include software engineering of parallel and distributed systems.

W. Morven Gentleman heads the Software Engineering Laboratory in the Institute for Information Technology at the National Research Council of Canada. He has taught at the University of Waterloo and has been among the technical staff at Bell Telephone Laboratories. His Ph.D. from Princeton in 1966 is in Mathematics. His research activities include software engineering, computer architecture, robotics, computer algebra, and numerical analysis.

Stuart Faulk is on the faculty of the University of Oregon's Department of Computer and Information Science. Previously, he led the development of the Consortium Requirements Engineering Method (CoRE) and successfully applied the SCR method in industry. *Connie Heitmeyer* heads the Software Engineering Section of the Naval Research Laboratory's High Assurance Computer Systems branch. She leads research and development efforts in formal methods and CASE tools supporting the construction of real-time, embedded software.

Dr. Volker Gruhn is chief technical officer of a German software house called LION since 1992. He is responsible for a software development department of 150 people. *Dr. Wilhelm Schaefer* is professor of Computer Science at the University of Paderborn, Germany. Prior appointments have been at the University of Dortmund and McGill University in Montreal and a position in industry where he served as the head of an RD department of a medium-size software house focussing on CASE tools and information systems.

Dr. Wolfgang Emmerich is a senior consultant of the OMG representative in Central Europe, LogOn Technology Transfer. Dr. Emmerich has given numerous CORBA training courses to the software industry in many European countries and consulted on the use of CORBA in companies and software houses. Dr. Emmerich is also a Lecturer at City University London, where he teaches distributed systems and software engineering.

Frank E. McGarry is a senior member of the Executive Staff at Computer Sciences Corporation after having spent 28 years at NASA/ Goddard, where he headed the Software Engineering Branch. *Victor Basili* is a professor of computer science at the Institute for Advanced Computer Studies at the University of Maryland at College Park, where he served as chairman for six years. His interests include quantitative approaches for software management, engineering, and quality assurance.

Watts S. Humphrey founded the Software Process Program of the Software Engineering Institute at Carnegie Mellon University. He is a Fellow of the Institute and is a research scientist on its staff. From 1959 to 1986, he was director of programming quality and process at IBM. *James Over*, a Senior Member of the Technical Staff at the Software Engineering Institute (SEI), has worked in several technical areas within the SEI Software Process Program. His interests include software engineering, software process, and quality management. He is the co-author of publications on software process definition and improvement.

Bashar Nuseibeh is head of the Software Engineering Laboratory at Imperial College, London. His research interests are in distributed software engineering and is working on supporting multiple views and managing inconsistencies in software development. *Suzanne Robertson* is a teacher and consultant specialising in modelling techniques for system development. She has co-authored courses on systems analysis and software design for systems, requirements engineering, quality assessment and problem solving. She develops techniques for identifying and reusing requirements patterns.

Mark Ardis, a member of the Technical Staff in the Software Production Research Department at Bell Laboratories, Lucent Technologies, taught software engineering courses at the University of Illinois at Urbana-Champaign, the Wang Institute of Graduate Studies, and Carnegie Mellon University (CMU). *David Weiss* is the head of the Software Production Research Department at Bell Laboratories, Lucent Technologies. He has been director of reuse and measurement for the Software Productivity Consortium.

Shari Lawrence Pfleeger is president of Systems/Software Inc., a consultancy specializing in software engineering research and technology transfer. In addition, she is a visiting professorial research fellow at City University's Centre for Software Reliability.

The Experience Factory: How to Build and Run One

Frank McGarry, Vic Basili

4A This course presents the fundamental concepts behind software process and product improvement using measurement and evaluation in an Experience Factory Organization. It will provide a set of examples associated with understanding the software engineering process, product, and environment, improving it over time and packaging experience in the form of models and measures to create an experience base that can be reused by future projects. It discusses how this approach is being used in the Software Engineering Laboratory (SEL) at NASA/ Goddard Space Flight Center and how it has been expanded to other NASA sites and to other production environments in private corporations such as Computer Sciences Corporation (CSC).

The Personal Process (PSP)SM in Software Engineering

Watts S. Humphrey, James W. Over

4B The PSP method uses quality management principles and the Capability Maturity Model (CMM)SM framework and uses sound engineering principles in software development and maintenance work. The principal message of the PSP is that engineers should use process management concepts to identify and perfect the methods that are most effective for them. Engineers using PSP significantly improve the quality of their work, learn how to plan their projects, and improve their productivity. Average quality improvements of five to ten times are normal, as are productivity improvements of 25% or more.

Personal Software Process, PSP, Capability Maturity Model, and CMM are service marks of Carnegie Mellon University.

Making Requirements Measurable

Bashar Nuseibeh, Suzanne Robertson

4C Participants in this interactive full-day tutorial examine measurability by building a requirements specification for a familiar system. After presenting an overview of requirements engineering activities, the tutorial focuses on how to measure requirements for testability, relevance, completeness, consistency, coherency, traceability and satisfaction. A requirements template is used as a guide to discovering requirements and building the specification. A requirement is "measurable" if there is an unambiguous way of determining whether a given solution fits that requirement.

Defining Families: The Commonality Analysis

Mark Ardis, David Weiss

4D One approach of systematically engineering software domains is to develop families of software and to invest in facilities for rapidly producing family members. This full-day tutorial describes the commonality analysis process, a systematic approach to analyzing families. The result of the analysis forms the basis for designing reusable assets that can rapidly produce family members. A practice commonality analysis will be guided by experienced users of the process.

Evaluating Software Technology

Shari Lawrence Pfleeger

4E This tutorial examines the issues involved in evaluating the effects of software methods and tools on our products, processes and resources. It reviews measurement theory and the basics of experimental design and analysis to suggest guidelines for how we should investigate the benefits and costs of our actions. Principles are illustrated with actual industrial examples, including two in-depth case studies. In addition to designing our own investigations, the tutorial enables us to understand when the results of others apply to our own situations.

A Survey of Object-Oriented Analysis and Design Methods

Martin Fowler

5A Over the last ten years many trees have been sacrificed for writings on OO methods. This tutorial gives a guided tour of these writings, pointing out the common techniques, some less-known but valuable techniques, and where to go for more information. A framework shows how various techniques fit together. We then explore techniques for structural, architectural, and behavioral modeling; together with process techniques used in OO development. The tutorial uses the Unified Modeling Language as its base, but also discusses important additional techniques such as model perspectives, design by contract, event modeling, and the translation process.

Simplifying the Evolution of Java Programs

Linda Seiter, Karl Lieberherr, Doug Orleans

5B We present an in-depth look at the ability of existing models and languages to support different forms of reuse and evolution. We present techniques for achieving dynamic behavior in a static, class-based model and language. We present the public-domain programming tool Demeter/Java which implements a special kind of context object, called a visitor object. To enable the direct expression of visitor objects and to express traversal strategies for objects in Java, we present a simple extension to Java (the Demeter/Java language) and show how it is translated back into Java. The benefits of Demeter/Java are that programs become more flexible, both structurally and behaviorally — and also shorter.

A Primer on Empirical Studies

Dewayne E. Perry, Adam A. Porter, and Lawrence G. Votta

5C A sound empirical basis for software and process engineering and research by focusing on the basic characteristics of empirical studies. The key to empirical work is credibility — everything else follows from this primary characteristic. We advocate a model in which there is more control involved, resulting in more detailed information gathered. Rather than emphasize general phenomenology, we emphasize trying to find underlying mechanisms. Attendees assess the credibility of empirical work either as reported in the software engineering literature or as done by themselves and to apply the results to their own work.

Reverse Engineering Strategies for Software Migration

Hausi A. Muller

5D The need to maintain and improve software and information systems has risen dramatically over the past decade. Dealing with old software systems that are billion-dollar assets to corporations and governments is a critical problem. Migrating and reengineering involves capturing, preserving, and extending knowledge about software, analyzing and understanding software, and finally changing, improving, and evolving software. Reverse engineering approaches have been particularly useful in the arena of reverse engineering; the process of generating new information about software.

Tailoring the CMM for Small Businesses, Small Organizations, and Small Projects

Judith G. Brodman, Donna L. Johnson

5E A small organization or project faces issues when implementing a software process improvement program. Alternative practices that organizations satisfy the intent of the SEI's Capability Maturity Model (CMM) practices and goals and addresses the setting up and sustaining of a viable process improvement program. The growth pattern shows areas of improvement as an organization matures — vertical linkages between process areas through the levels of the CMM and the evolutionary definition of metrics through the levels of maturity.

Martin Fowler is an independent consultant who has pioneered the use of Object Oriented analysis and design for business information systems. These include health care for the UK National Health Service, derivatives trading for Citibank, and payroll for Chrysler. He is also a leader in developing analysis patterns and is the author of the book *Analysis Patterns: Reusable Object Models*.

Karl Lieberherr is a Professor in the College of Computer Science at Northeastern University. His current research interests are methods and tools for developing adaptable software for open systems, with special emphasis on adaptive software. *Linda Seiter* is a visiting professor at Boston College. Her current research interests include semantics of languages that support evolution and reuse. *Doug Orleans* is a PhD student at Northeastern University, and the principal architect of Demeter/Java.

Dewayne E. Perry, Adam A. Porter, and Lawrence G. Votta have a rich background in both software engineering research and practice, covering the entire range of technical and management aspects of engineering large and small software systems, and a broad range of software engineering research problems. In particular, they have been instrumental in forging a new approach to software engineering experimentation in both in vitro and in vivo contexts.

Dr. Hausi Muller is an Associate Professor of Computer Science at the University. He is a principal investigator of CSER (Centre for Software Engineering Research), a Canadian Centre for Excellence sponsored by NSERC, NRC, and industry including IBM. His research interests include software evolution, software reverse engineering, software architecture, program understanding, software re-engineering, and maintenance. He served as Program Co-Chair for ICSM-94, CASE-95, and WPC-96. He is on the Editorial Board of IEEE TSE.

Judith G. Brodman and *Donna L. Johnson* are CEO and President, respectively, of LOGOS International, Inc. They consult with software organizations on software process improvement initiatives. They have also researched the return-on-investment for software process improvement initiatives and issues facing small organizations in their improvement efforts.

Wolfgang Pree is Associate Professor at the University of Linz. He is author of *Design Patterns for Object-Oriented Software Development* (Addison-Wesely/ACM Press, 1995) and of *Framework Patterns* (SIGS Books, New York City, 1996). *Hermann Sikora* is a managing director of RACON Linz Software, Inc., a company owned by and producing software for the largest private bank consortium in Austria. Sikora holds a Ph.D. in computer science and a degree in management information systems.

Stefan Fischer is a graduate of the University of Mannheim. Formerly a research assistant at the Institute of Applied Computer Science (Department for Computer Networks) of the University of Mannheim, he is a postdoctoral fellow at the University of Montreal, Canada. *Stefan Leue* received his Master's Degree in Computer Science from the University of Hamburg in 1990, and his Ph.D. degree from the University of Berne. Formerly a research associate and doctoral candidate at the University of Berne, he is an Assistant Professor at the University of Waterloo.

Amrit L. Goel is Professor of Electrical and Computer Engineering and a member of the Computer and Information Science Faculty at Syracuse University - Dr. Goel's current interests are in software reliability and testing, fault tolerant software, and performance modeling of parallel systems. He was recently elected a fellow of the IEEE for his contributions to the reliability of computer software.

Vaclav Rajlich is a professor and former chair of the Department of Computer Science at Wayne State University, Detroit, Michigan. Before that, he was an associate professor at the University of Michigan in Ann Arbor, and software manager at the Research Institute for Mathematical Machines in Prague, Czech Republic.

Marilyn Bush, an independent consultant, is one of the authors of the revised SEI Capability Maturity Model as well as an author of the SEI's CMM Introductory Course and a qualified SEI Lead Assessor. She was recently a member of the SEI team tasked to revamp the SEI Assessment Method and Lead Assessor Course, and she is a certified instructor for the SEI CMM Introductory Course and one of three people worldwide now certified to teach the SEI Lead Assessor Course.

Design Patterns for Object-Oriented Software Development

Wolfgang Pree, Hermann Sikora

6A Design patterns support the development and reuse of extensible OO software components. They represent a complimentary enhancement of existing OO analysis and design (OOAD) methods. The tutorial will give an overview of state-of-the-art design patterns approaches, including pattern catalogs and framework patterns. A selection of useful patterns will be discussed in detail. The tutorial will also introduce hot spot cards, which have proved to be a useful communications vehicle between domain experts and software engineers in order to exploit the potential of design patterns.

Formal Methods for Broadband and Multimedia Systems

Stefan Fischer, Stefan Leue

6B Formal methods have been applied successfully to specify "traditional" communication protocols, services, and network applications. With high-speed networks, new distributed applications impose requirements on the communication subsystem that are different from those on traditional systems. To meet application-level performance requirements, highly efficient techniques to implement communication software have been developed. Formal description techniques (in particular, SDL and Estelle) can be used for the requirements specification, design, and implementation stages in the life-cycle of broadband communication systems.

Software Engineering Data Analysis Techniques

Amrit Goel

6C During the past few years, there has been an increasing emphasis on the use of quantitative measures for monitoring and controlling software projects. As a result, collection of product and process data has become a standard practice in major software development organizations. While there has been a growing emphasis on the collection of metrics data, relatively very little work has been done on the systematic use of appropriate data analysis techniques. For metrics or measurement undertakings to be successful, we feel that it is essential to address both the data collection and analysis activities as mutually inseparable.

Comprehension and Evolution of Legacy Software

Vaclav Rajlich

6D Legacy systems have one or several of the following attributes: they were implemented many years ago, their technology became obsolete, their structure deteriorated, they represent a large investment, they contain business rules not recorded elsewhere, they cannot be easily replaced, or the original authors are not available. Software comprehension typically consumes more than a half of the difficult effort of maintaining legacy systems. The tutorial will give an overview of the available techniques and tools.

European and American Software Process Maturity Models and Assessments

Marilyn Bush

6E How do assessments generate increased productivity? How do software process assessments work? What principles underlie all current software process improvement models worldwide? Four major software process improvement models are on the international scene: ISO 9001 and 9000 3, Bootstrap, the SEI Capability Maturity Model (CMM) and Software Process Improvement and Capability Determination (SPICE). All these models acknowledge similar state of the practice principles of good software quality development process, but each involves a different take on assessing organizational process maturity.

TECHNICAL PROGRAM

TUESDAY, MAY 20, 1997

Morning

- (1A) OPENING SESSION AND KEYNOTE ADDRESS: **Beyond Software Engineering**, Ed Yourdon
- (2A) TECHNICAL PAPERS: **ICSE/SSR Joint Session**
- (2B) TECHNICAL PAPERS: **Exploiting the Internet**
- *An Architecture for WWW-based Hypercode Environments*: Gail E. Kaiser, Stephen S. Dossick, Wenyu Jiang, Jack Jingshuang Yang (Columbia University, USA)
 - *Anywhere, Anytime Code Inspections: Using the Web to Remove Inspection Bottlenecks in Large-Scale Software Development*: Dewayne E. Perry (Bell Labs, USA), James Perpich (Lucent Technologies, Inc., USA), Adam Porter (University of Maryland, USA), L. G. Votta (Bell Labs, USA), Michael W. Wade (Lucent Technologies Inc., USA)
 - *Designing Distributed Applications with Mobile Code Paradigms*: Giovanni Vigna and Antonio Carzaniga (Politecnico di Milano, Italy), Gian Pietro Picco (Politecnico di Torino, Italy)
- (2C) LESSONS IN ORGANIZATIONS: **Object Technology**
- *Tailoring OMT for an Industry Software Project*: Jeffrey Melanson (Siemens Medical Systems, USA), Robert L. Nord (Siemens Corporate Research, USA), Dilip Soni (Siemens Corporate Research, USA)
- (2D) MINI-TUTORIAL: **TBA**
- (2E) FORMAL RESEARCH DEMOS: **Analysis**
- *Verification of Concurrent Software with FLAVERS*: Gleb Naumovich (University of Massachusetts, Amherst, USA), Matthew Dwyer (Kansas State U., USA), Lori A. Clarke, Leon J. Osterweil (University of Massachusetts, Amherst, USA)
 - *Nitpick, A Tool for Interactive Design Analysis*: Craig Damon (Carnegie Mellon U., USA)

Afternoon

- (3A) TECHNICAL PAPERS: **Formal Specifications**
- *An Object-Oriented Modeling Method for Algebraic Specifications in CafeOBJ*: Shin Nakajima (NEC Corp., Japan), Kokichi Futatsugi (Japan Advanced Institute of Science & Technology, Japan)
 - *Formalizing and Integrating the Dynamic Model within OMT*: Betty H.C. Cheng, Enoch Y. Wang, Heather A. Richter (Michigan State University, USA)
 - *Introducing Formal Specification Methods in Industrial Practice*: Luciano Baresi, Alessandro Orso, Mauro Pezzè (Politecnico di Milano, Italy)
- (3B) TECHNICAL PAPERS: **Reliability**
- *Choosing a Testing Method to Deliver Reliability*: Lorenzo Strigini (City University, U.K.), Phyllis Frankl (Polytechnic University, U.K.), Dick Hamlet (Portland State University, USA), Bev Littlewood (City University, U.K.)
 - *Re-estimation of Software Reliability After Maintenance*: Andy Podgurski (Case Western Reserve Univ., USA), Elaine J. Weyuker (AT&T Research Labs, USA)
 - *A Study on the Failure Intensity of Different Software Faults*: Kazuyuki Shima, Shingo Takada, Ken'ich Matsumoto, Koji

- Torii (Nara Institute of Science and Technology, Japan)
- (3C) PANEL: **Platforms for Software Execution: Databases vs. Operating Systems vs. Browsers**
- Chair, Richard Selby (UC Irvine, USA)
- Panelists (partial list): Randy Davis (MIT, USA); Paul Dorsey (Dulcian, Inc., USA)
- (3D) EXPERIENCE REPORTS: **Large Scale Systems**
- Chair: Jerry Fiddler, Wind River Systems
- [Title TBD], Eric Sumner (Lucent Technologies)
- Architecting Families of Software Intensive Products*, Alexander Ran (Nokia Research Center) and Hamish Kellock (Nokia Telecommunications)
- (3E) FORMAL RESEARCH DEMOS: **Software Evolution**
- *Endeavors: A Process System Infrastructure*: Arthur S. Hitomi, Gregory Alan Bolcer, Richard N. Taylor (University of California, Irvine, USA)
 - *Argo: A Design Environment for Evolving Software Architectures*: Jason E. Robbins, David M. Hilbert, David F. Redmiles (University of California, Irvine, USA)
- (4A) TECHNICAL PAPERS: **Inspections and Reviews**
- *An Empirical Study of Communication in Code Inspections*: Carolyn B. Seaman, Victor R. Basili (University of Maryland, USA)
 - *Experiences with Distributed, Asynchronous Software Inspection*: Michael V. Stein, John Riedl (University of Minnesota, USA), Sören J. Harner (ICEM Systems, GmbH, Germany), Vahid Masheyekhi (DELL Computer Corp., USA)
 - *Assessing software review meetings: A controlled experimental study using CSRS*: Philip M. Johnson, Danu Tjahjono (University of Hawaii, USA)
 - *Understanding the Effects of Developer Activities on Inspection Interval*: Harvey P. Siy, Adam Porter (University of Maryland, USA), Lawrence G. Votta Jr. (Lucent Technologies, USA)
- (4B) TECHNICAL PAPERS: **Process Improvement**
- *Measuring Requirements Testing*: Theodore Hammer, Lawrence E. Hyatt (NASA Goddard Space Flight Center, USA), Lenore Huffman (SATC, USA), Linda Rosenberg (Unisys Federal Systems, USA)
 - *Integrating Measurement with Improvement: An Action-Oriented Approach*: Jo Ann Lane (Science Applications International Corporation, USA), David Zubrow (Carnegie Mellon University, USA)
 - *Total Software Process Model Evolution in EPOS*: Minh N. Nguyen, Alf Inge Wang, Reidar Conradi (Norwegian University of Science & Technology, Norway)
 - *An Improved Process for the Development of PLC Software*: Flavio Bonfatti, Paola Daniela Monari (University of Modena, Italy), Gianni Gadda (Democenter Srl, Italy)
- (4C) LESSONS IN ORGANIZATIONS: **Legacy Systems**
- *Manipulating Recovered Software Architecture Views*: Alexander S. Yeh, David R. Harris, Melissa P. Chase (Mitre Corporation, USA)
 - *Lessons on Converting Batch Systems to Support Interaction Testing*: Gregory Zelesnik, Robert DeLine, Mary Shaw (Carnegie Mellon University, USA)
 - *Applying Design of Experiments to Software Testing*: W. K. Ehrlich, I. S. Dunietz, B.D. Szablak (AT&T NCS OTC, USA), C. Malloes (AT&T Laboratories, USA), A. Iannino

TECHNICAL PROGRAM

(Pipeline Associates, USA)

- *A Theory of Probabilistic Functional Testing*: Gilles Bernot (Université d'Evry, France), Laurent Bouaziz (CERMICS-ENPC, France), Pascale LeGall (Université d'Evry, France)

(4D) PANEL: **Advantages of Mature Software Development Organizations**

Chair: Marie Silverthorn (Texas Instruments, USA)

Panelists (partial list):

Bill Curtis (TeraQuest Metrics, USA)

Jeff Perdue (Inst. for Software Process Improvement, USA)

(4E) FORMAL RESEARCH DEMOS:

Software Evolution and Maintenance

- *Automatic Monitoring of Software Requirements*, Martin S. Feather, K. Narayanaswamy, Don Cohen, Stephen S. Fickas (Computing Services Support Solutions, USA)
- *Preventive Program Maintenance in Demeter/Java*, Karl Lieberherr, Doug Orleans (Northeastern U., USA)

Evening

RECEPTION: **Sheraton Hotel**

WEDNESDAY, MAY 21, 1997

Morning

(5A) KEYNOTE ADDRESS: **Java and the Evolution of Web Software** Guy Steele (Sun Microsystems, USA)

(6A) TECHNICAL PAPERS: **Static Analysis**

- *Analyzing Partially-Implemented Real-Time Systems*: George S. Avrunin (University of Massachusetts, USA), James C. Corbett (University of Hawaii, USA), Laura K. Dillon (University of California-Santa Barbara, USA)
- *Constructing Multi-Formalism State-Space Analysis Tools*: Michal Young (Purdue University, USA), Mauro Pezzè (Politecnico di Milano, Italy)
- *Software Deviation Analysis*: Jon Damon Reese, Nancy G. Leveson (University of Washington, USA)

(6B) TECHNICAL PAPERS: **Metrics**

- *A Predictive Metric Based on Discriminant Statistical Analysis*: Maurizio Pighin, Roberto Zamolo (Università degli Studi di Udine, Italy)
- *Software Metrics for Distributed Development*: Bernd Bruegge, Allen H. Dutoit (Carnegie Mellon University, USA)
- *Modeling the Cost of Rework in a Library of Reusable Software Components*: Walcelio L. Melo (Centre de Recherche Informatique de Montréal), Victor Basili (University of Maryland, USA), Steven Condon (CSC, USA), Khaled El Eman (FIESE, Germany)

(6C) TUTORIAL: **TBA**

(6D) JAVA PANEL: **Java Development Environments**

Chair: Tony Wasserman (tentative)

Panelists (partial list): Peter Kellogg-Smith (Asymetrix), Leo Lucas (Aimtech), Laurent Visconti (Metrowerks)

(6E) FORMAL RESEARCH DEMOS:

Reverse Engineering and Maintenance

- *Rigi: A Visualization Environment for Reverse Engineering*, Margaret-Anne Storey, Kenny Wong, Hausi Muller, University of Victoria, Canada
- *An Object-Oriented Testing and Maintenance Environment*, Pei Hsia, David Kung, University of Texas, Arlington, USA

(7A) EXPERIENCE REPORTS: **Process**

- *A New Software Project Simulator Based on Generalized*

Stochastic Petri-net: Shinji Kusumoto, Osamu Mizuno, Tohru Kikuno (Osaka University, USA), Yasunari Takagi, Keishi Sakamoto (OMRON Corporation, Japan)

- *The Criticality of Modeling Formalisms in Software Design Method Comparison*: Rodion M. Podorozhny, Leon Osterweil (University of Massachusetts, USA)

(7B) TECHNICAL PAPERS: **Hardware/Software Issues**

- *A Specification of Software Controlling a Discrete-Continuous Environment*: Viktor Friesen, Stefan Jahnichen, Matthias Weber (Technische Universität Berlin, Germany)
- *Automatic Checking of Instruction Specifications*: Mary Fernandez (AT&T Research, USA), Norman Ramsey (University of Virginia, USA)

(7C) LESSONS IN ORGANIZATIONS: **Architecture Recovery and Reverse Engineering**

- *Software Architecture Recovery of Embedded Software*, Wolfgang Eixelsberger, Lasse, Warholm (ABB Corporate Research, Norway), Rene Kloesch, Harald Gall (Technical University of Vienna, Austria)
- *Integrating Forward and Reverse Object-Oriented Software Engineering*, Christoph Welsch (ABB Corporate Research, Germany), Alexander Schalk (Adtranz, Germany), Stefan Kramer (ABB Color Emag Schaltanlagen, Germany)

(7D) PANEL: **Large Systems Experience II**

Chair: to be named

Measuring Software Scrap and Rework — A Useful Perspective for Managing an Iterative Development Process, Walker Royce (Rational Software); Ian M. Graham

(7E) FORMAL RESEARCH DEMOS: **Requirements**

- *SCR*: A Toolset for Specifying, Verifying, and Validating Software Requirements*. Connie Heitmeyer, James Kirby, Jr., Bruce Labaw (Naval Research Lab, USA).
- *GRAIL/KAOS: a Requirements Engineering Environment*. Robert Darimont, Emmanuelle Delor, Philippe Massonet, Axel van Lamsweerde (University Louvain, Belgium)

(8A) PLENARY AWARD SESSION

Chair: W. Richards Adrion (University of Massachusetts, Amherst, USA)

- ACM, ACM/SIGSOFT, IEEE and IEEE/TCSE awards
- ICSE 9 paper award

• *Software Processes are Software Too*, Leon J. Osterweil (University of Massachusetts, Amherst, USA)

• *Process Models, Process Programs, Programming Support: Response to An ICSE9 Keynote Address by Lee Osterweil*, M.M. Lehman (Imperial College, England)

Afternoon

(9A) TECHNICAL PAPERS: **Reverse Engineering and Program Understanding**

- *Lackwit: A Program Understanding Tool Based on Type Inference*: Robert O'Callahan, Daniel Jackson (Carnegie Mellon University, USA)
- *Assessing Modular Structure of Legacy Code Based on Mathematical Concept Analysis*: Christian Lindig, Gregor Snelting (Technische Universität Braunschweig, Germany)
- *Visualizing Interactions in Program Executions*: Dean F. Jerding, John T. Stasko (Georgia Institute of Technology, USA), Thomas Ball (Bell Laboratories, USA)

(9B) TECHNICAL PAPERS:

User Interface and Specifications

- *Early Specification of User-Interfaces: Toward a Formal Approach*: J.-P. Jacquot, D. Quesnot (Centre de Recherche en Informatique de Nancy, France)
- *Automated Analysis of Requirement Specifications*: William

TECHNICAL PROGRAM

M. Wilson (Software Assurance Technology Center/GSFC, USA), Linda H. Rosenberg (Unisys Federal Systems/GSFC, USA), Lawrence E. Hyatt (NASA/GSFC, USA)

- *Integrating Support for Temporal Media into an Architecture for Graphical User Interfaces*: T.C. Nicholas Graham, Tore Urnes (York University, Canada)

(9C) LESSONS IN ORGANIZATIONS: **Management Issues**

- *Prioritizing Software Requirements in an Industrial Setting*: Kevin Ryan (University of Limerick, Ireland), Joachim Karlsson (SoftLab, Sweden)

- *Developmental Assurance with the Systems Security Engineering Capability Maturity Model*: Rick Hefner (TRW, USA)

- *BOOTSTRAP: Four Years of Assessment Experience*: Hans StienenS, Ernst Lebsanft, Franz Engelmann (YNSPACE AG, Switzerland)

(9D) PANEL: **Cooperative Software Development**

Chair: Kanth Miriyala, Andersen Consulting

Speakers: Prasun Dewan (University of North Carolina, USA), Philip Johnson (University of Hawaii, USA), Remo Pareschi (Rank Xerox Research Centre, France)

(9E) FORMAL RESEARCH DEMOS:

Component Integration

- *An Architecture Design Environment for Component-Based Software*: Francois Bronsard, Douglas Bryan, W. (Voytek) Kozaczynski, Edy S. Liongosari, Jim Q. Ning, Asegeir Olafsson, (Andersen Cons., USA)

- *Building Engineering Tools on Shrink-Wrapped Packages*: Kevin Sullivan, Jake Cockrell (University of Virginia, USA)

Evening

Conference Reception at the Top of the Hub

THURSDAY, MAY 22, 1997

Morning

(10A) KEYNOTE ADDRESS:

Software Engineering That Matters to People

Mark Weiser (Xerox PARC, USA)

(11A) TECHNICAL PAPERS: **Analysis of C & C++**

- *An Investigation into Coupling Measures for C++*: Lionel Briand Fraunhofer (IESE, Germany), Prem Devanbu (AT&T Research, USA), Walcelio Melo (CRIM, Canada)

- *Incremental Analysis of Side Effects for C Software Systems*: Jyh-shiarn Yur, Barbara G. Ryder (Rutgers University, USA), William A. Landi (Siemens Corporate Research, USA), Phil Stocks (Rutgers University, USA)

- *Flow Insensitive C++ Pointers and Polymorphism Analysis and its Application to Slicing*: Paolo Tonella, Giuliano Antoniol, Roberto Futem (IRST, Italy), Ettore Merlo (Ecole Polytechnique C.P., Canada)

(11B) TECHNICAL PAPERS: **Economic and Legal Issues**

- *The Effect of Department Size on Developer Attitudes to Prototyping*: J. M. Verner (City University of Hong Kong,

Hong Kong), N. Cerpa (University of New South Wales, Australia)

- *Copyright in Shareware Software Distributed on the Internet — The Trumpet Winsock Case*: Cristina Cifuentes (University of Queensland, Australia), Anne Fitzgerald (University of Tasmania, Australia)

- *On the Economics of Mass-Marketed Software*: Richard J. Botting (California State University at San Bernadino, USA)

(11C) MINI-TUTORIAL: **TBA**

(11D) MINI-TUTORIAL: **An Introduction to CORBA**

Speaker: Richard Soley (Object Management Group, USA)

(11E) FORMAL RESEARCH DEMOS: **Environments**

- *Developing Graphical SE Tools with PROGRES*, Andy Schuerr, Andreas J. Winter (University Aachen, Germany)

- *TINA ACE: an environment for Specifying, Developing and Generating TINA Services*. Piergiorgio Bosco, Giovanni Martini, Corrado Moiso (CSELT, Italy)

Afternoon

(12A) TECHNICAL PAPERS:

Object-Oriented Technology

- *Abstract Syntax from Concrete Syntax*: David S. Wile (University of Southern California, USA)

- *Open Implementation Design Guidelines*: Gregor Kiczales, John Lamping, Cristina Videira Lopes, Anurag Mendhekar (Xerox Palo Alto Research Center, USA), Gail Murphy (University of Washington, USA)

- *Hooking into Object-Oriented Application Frameworks*: Garry Froehlich, H. James Hoover, Ling Liu, Paul Sorenson (University of Alberta, Canada)

(12B) EXPERIENCE REPORTS: **Testing & Analysis**

- *Shaky Foundations? Using Formal Methods to Reason about Architectural Standards*: Kevin Sullivan (University of Virginia, USA), John Socha (Socha Computing Inc., USA)

- *Model-Checking of Real-Time Systems: A Telecommunications Application*: Raveev Alur, Lalita Jategaonkar Jagadeesan (Bell Laboratories, USA), Joseph J. Kott, James E. Von Olnhausen (Lucent Technologies, USA)

- *Specification-based Testing of Reactive Software: Tools and Experiments*: Lalita Jategaonkar Jagadeesan (Bell Laboratories, USA), Adam Porter (University of Maryland at College Park, USA), Carlos Puchol (The University of Texas at Austin, USA), J. Christopher Ramming (AT&T Laboratories, USA), Lawrence G. Votta (Bell Laboratories, USA)

(12C) LESSONS IN ORGANIZATIONS: **Process Issues**

- *Implementing Cleanroom Software Engineering into a Mature CMM-Based Software Organizations*: Robert Oshana, Texas Instruments, USA, Frank P. Coyle (Southern Methodist University, USA)

- *Redesigning the Systems Development Process*: Debra Bond, Patty Rishi (AT&T, USA)

(12D) PANEL: **Databases and the World-Wide Web**

Chair: Gail Kaiser (Columbia Univ., USA)

Panelists (partial list): Larry Alston (Object Design, USA); Tom George (Centura Software, USA); Doron Sheffer (NetDynamics, USA)

(13A) PANEL: **Closing Remarks**

EXHIBITION

Some exhibit space remains, and applications will be accepted on a first-come, first-served basis. While the exhibition is new to ICSE, interest has been high, so prompt registration is recommended. The ICSE 97 Exhibition is intended for commercial vendors of products and services of interest to the software engineering community.

In addition to booth space, each exhibitor receives the following benefits:

- One full conference pass at a reduced rate
- A complete list of registered conference attendees, including title and address

CONTACT

Susan Hines
Nth Degree, Inc.
490 Boston Post Road
Sudbury, MA 01776 USA

+1-508-443-3330, extension 1227

+1-508-443-4715 (fax)

Exhibitors:

- *Addison-Wesley*
- *AGCS, Inc., USA*
- *Birkauser*
- *Extended Intelligence, Inc.*
- *IFAD, Denmark*
- *Kluwer Academic Publishers*
- *Neuma Technology Corp., USA*
- *Softstar Systems*
- *TakeFive Software, Inc.*
- *Visible Systems, USA*

RESEARCH DEMONSTRATIONS

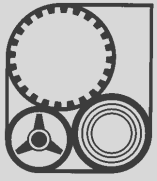
COORDINATORS: Anne Duncan, Digital Equipment Corporation; Volker Gruhn, LION GmbH;
Barbara Lerner, University of Massachusetts

Research demonstrations enable conference participants to view research systems in actions and to discuss the systems with the people who created them. Research demonstrations are intended to show early implementations of novel software engineering concepts. Demonstrations have been selected on the basis of their potential as demonstrations, their originality, and their promise for bringing new and exciting ideas to the conference. We consider any system which is at least six months away from appearing in a commercial product as a "research system." Commercial systems are not accepted as research demonstrations.

Formal Research Demonstrations will be given as part of the technical track.

Informal Research Demonstrations will be given by research groups from a number of universities and industrial laboratories in a special exhibition area.

- *Andersen Consulting, USA*
- *Brown University, USA*
- *Carnegie Mellon University, USA*
- *Computing Services Support Solutions, USA*
- *CSELT, Italy*
- *The Foxboro Company, USA*
- *Hughes Technical Services and Area Systems, Inc.*
- *Naval Research Lab, USA*
- *Northeastern University, USA*
- *Pohang University of Science and Technology, Korea*
- *University Aachen, Germany*
- *University of California, Irvine, USA*
- *University Louvain, Belgium*
- *University of Massachusetts & Flinders University, Australia*
- *University of Massachusetts & Kansas State University, USA*
- *University of Texas, Arlington, USA*
- *University of Victoria, Canada*
- *University of Virginia, USA*



ICSE98

The 20th International Conference on Software Engineering

April 19-25, 1998 • Kyoto International Conference Hall Kyoto, JAPAN

SPONSORED BY

Science Council of Japan • Information Processing Society of Japan
Japan Society for Software Science and Technology
IEEE Computer Society • ACM Special Interest Group on Software Engineering

ICSE's twentieth birthday affords an opportunity to reflect on profound changes of the past two decades.

Since ICSE's founding, politics and technology have converged to shatter once formidable barriers to international cooperation. The collapse of the Berlin Wall symbolizes the end of the Cold War, and links in a Web page are a metaphor for our enhanced capacity for global information exchange.

In this spirit, we have worked hard to ensure that ICSE98 will see an unprecedented increase in the level of international participation. In particular, we will give a greater voice to members from the Asian-Pacific region, an area under-represented in the past. Yet our outreach must extend beyond geography.

As we continue to build bridges to other software disciplines, researchers and practitioners in allied fields will benefit from an understanding of the contributions that software engineering can make to their work. In turn, we must address their problems in our research. New collaborations between academia and industry will also enrich ICSE98 and our profession as a whole.

I hope you will take advantage of this unique opportunity to meet new colleagues, exchange ideas, and explore the nearly limitless possibilities that lie ahead. Join me in Kyoto for ICSE98 and share the excitement that comes from forging new links.

—Katsuro Inoue

For more information: <http://icse98.aist-nara.ac.jp/>
mail: icse98-info@itc.aist-nara.ac.jp

SUBMISSIONS AT A GLANCE

Participation Category	Description	Deadline
Technical Papers	<i>Report research or practical experiences</i>	20 August 1997 (electronic abstract) 3 September 1997 (full paper due)
Panels	<i>Discuss and debate pressing issues</i>	1 August 1997
Software Engineering in Organizations: Lessons and Status Reports	<i>Summaries of work taking place in institutions, organizations, or groups</i>	1 December 1997
Tutorials	<i>Teach Software Engineering techniques and theory</i>	15 September 1997
Workshops	<i>Discuss focused topic in a small-group setting</i>	1 July 1997
Exhibition	<i>Show commercial and research tools</i>	1 January 1998
Posters & Research Demonstrations	<i>Present late-breaking results and ongoing work</i>	1 January 1998
Doctoral Symposium	<i>Ph.D. students share their work</i>	30 September 1997
Asia Pacific Forum	<i>Discuss emerging technology and industry</i>	30 June 1997

ICSE98 Committee Chairs GENERAL CHAIR: Koji Torii (NAIST) • EXECUTIVE OFFICE: Masaki Koyama (NAIST) • TREASURER: Seishiro Tsuruho (NTT Data) • OPERATING COMMITTEE: Norihisa Doi (Keio U.) • PUBLICITY: Katsuro Inoue (Osaka U.) LOCAL ARRANGEMENT: Kiyoshi Agusa (Nagoya U.) • REGISTRATION: Ryohei Nakatsu (ATR) • PUBLICATION: Tsuneo Ajisaka (Kyoto U.) • TECHNICAL SERVICE: Nobuo Araki (NTT) • TECHNICAL COMMITTEE: Takuya Katayama (JAIST), David Notkin (U. of Washington) • PROGRAM COMMITTEE: Kokichi Futatsugi (JAIST), Richard Kemmerer (UCSB) • WORKSHOP: Mikio Aoyama (NIIT) • TUTORIAL: Motoshi Saeki (TITEC), Bashar Nuseibeh (Imperial College) • EXHIBITION: Shinichi Honiden (Toshiba), Bernd J. Kraemer (Fern U. Hagen) • DOCTOR SYMPOSIUM: Tetsuo Tamai (Tokyo U.), Simon Kaplan (U. of Queensland) • POSTER & DEMO: Yoshiaki Fukazawa (Waseda U.) • SE IN ORGANIZATION: Motoei Azuma (Waseda U.), Ryoichi Hosoya (NTT) • ASIA PACIFIC: Koichi Kishida (SRA), Keiji Araki (Kyushu U.), Dines Bjorner (UNU/IIST)

ICSE 97 HOTEL REGISTRATION FORM

Sheraton Boston Hotel & Towers, Boston, MA
May 17-23, 1997

Reservations must be received no later than April 18, 1997.

Name: _____
Address: _____

City: _____
State: _____ Postal Code/Zip: _____ Country: _____
Telephone: _____ Fax: _____

Arrival Date: _____ Departure Date: _____

Expected Arrival Time: _____

Sharing With: _____

Sheraton Club International ID # _____

Please Check One: If choice not available, alternate rate will be assigned.

- Accommodations** ☐ Single (one person): \$160 ☐ Double (two persons): \$175 ☐ Government: \$105
- Currently a State/City tax of 9.7% is applicable and subject to prior credit arrangements at time of registration.
 - Check-in time will be after 3 pm on date of arrival.
 - Check-out time is 12:00 pm
- ☐ I desire a wheelchair accessible room. ☐ I prefer a non-smoking room.

To receive these rates, you must inform the hotel that you are attending the ICSE Conference.

Payment ☐ Check or First night deposit and tax enclosed or
☐ Charge my American Express Card, Visa, Mastercard, Diners Club, or Discover for first night's deposit and tax.

Credit Card Type: ☐ Amex ☐ Visa ☐ MC ☐ DC ☐ DISC

Credit Card # _____

Expiration Date: _____

Card Holder's Authorized Signature: _____

Please mail, FAX, or telephone the
Sheraton Boston Hotel & Towers
39 Dalton Street
Boston, MA 02199
Tel: 1-800-325-3535 or 1-617-236-2000
Fax: 1-617-236-6061

Cancellations will be accepted up until 48 hours prior to arrival.

To obtain discounted airfares, call
American Airlines
1-800-433-1790
Mention Star File Number S1257MJ

ACM/SIGSOFT, IEEE-CS Member No _____

Please complete reverse side

Tutorials

Saturday, May 18, 1997

SSR HALF DAY TUTORIALS AM

- ☐ (SSR-T1) Generative Reuse: A Survey of Tools and Processes
- ☐ (SSR-T2) Designing High-Performance Reusable Code
- ☐ (SSR-T3) Extending the Software Process to Include Reuse
- ☐ (SSR-T4) Requirements-driven Software Reusability

PM

- ☐ (SSR-T5) Software System Generators, Architectures, and Reuse
- ☐ (SSR-T6) Design Dilemmas That Impede Construction of High-Quality Components
- ☐ (SSR-T7) Software Reuse Metrics, Reusability Metrics, and Economic Models
- ☐ (SSR-T8) Software Reuse with Java

Sunday, May 18, 1997

FULL DAY TUTORIALS

- ☐ (1A) Software Process Improvement: Methods and Lessons Learned
- ☐ (1B) A Realistic, Commercially Robust Process for the Development of Object-Oriented Software Systems
- ☐ (1C) Software-Reliability-Engineered Testing (SRET) Practice
- ☐ (1D) Java: A Language for Software Engineering

HALF DAY TUTORIALS

(You may select two at Full-Day rate)

AM

- ☐ (2A) Software Interoperability: Principles and Practice
- ☐ (2B) Distributed Software Architectures
- ☐ (2C) Effective Use of COTS Software in Long-Lived Systems

PM

- ☐ (3A) Rigorous Requirements for Real-Time Systems Evolution and Application of the SCR Method
- ☐ (3B) Software and Business Process Technology
- ☐ (3C) An Introduction to OMG/CORBA

Monday, May 19, 1997

FULL DAY TUTORIALS

- ☐ (4A) The Experience Factory: How to Build and Run One
- ☐ (4B) The Personal Process in Software Engineering
- ☐ (4C) Making Requirements Measurable
- ☐ (4D) Defining Families: The Commonality Analysis
- ☐ (4E) Evaluating Software Technology

HALF DAY TUTORIALS

(You may select two at Full-Day rate)

AM

- ☐ (5A) A Survey of Object-Oriented Analysis and Design Methods
- ☐ (5B) Simplifying the Evolution of Java Programs
- ☐ (5C) A Primer on Empirical Studies
- ☐ (5D) Reverse Engineering Strategies for Software Migration
- ☐ (5E) Tailoring the CMM for Small Business, Small Organizations and Small Projects

PM

- ☐ (6A) Framework Patterns
- ☐ (6B) Formal Methods for Broadband and Multimedia Systems
- ☐ (6C) Software Engineering Data Analysis Techniques
- ☐ (6D) Comprehension and Evolution of Legacy Software
- ☐ (6E) European and American Software Process Maturity Models and Assessments

Workshops (by Invitation only)

You need to receive an invitation letter from the conference chair/organizers to register for the following workshops. Please see pages 4-5 of the Advance Program for information.

Saturday, May 17

- ☐ Migration Strategies for Legacy Systems
- ☐ Living With Inconsistency
- ☐ 2nd ISEW Cleanroom Workshop

Sunday, May 18

- ☐ Process Modeling and Empirical Studies of Software Evolution

Monday, May 19

- ☐ Software Engineering on the World Wide Web
- ☐ Doctoral Consortium

Friday, May 23

- ☐ Software Engineering Education

Co-Located Symposia

You may register for these on the ICSE Registration form. Please see pages 4-5 of this Advance Program for information.

Saturday, May 17-Sunday, May 18

- ☐ 2nd IFIP International Workshop on Software Engineering for Parallel and Distributed Systems (PDSE-97)

Sunday, May 18-Monday, May 19

- ☐ 7th Int. Workshop on Software Configuration Management (SCM-7)
- ☐ 1997 Symposium on Software Reuse (SSR)

Special Needs

Do you have any special needs? Please specify:

Accommodations

Sheraton Boston Hotel & Towers, 39 Dalton Street, Boston, MA 02199

Tel: +1-800-325-3535 or +1-617-236-2000 • Fax: +1-617-236-6061

Cancellations will be accepted up until 48 hours prior to arrival.

Single (one person): \$160 • Double (two persons): \$175 • Government: \$105

To receive these rates, you must inform the hotel that you are attending the ICSE Conference. These rates good until April 18, 1997

To obtain discounted airfares, call American Airlines, +1-800-433-1790
Mention Star File Number S1257MJ

OPERATIONS

CONFERENCE MANAGEMENT

Elisa Van Dam
Monica Stevenson
Nth Degree, Inc.

EXHIBITS

Susan Hines
Nth Degree, Inc.

LOCAL ARRANGEMENTS

Robert Morris
University of Massachusetts, Boston, USA

WWW COORDINATOR

Roy Fielding
University of California, Irvine

PUBLICITY LIAISON

Debra A. Brodbeck
University of California, Irvine

PROCEEDINGS

Kari Nies
University of California, Irvine

REGISTRATION

Registration Department
Nth Degree

STUDENT VOLUNTEERS

Kevin J. Sullivan
University of Virginia
Gregory Abowd
Georgia Tech University

CONFERENCE NEWSLETTER

Will Tracz
Loral Federal Systems

INTERNATIONAL RELATIONS

AMERICAS

Will Tracz
Loral Federal Systems, USA

CANADA

Nazim Madhavji
McGill University, Canada

ASIA

Koji Torii
Nara Institute of Technology, Japan

AUSTRALIA/NEW ZEALAND

Chris Marlin
Flinders University, Australia

EUROPE

Michel Lemoine
CERT, France

CONFERENCE COMMITTEE

GENERAL ARRANGEMENTS CHAIR

W. Richards Adrion
University of
Massachusetts
Lederle GRC
Box 34610
Amherst, MA 01003
USA
adrion@cs.umass.edu
<http://www.cs.umass.edu/~adrion>
+1-413-545-2475
+1-413-545-3729 (fax)

PROGRAM COORDINATORS

Alfonso Fuggetta
Dipartimento di
Elettronica e
Informazione
Politecnico di Milano
Pza Leonardo da Vinci,
32
20133 Milano, Italy
fuggetta@elet.polimi.it
+39-2-2399-3540
+39-2-2399-3411 (fax)
<http://www.elet.polimi.it/~fuggetta>

Richard N. Taylor
Information and
Computer Science
University of

California, Irvine
Irvine, CA 92717-3425
USA
taylor@ics.uci.edu
+1-714-824-6429
+1-714-824-4056 (fax)
<http://www.ics.uci.edu/~taylor>

Anthony I. Wasserman
Software Methods &
Tools
176 Gold Mine Dr.
San Francisco, CA
94131
tonyw@methods-tools.com
<http://www.methods-tools.com>

PROGRAM COMMITTEE (PAPERS)

Guillermo Arango
(USA)
David Barstow (USA)
Don Batory (USA)
Marty Cagan (USA)
Aniello Cimitile (Italy)
Bruce Croft (USA)
Michael Cusumano
(USA)
Jean-Claude Derniame
(France)

Laurie Dillon (USA)
David Eichmann (USA)
Jacky Estublier
(France)
Herm Fischer (USA)
Kokichi Futatsugi
(Japan)
John Gannon (USA)
David Garlan (USA)
Marie-Claude Gaudel
(France)
Sol Greenspan (USA)
William Griswold
(USA)
Jonathan Grudin
(USA)
Mehdi Harandi (USA)
Mary Jean Harrold
(USA)
Ellen Isaacs (USA)
Paola Inverardi (Italy)
Matthias Jarke
(Germany)
Ross Jeffery (Australia)
Takuya Katayama
(Japan)
Richard Kemmerer
(USA)
Jeffrey Kramer (UK)
Mark Linton (USA)
Bev Littlewood (UK)
Tom Maibaum (UK)
John McHugh (USA)
Mark Moriconi (USA)

John Mylopoulos
(Canada)
Oscar Nierstrasz
(Switzerland)
David Notkin (USA)
Dewayne Perry (USA)
Mauro Pezzé (Italy)
Adam Porter (USA)
David Rosenblum
(USA)
Will Tracz (USA)
Axel van Lamsweerde
(Belgium)
Lawrence Votta (USA)
Bruce Waddington
(USA)
Brian Warboys (UK)
Martin Wirsing
(Germany)
Alexander Wolf (USA)

RESEARCH DEMONSTRATIONS

Anne Duncan
Digital Equipment
Corporation
Volker Gruhn
LION GmbH
Barbara Lerner
University of
Massachusetts,
Amherst

SE IN ORGS: LESSONS AND STATUS REPORTS

Elliot J. Chikofsky
DMR Group
Ian Thomas
Ian Thomas and
Associates

PANELS

Anthony I. Wasserman
Richard Selby,
University of
California, Irvine
(USA)

POSTERS

Debra J. Richardson
University of
California, Irvine

WORKSHOPS

Reino Kurki-Suonio
Tampere University of
Technology, Finland

Lawrence Votta
AT&T Bell
Laboratories, USA

DOCTORAL CONSORTIUM

Michal Young
Purdue University,
USA

About ACM

ACM, the First Society in Computing, is a major force in advancing the skills and knowledge of Information Technology professionals and students throughout the world.

About ACM SIGSOFT

ACM SIGSOFT focuses on issues related to all aspects of software development and maintenance. To that end, SIGSOFT provides a forum for computing professionals from industry, government, and academia to examine principles, practices, and new research results.

About IEEE Computer Society

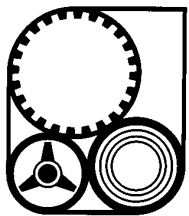
Celebrating its 50th anniversary in 1996, the IEEE Computer Society is the oldest and largest association of computer professionals in the world. (<http://www.computer.org>)

About IEEE Technical Council on Software Engineering (TCSE)

The Technical Council on Software Engineering (TCSE) is the IEEE Computer Society's coordinating body for innovative programs and services in software engineering. TCSE is at the forefront of information exchange and support for both practitioners and researchers throughout the software engineering field. (<http://www.tcse.org>)

Cooperating Societies

Council of European Professional Informatics Societies (CEPIS)



ICSE 97

International Conference
on Software Engineering

c/o Nth Degree
490 Boston Post Road
Sudbury, MA 01776 USA

ABOUT BOSTON

When you visit Boston, the birthplace of the American Revolution, you will be steps from world-class shopping and dining in the cosmopolitan Back Bay. Nearby, you'll also find the flower-lined paths of the Public Garden, the nightclubs and shows of the Theatre District, Red Sox baseball at Fenway Park, and many museums and galleries. The Financial District is minutes away, and the historic Faneuil Hall marketplace abounds with shops and cafés, and the city is home to museums of themes that range from natural history to fine art to computers. The city's rapid transit system is extensive and reasonably priced.

About The Sheraton: The Sheraton Boston (617-236-2000) offers 1,208 guest rooms and suites with personal voice mail. The Executive Level offers guests complimentary continental breakfast and exclusive use of the Executive Lounge. The Sheraton also provides personal valet and laundry services, shoe shine, car rental services, tour information, 24-hour room service, and in-room movies.

Directions to the Sheraton: FROM LOGAN INTERNATIONAL AIRPORT: Take Sumner Tunnel (toll: \$1) to Rt. 93 North. Take Exit 26 (Storrow Drive) and follow for 2 miles. Take Fenway exit (left exit). Stay in the left lane, bearing to the left following "Boylston St. Inbound" signs. You will be on Boylston Street. At fifth light, take right onto Dalton Street. Hotel is on the left. FROM THE WEST: Take Rt. 90 (Mass Pike) East. Take underground Exit 22 (Prudential Center/Copley Square). At fork, follow Prudential Center sign. Above ground, you will be on Huntington Ave. Take right at first set of lights onto Belvidere St. Hotel is on right. FROM THE NORTH: Take Rt. 93 South to Storrow Drive exit. Follow Storrow Drive for 2 miles. Take Fenway exit (left). Stay in the left lane, bearing to the left following "Boylston St. Inbound" signs. You will be on Boylston St. At fifth light, take right to Dalton St. Hotel is on left. FROM THE SOUTH: Take Rt. 93 North. Take left Exit 18 (Mass Ave./Roxbury). Take right at end of off-ramp onto Mass. Ave. Follow for 11/4 miles. Take right to Huntington Ave. At second light, take left to Belvidere St. Hotel is on right.

About the Top of the Hub: The restaurant, on the 52nd floor of the Prudential Center, overlooks the city. It will be the site for the Wednesday evening conference reception.

Web sites with tourism and travel information can be found at <http://www.ics.uci.edu/icse97/boston.html>

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 2
WORCESTER, MA