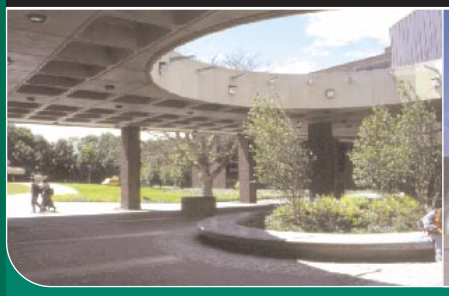
ICSE 2000

LIMERICK, IRELAND 4th - 11th June

THE 22ND INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING

FINAL PROGRAM



SPONSORED BY:

IEEE Computer Society Technical Council on Software Engineering

ACM Special Interest Gruop on Software Engineering

IN COOPERATION WITH:

ACM Special Interest Group on Programming Languages

University of Limerick
Ollscoil Luimnigh

ICSE 2000 CHAIRS & COMMITTEES

General Chair
Carlo Ghezzi (Politecnico di Milano, Italy)
ghezzi@elet.polimi.it

Program
Mehdi Jazayeri (Technical University of Vienna, Austria)
jazayeri@infosys.tuwien.ac.at
Alexander Wolf (University of Colorado at Boulder, USA)
alw@cs.colorado.edu

Conference Organisation
Kevin Ryan (University of Limerick, Ireland)
kevin.ryan@ul.ie

Local Arrangements
Nuala Kitson (University of Limerick, Ireland)
icse2000@ul.ie
Ita Richardson (University of Limerick, Ireland)
ita.richardson@ul.ie

Publicity
Gianpaolo Cugola (Politecnico di Milano, Italy)
cugola@elet.polimi.it

Tutorials
Michal Young (University of Oregon, USA)
michal@cs.uoregon.edu

Workshops
Antonia Bertolino (CNR, Pisa, Italy)
bertolino@iei.pi.cnr.it
Gail Murphy (University of British Columbia, Canada)
murphy@cs.ubc.ca

Future of Software Engineering Reports
Anthony Finkelstein (University College of London, UK)
A.Finkelstein@cs.ucl.ac.uk

Software Engineering Education and Training
Bashar Nuseibeh (Imperial College, UK)
ban@doc.ic.ac.uk

Doctoral Workshop
Jeff Magee (Imperial College, UK)
jnm@doc.ic.ac.uk
Mauro Pezzè (Politecnico di Milano, Italy)
pezze@elet.polimi.it

Education Panels
Nancy Mead (Software Engineering Institute, USA)
nrm@sei.cmu.edu
Hossein Saiedian (University of Nebraska at Omaha, USA)
hossein@cs.unomaha.edu

Teaching Demos
Michael Goedicke (University of Essen, Gerrmany)
goedicke@informatik.uni-essen.de

Practical Experience and Technology Transfer
Hans-Jurgen Kugler (Q-Labs, Ireland)
HansJurgen.Kugler@q-labs.com

Research Demonstrations and Posters
Wolfgang Emmerich (University College of London, UK)
W.Emmerich@cs.ucl.ac.uk
David Rosenblum (University of California, Irvine, USA)
dsr@ics.uci.edu

Proceedings
Harald Gall (Technical University of Vienna, Austria)
gall@infosys.tuwien.ac.at

Electronic Paper Submission
Michael Fischer (Technical University of Vienna, Austria)
fischer@infosys.tuwien.ac.at

Program Committee
K. Anderson (University of Colorado at Boulder, USA)
A. Bertolino (CNR, Pisa, Italy)
L. Briand (Carleton University, Canada)
S.-C. Cheung (Hong Kong University of Science & Tech., China)
J. Cook (New Mexico State University, USA)
P. Devanbu (University of California at Davis, USA)
M. Dwyer (Kansas State University, USA)
W. Emmerich (University College London, UK)
J. Estublier (LSR-IMAG, France)
H. Gall (Technical University of Vienna, Austria)
P. Garg (Hewlett Packard Laboratories, USA)
M. Glinz (University of Zurich, Switzerland)
W. Griswold (University of California at San Diego, USA)
M. J. Harrold (Georgia Tech, USA)
C. Heitmeyer (Naval Research Laboratory, USA)
D. Hoffman (University of Victoria, Canada)
E. Hull (University of Ulster, UK)
P. Inverardi (University of L'Aquila, Italy)
M. Jazayeri (Technical University of Vienna, Austria)
R. Kemmerer (University of California at Santa Barbara, USA)
K. Koskimies (Tampere University of Technology, Finland)
J. Kramer (Imperial College, UK)
G. Leavens (Iowa State University, USA)
D. Le Metayer (IRISA/INRIA Rennes, France)
B. Littlewood (City University London, UK)
W. Melo (Oracle Brazil and Catholic University of Brazil, Brazil)
K. Nakakoji (Nara Institute of Science and Technology, Japan)
O. Nierstrasz (University of Berne, Switzerland)
B. Nuseibeh (Imperial College, UK)
L. Osterweil (University of Massachusetts at Amherst, USA)
J. Palsberg (Purdue University, USA)
M. Pezzè (Politecnico di Milano, Italy)
W. Prinz (GMD FIT, Germany)
A. Ran (Nokia Research Center, USA)
G.-C. Roman (Washington University, USA)
D. Rosenblum (University of California at Irvine, USA)
G. Rothermel (Oregon State University, USA)
K. Ryan (University of Limerick, Ireland)
H. Saiedian (University of Nebraska at Omaha, USA)
M. Saeki (Tokyo Institute of Technology, Japan)
W. Schaefer (University of Paderborn, Germany)
C. Szyperski (Microsoft Research, USA)
R. Taylor (University of California at Irvine, USA)
G. Vigna (University of California at Santa Barbara, USA)
A. Wolf (University of Colorado at Boulder, USA)
D. Yankelevich (University of Buenos Aires, Argentina)

Local Committee:
Mikael Fernstrom
Mary Finucane
Eileen Madden
Breege Moynihan
Kate Sheahan
Dermot Shinners-Kennedy

Conference Management:
Incentive Conference Ireland

Greetings

As we enter the third millennium, a revolutionary transition is under way as the world's industrial society gives way to an information society. Potentially, the information society can create a level playing field by providing equal access to information for all people in the world. The challenge of the third millennium is to realise this potential and to make this dream come true.

Software is at the heart of information technology. Commonly perceived as providing the advanced functions through which we communicate and transform information, software engineering must also address another tier of issues to meet the challenges of the third millennium. Development must not be limited to providing new, advanced functions in a timely fashion, but must also focus on the largely intangible qualities with which such functions are provided. If information technology is to be deployed throughout the world to the benefit of all, key requirements such as usability, dependability, interoperability and affordability must be met.

The ambitious goal of ICSE 2000 is to be the benchmark conference on software engineering for the new millennium. Its goal is to bring together researchers, educators, and practitioners from across the world to mark the starting point of a response to the challenges we all face. We believe that the conference program will satisfy the needs and interests of all the delegates, because of its breadth and depth.

A conference program of this quality could only be put together through the talent and the enthusiastic dedication of volunteer committee members. We also wish to acknowledge the support of our sponsors.

We look forward to your participation in ICSE 2000 and we hope you will share our enthusiasm for this exciting program.

Carlo Ghezzi, General Chair
Mehdi Jazayeri, Program Co-Chair
Alexander L. Wolf, Program Co-Chair

Welcome to Limerick / Fáilte go Luimneach

It is fitting that Ireland should host the most prestigious software engineering conference in the world. The remarkable performance of the Irish economy in the past five years has been due, in no small measure, to the explosive growth in the Irish IT sector. Both multi-national and indigenous software companies have been encouraged by the Irish government to draw on the ready supply of excellent graduates and expand their operations and employment. Many of these companies are among our sponsors, whose generous support helped make this event possible and enjoyable.

The University of Limerick has played a key part in Ireland's economic rebirth. In its short history it has built a reputation for excellent undergraduate education, which is designed to be relevant to Ireland's needs. About 25% of its students are studying IT subjects and it is rapidly expanding its research in key areas of software. The UL campus, while scenic and relaxed, has every modern communication facility required by a state-of-the-art conference. Consequently, I expect that every delegate will have an enjoyable, productive and memorable time at ICSE 2000.

Kevin Ryan, Conference Organisation Chair

Conference Program - An Overview

INTRODUCTION

ICSE 2000 offers a wide range of opportunities for participation to all the members of the community, by providing a coordinated set of events, covering a variety of themes and formats (such as research papers, invited presentations, a series of talks on future trends in software engineering, education papers, panels, technology transfer experience reports, and research demonstrations). In addition, ICSE 2000 offers a Tutorial Program, featuring 3 full day and 17 half day tutorials, a Workshop Program, including 16 workshops, and 3 collocated events: SPICE 2000, PDSE 2000, and IWPC 2000.

TUTORIALS

ICSE 2000 tutorials provide conference participants with the opportunity to gain new insights, knowledge and skills in a broad range of areas in the field of software engineering. Participants at the tutorials include software professionals, teachers, researchers and students both in software engineering and in other areas who seek a better understanding of software engineering topics. The tutorials selected cover a wide range of topics, from practical guidelines, standards and surveys, to emerging technologies and theoretical issues.

WORKSHOPS

ICSE 2000 workshops provide a forum for small groups of participants to exchange opinions, advance ideas, and present preliminary results on focused topics in research and/or applications. Workshops allow participants to closely interact with colleagues on topics of shared interest. Some workshops have open participation, others are by invitation only, based on submitted papers or position statements. The program includes 16 workshops, balanced between a number based on novel but promising ideas, and others strongly continuing the work started in previous ICSEs.

INVITED PRESENTATIONS

ICSE 2000 has four invited speakers who will give presentations in plenary sessions. Manuel Castells, from the University of California at Berkeley, will talk about the relationship between information technology, the new economy that emerged at the turn of the millennium, and social sustainability. Grady Booch, from Catapulse, will present his view of the evolution of software and how it will meet the challenges of the new economy. Chris Horn, from IONA Technologies, will provide further insights into the new economy, by discussing Internet-based versus traditional business, and how portal technology affects this scenario. Axel Van Lamsweerde, from the University of Louvain, will provide a research perspective on the critical field of requirements engineering.

TECHNICAL PAPERS

ICSE 2000 technical program features 49 refereed papers organised in twenty-two sessions. The papers report on the latest research results in a wide variety of software engineering areas. All major areas of software engineering are covered, with sessions devoted to software architecture, web-based systems, component-based systems, open source, real-time systems, empirical studies, testing, case studies, formal methods, requirements engineering, visual techniques, inspections and reviews.

FUTURE OF SOFTWARE ENGINEERING

ICSE 2000 features a “Future of Software Engineering” track, which provides delegates with a unique opportunity to assess the current status of software engineering and to indicate where the field is heading in the future. To celebrate the turn of the century, an international group of leading experts has been invited to report on different topics, to provide a broad and in-depth view of the evolution of the field.

PRACTICAL EXPERIENCE AND TECHNOLOGY TRANSFER

ICSE 2000 goes beyond being the major conference for research in software engineering by providing a forum for examining issues raised by the transfer of software engineering technology from research to industry, or from one industrial application to another. Papers, invited presentations, and panels on these topics will be presented during the sessions that compose this track.

SOFTWARE ENGINEERING EDUCATION AND TRAINING

ICSE 2000 provides a variety of forums for exchanging research and experiences in software engineering education and training. The program includes technical paper presentations, education panels, a doctoral workshop, and a number of teaching demonstrations.

RESEARCH DEMOS AND POSTERS

ICSE 2000 has a special focus on formal research demonstrations, which provide an opportunity for researchers to exchange in-depth experiences in the development of advanced tools and environments. The conference also features informal research demonstrations and posters, to allow attendees to present their on-going research projects in a more informal setting.

DOCTORAL WORKSHOP

The Doctoral Workshop is a forum for graduate students to present and discuss their dissertation research objectives, approaches, and preliminary results.

It is intended for students who have a specific research proposal and some preliminary results, but with sufficient time prior to thesis completion to benefit from workshop experience. Attendance is by prior application and invitation.

Invited Presentations

WEDNESDAY, 7 JUNE 2000

Is The New Economy Socially Sustainable?
by Manuel Castells, Professor of Sociology, University of California at Berkeley, USA.

At the turn of the millennium, the revolution in information technology has ushered in a new economy. This economy, originated in the United States, and more specifically in the American West Coast, is spreading throughout the world, in an uneven, yet dynamic pattern. It is essentially characterised by the key role of knowledge and information in spurring productivity and enhancing competitiveness; by its global reach; and by its networked form of business organisation. Well managed, this new economy may yield an extraordinary harvest of human creativity and social well being. However, several major contradictions threaten the stability of this new economy: the volatility of global financial markets; the institutional rigidity of business, legislation, and governments in many countries; increasing social inequality and social exclusion throughout the world, limiting market expansion and triggering social tensions; and the growing opposition to globalization without representation on behalf of alternative values, and legitimate concerns on the environmental and social costs of this model of growth. Information technology offers great potential in helping to supersede these contradictions at the dawn of an emerging socio-economic system. But the speed of technological innovation requires the parallel development of institutional and cultural innovation, away from bureaucracy but closer to people, to ensure the sustainability of the new economy, and to spur the new wave of technological creativity.

THURSDAY, 8 JUNE 2000

The Future of Software
by Grady Booch, Chief Technical Officer at Catapulse, USA.

Software is the fuel of the world's new economy. Software has been used to create new markets, heal the human body, explore distant worlds, and bring individuals into community. Software transcends all political boundaries, consumes few resources in its execution, and permits the creation of new worlds with new laws of physics. At its best, software extends the human experience; at its worse, it can amplify our basest faults. And yet, the activity of engineering software falls short of what we would expect to be possible. Software development and deployment remain labour-intensive and intellectually demanding, requiring the best from developers who must play a number of different roles. There is still much friction in the process of crafting complex software; the goal of creating quality software in a repeatable and sustainable manner remains elusive to many organisations, especially those who are driven to develop in Internet time. This problem is exacerbated by the reality that, worldwide, there exists a shortage of skilled developers. In this presentation, we examine the future of software and the future of engineering software. We begin by briefly considering the past and then level-setting where we are in the present, focusing especially on the state of the practice of software development in the world today. We continue with a consideration of the technological, theoretical, economic, and social trends that are shaping the nature of software development. We conclude with a challenge for what software and software engineering can be in a frictionless environment.

THURSDAY, 8 JUNE 2000

Dot.Com Versus Bricks and Mortar – The Impact of Portal Technology by Chris Horn, Founder and CEO of IONA Technologies plc., Ireland.

The "New Economy" is rapidly being adopted on a global scale as corporations vie for new competitive positions and defensive responses. Incumbents - so-called "bricks'n'mortar" corporations - are generally finding it challenging, but usually rewarding, to extend their business practices to the Internet. New entrants - so-called "Dot.Com" companies - are unfettered from institutional rigidity and thus have an enormous opportunity to gain market share, but at the same time are frequently challenged to provide the same levels of brand awareness, product and service as at least some of the incumbents. In this presentation we consider how Internet infrastructure software is evolving, and its implications for both brick'n'mortar and dot com organisations.

FRIDAY, 9 JUNE 2000

Requirements Engineering in the Year 00: A Research Perspective
by Axel van Lamsweerde, Professor of Computer Science, University of Louvain, Belgium.

Requirements engineering (RE) is concerned with the identification of the goals to be achieved by the envisioned system, the operationalization of such goals into services and constraints, and the assignment of responsibilities for the resulting requirements to agents such as humans, devices, and software. The processes involved in RE include elicitation, modelling, specification, analysis, negotiation, documentation, and evolution. Getting high-quality requirements is difficult and critical. Recent surveys have confirmed the growing recognition of RE as an area of utmost importance in software engineering research and practice. The talk will first present a brief history of the main concepts and techniques developed so far to support the RE task. We will then review a number of current research directions in RE-specific areas, such as goal-oriented requirements elaboration, scenario-based elicitation, conflict management, and the handling of abnormal agent behaviours. We will then argue that RE is still at its infancy by providing a research agenda based on weaknesses of current techniques, notably, their limited scope, their difficulty of use, their lack of guidance in the RE process, and their poor connection to architectural design.

Summary Conference Schedule

Sunday June 4

09:00

W01 day 1
ISAW - 4

W02 day 1
Web Engineering

W03 day 1
Automated Analysis

W04 day 1
COTS

12:30

Lunch

14:00

workshops continued from morning

17:30

18:00

Monday June 5

W01 day 2
ISAW - 4

W02 day 2
Web Engineering

W03 day 2
Automated Analysis

W04 day 2
COTS

W05
Empirical SE Research

W06
COSET 2000

W08 day 1
DSV - IS 2000

Lunch

W09 day 1
CBSE

& workshops continued from morning

T01
COCOMO II

T02
Quality in Use

T03
Architectures using ABASs

T04
Reusable Collaborations

T05
CORBA Introduction

T06
ISO9000 to CMM

Lunch

T01 Continued

T02 Continued

T07
Scheduling using Architecture

T08
Aspect J

T09
CORBA Scalability

T10
Software Patents and Copyrights

Tuesday June 6

W07
SEWPC

W08 day 2
DSV - IS 2000

W09 day 2
CBSE

W10
WOSEF

W11
SE over Internet

W12
Separation of Concerns

W13
EDSER -2

W14
WISE 3

Lunch

workshops continued from morning

Welcome Reception
Atrium, Foundation Building

T11
SPI Best Practice

T12
Commonality Analysis

T13
RT and Distribution UML

T14
Using Application Services

T15
Software Reliability

Lunch

T11 Continued

T16
Product - Line Architectures

T17
Advanced Visual Modelling

T18
Code Mobility

T19
Fault Tolerance

T20
Software Inspections

Doctoral Workshop

Saturday June 10

W15
Software Product Lines

W16
Agent - Oriented SE

Lunch

workshops continued from morning

PDSE2000

IWPC2000

SPICE2000

Sunday June 11

PDSE2000

IWPC2000

SPICE2000

Main Conference Schedule (Wednesday)

TP: Technical Papers

SEAT: Software Engineering Education and Training

RD: Research Demos

PETT: Practical Experience and Technology Transfer

FoSE: Future of Software Engineering

Wednesday June 7

09:00

S1 Opening Ceremony + Invited Presentation by M. Castells
FG061 Concert Hall

10:45

Break

11:15

S2.1 TP Components & COTS ***FG061***	**S2.2 TP** Software Architectures and Product Families ***DG016***	**S2.3 FoSE** Reasoning and Representation ***CSG01***	**S2.4 PETT** Technology Transfer "in the large" ***FG042***	**S2.5 TP** Panel: Impact of Software Engineering Research upon Practice ***FB028***

12:45

Lunch

14:15

S3 Awards Presentation
FG061 Concert Hall

15:15

S4.1 TP New Perspectives on Software Engineering ***FG061***	**S4.2 TP** Data Analysis ***DG016***	**S4.3 FoSE** Infrastructure I ***CSG01***	**S4.4 PETT** Professionalization of Software Engineering ***FG042***	**S4.5 SEAT** Teaching Demos ***FB028***

15:30

16:30

Break

17:00

S5.1 TP Testing I ***FG061***	**S5.2 TP** Evolution and Reuse ***DG016***	**S5.3 FoSE** Process I ***CSG01***	**S5.4 PETT** Panel: Component-Based Software Engineering and the Issue of Trust ***FG042***	**S5.5 SEAT** Panel: Shortages of Qualified Software Engineering Faculty and Practitioners ***FB028***

18:30

19:30

BBQ Stables Courtyard U.L.

Main Conference Schedule (Thursday)

TP: Technical Papers

SEAT: Software Engineering Education and Training

RD: Research Demos

PETT: Practical Experience and Technology Transfer

FoSE: Future of Software Engineering

Thursday June 8

09:00	**S6** Invited Presentation by G. Booch ***FG061 Concert Hall***				
10:00					
10:15	**S7.1 TP** Component-Based Systems ***FG061***	**S7.2 TP-SEAT** Software Engineering Training ***DG016***	**S7.3 FoSE** Methods I ***CSG01***	**S7.4 PETT** Experience with a Product Line and Family Approach ***FG042***	**S7.5 RD** Research Demos ***FB028***
11:15	Break				
11:45	**S8.1 TP** Testing II ***FG061***	**S8.2 TP** Software Architecture ***DG016***	**S8.3 FoSE** Methods II ***CSG01***	**S8.4 PETT** Technology Transfer as an Explicit Business and Process Issue ***FG042***	**S8.5 SEAT** Teaching Demos ***FB028***
12:45	Lunch				
14:15	**S9** Invited Presentation by C. Horn ***FG061 Concert Hall***				
15:15					
15:30	**S10.1 TP** Open Source and Software Markets ***FG061***	**S10.2 SEAT** Software Engineering Education ***DG016***	**S10.3 FoSE** Process II ***CSG01***	**S10.4 PETT** Support for Effective Project Estimation ***FG042***	**S10.5 RD** Research Demos ***FB028***
16:30	Break				
17:00	**S11.1 TP** System Model Derivation ***FG061***	**S11.2 TP** Model Checking ***DG016***	**S11.3 FoSE** Components and Structure ***CSG01***	**S11.4 PETT** Technology Transfer in the Internet World ***FG042***	**S11.5 SEAT** Panel: Who needs Doctors? ***FB028***
18:30	**Bunratty Folk Park and Conference Banquet**				

Main Conference Schedule (Friday)

TP: Technical Papers

SEAT: Software Engineering Education and Training

RD: Research Demos

PETT: Practical Experience and Technology Transfer

FoSE: Future of Software Engineering

Friday June 9

09:00

S12 Invited Presentation by A. van Lamsweerde
FG061 Concert Hall

10:00

10:15

S13.1 TP Program Analysis I ***FG061***	**S13.2 TP** Empirical Studies ***DG016***	**S13.3 FoSE** Properties I ***CSG01***	**S13.4 PETT** From Research to Business Success ***FG042***	**S13.5 RD** Research Demos ***FB028***

11:15

Break

11:45

S14.1 TP Web-Based Systems ***FG061***	**S14.2 TP** Case Studies ***DG016***	**S14.3 FoSE** Properties II ***CSG01***	**S14.4 PETT** Practical Experience: Company Case Studies I ***FG042***	**S14.5 SEAT** Teaching Demos ***FB028***

12:45

Lunch

14:15

S15.1 TP Program Analysis II ***FG061***	**S15.2 TP** Review and Inspection Techniques ***DG016***	**S15.3 FoSE** Properties III ***CSG01***	**S15.4 PETT** Practical Experience: Company Case Studies II ***FG042***	**S15.5 RD** Research Demos ***FB028***

15:15

15:30

S16.1 TP Verification and Proofs ***FG061***	**S16.2 TP** Visual Techniques ***DG016***	**S16.3 FoSE** Infrastructure II ***CSG01***		**S16.5 RD** Research Demos ***FB028***

16:30

16:45

S17 ICSE Summary and Prospects
FG061 Concert Hall

17:15

Tutorial Program

Rooms are colour coded according to map on back of program.

Monday, 5 June 2000

T01: SOFTWARE COST ESTIMATING WITH THE COCOMO II MODEL
Donald J. Reifer, University of Southern California
Monday, 5 June 2000, 9:00 - 17:30, C1058

This one-day tutorial introduces participants to the subject of software economics. Emphasis is placed on developing software cost estimating skills using the newly released COCOMO II-2000 model. Use of the model for developing estimates, performing what-if analysis, bounding risks, and conducting a variety of trade studies is also highlighted. The seminar uses a variety of examples, case studies, and hands-on exercises to develop estimating skills, knowledge and abilities. It generates example cost model runs using both the USC and COSTAR packages.

T02: SPECIFYING AND MEASURING QUALITY IN USE
Nigel Bevan, Serco Usability Services
Monday, 5 June 2000, 9:00 - 17:30, C1056

The tutorial will explain how quality in use can be operationalized as the prime quality goal for a system. Quality in use is the extent to which a system meets end user needs. The tutorial is based on a methodology originally developed by the collaborative European ESPRIT MUSiC (Measurement of Usability in Context) project. This methodology implements the principles of ISO 9126-4 (Quality in Use Metrics). Participants will learn how to use the US Common Industry Format to specify quality in use requirements, and to report the results of a quality in use test.

T03: DESIGNING AND ANALYSING SOFTWARE ARCHITECTURES USING ABASs
Rick Kazman, Software Engineering Institute, CMU
Monday, 5 June 2000, 9:00 - 12:30, C1059

This tutorial will discuss, exemplify, and involve the participants in the use of Attribute-Based Architectural Styles (ABASs) architectural styles accompanied by explicit analysis reasoning frameworks in both the design and analysis of software and system architectures. The tutorial has several objectives: to introduce the participants to a catalog of ABASs covering performance, availability, security, modifiability, and usability; to convince participants that ABASs provide a basis for insightful reasoning about a software architecture's ability to meet its quality attribute goals; to demonstrate the utility of ABASs by showing examples of how ABASs are used to design and analyse real-world system architectures; and to give the participants some experience with using ABASs directly. We will present some large excerpts from our growing ABAS handbook and show that ABASs help us in designing architectures efficiently and predictably and in quickly finding architectural risks and tradeoffs when doing analysis.

T04:BUILDING MODULAR OO SYSTEMS WITH REUSABLE COLLABORATIONS
Karl Lieberherr, Northeastern University and UBS AG
Monday, 5 June 2000, 9:00 - 12:30, C1061

Collaborations are a practical approach to avoiding tangling of concerns that should be separated in OO development. One way to develop a system is to start with use cases, realise them as collaborations, code them as collaborating classes and deploy them in an application. This tutorial shows participants how to use collaborations and describes tools to facilitate their use in practice. Participants will learn how to use collaborations to address some important problems in their specification, design and implementation activities. They will also learn how to use collaborations to facilitate the use of design patterns and aspect-oriented designs and programs.

T05: INTRODUCTION TO CORBA
Steve Vinoski, IONA Technologies
Monday, 5 June 2000, 9:00 - 12:30, EM009

This tutorial provides the basics that developers need to begin understanding CORBA and using it to write industrial-strength distributed systems. You will learn about the basics of the OMG's Object Management Architecture (OMA), with a focus on its CORBA component. By the end of the tutorial, you will understand how to write object interface specifications using the OMG Interface Definition Language (IDL), how to write simple distributed applications in C++, how to use the new Portable Object Adaptor (POA), the Dynamic Invocation Interface (DII) and the Dynamic Skeleton Interface (DSI), and the Interface Repository (IFR). You will also know the basics of several CORBA services such as Naming, Trading, and Events.

T06: MOVING FROM ISO9000 TO HIGH MATURITY LEVELS OF THE CMM
Pankaj Jalote, I.I.T. Kanpur
Monday, 5 June 2000, 9:00 - 12:30, EM010

There are a large number of software development organisations in the world that are ISO 9001 certified. Many of these are now considering adopting the SEI's Capability Maturity Model (CMM), which provides a basis for process management and improvement. In this transition from ISO 9001 to CMM, processes have to be enhanced to suit the CMM (while preserving ISO 9001). This tutorial discusses the typical enhancements that might be needed by an ISO organisation when moving to higher levels of maturity in the CMM framework, how to leverage ISO structures and practices for implementing CMM, and how to manage the transition. This tutorial is based on the experience of the author in spearheading the transition from ISO 9000 to CMM of Infosys, a large Bangalore-based software organisation, and in helping some other organisations in transitioning. It is intended for members of those organisations that are ISO-certified, or are going for ISO certification, and who wish to consider moving to the CMM. It will also be of interest to professionals interested in these models.

T07: PLANNING REALISTIC SCHEDULES USING SOFTWARE ARCHITECTURE
Robert L. Nord, Siemens Corporate Research
Daniel J. Paulish, Siemens Corporate Research
Dilip Soni, Siemens Corporate Research
Monday, 5 June 2000, 14:00 - 17:30, C1059

Software architecture forms the organisational plan for development. This tutorial introduces the concepts of architecture design and shows how they can be used in software project planning to better plan and manage software projects. By applying the architecture-centred approach described in this tutorial, we believe that projects can be planned within a schedule estimation error of 15-20%. A software architecture design document is the primary input to the top-down and bottom-up planning processes, and a software development plan is the primary output. Each development team member uses the software development plan for generating a personal schedule with weekly milestones for the components that they are developing. In addition, there is an overall project schedule monitored by the project manager that identifies how the components are allocated to the incremental releases. The approach is best applied when the high-level design is complete and the development team is staffed. It helps transfer the design knowledge to the team members and builds buy-in of the schedule. As a participant in this tutorial, you will learn how architecture-centred software project planning can be used to help plan and manage software projects. This approach has been applied to several projects and we will share our experience on how this works in practice. We will work through an example based on a real system and ask you to participate in a couple of exercises. You will also learn about the critical roles of the project manager, software architect, and team members in making this process successful.

T08: IMPROVING DESIGN AND SOURCE CODE MODULARITY USING ASPECTJ™
Cristina Lopes, Xerox PARC
Gregor Kiczales, University of British Columbia,
Monday, 5 June 2000, 14:00 - 17:30, C1061

Aspect-oriented programming (AOP) can be used to improve the modularity of a wide range of software. Using only traditional techniques, the implementation of concerns like exception handling, multi-object protocols, synchronization constraints, and security policies tends to be spread out in the code. The lack of modularity for these concerns makes them more difficult to develop and maintain. This tutorial shows how to use AOP to implement concerns like these in a concise, modular way. We will discuss the effect aspects have on software design and on code modularity. The concrete examples in the tutorial use AspectJ, a freely available aspect-oriented extension to the Java™ programming language.

Tutorial Program (cont.)

T09: SCALABILITY ISSUES IN CORBA-BASED SYSTEMS
Steve Vinoski, IONA Technologies
Monday, 5 June 2000, 14:00 - 17:30, EM009

This tutorial addresses how both the OMG specifications and the implementation choices made by middleware providers and application developers affect application scalability. We will cover a range of scalability issues, starting with ORB internals and working outward to full-scale applications, addressing issues such as connection management, POA scalability features, multithreading, object lifecycle issues, object location, system configuration, maintenance, and management, and common application architectures. This tutorial is not language-centric and is useful to developers using Java, C++, or any other language to develop CORBA-based applications.

T10: INTELLECTUAL PROPERTY PROTECTION FOR SOFTWARE IN THE UNITED STATES AND EUROPE: THE CHANGING ROLES OF PATENTS AND COPYRIGHTS
Gregory J. Kirsch, Needle & Rosenberg, P.C.
Yannis Skulikaris, European Patent Office
Monday, 5 June 2000, 14:00 - 17:30, EM010

Intellectual Property (IP) is expected to play an increasingly important role in the coming years. IP laws evolve not just on a national or regional basis, but also globally, influenced by the growing importance of technology and international trade. Moreover, certain IP laws and specifically patent law have been steadily expanding to cover emerging technologies, such as computer software, and more recently automated business methods. At the same time, the legal protection afforded by other types of IP laws, namely copyright protection, has been narrowed in recent years, resulting in a dramatic shift in how software developers are seeking to protect their innovation. Specifically, there has been an explosive shift to patents as a primary mechanism for legally protecting software and Internet-related inventions. This tutorial will shed light regarding the important changes that are occurring in the IP laws in the United States and Europe, and how companies and other institutions are taking advantage of and responding to these changes.

Tuesday, 6 June 2000

T11: SOFTWARE PROCESS IMPROVEMENT: BEST PRACTICES AND LESSONS LEARNED
Bill Curtis, TeraQuest Metrics
Tuesday, 6 June 2000, 9:00 - 17:30, C1060

This tutorial will provide an excellent opportunity for anyone wishing to learn more about how to improve their software operations. The tutorial will begin by discussing the problems in software development that are solved by SPI program, and will later describe problems whose solutions are enabled by SPI, even though the solutions must emerge from technical activities. The tutorial will describe the results achieved in successful SPI programs over the last decade, but will focus heavily on results emerging in the last couple of years from companies like Ericsson, Boeing, Tata, Lockheed Martin, and Telcordia Technologies. A brief comparative description of SPI models such as CMM and SPICE will be provided. The tutorial will then review the critical success and failure factors that have emerged from SPI programs over the last decade. In particular, 10 known failure conditions SPI programs frequently experience will be described. Responsibilities in SPI programs for executives, managers, process groups, and developers will be described. A new paradigm for conducting SPI programs that has been designed to avoid many of the pitfalls will be described. SPI's relevance to internet software and outsourcing will be discussed. Participants will be given frequent opportunities to question and interact with the instructor.

T12: COMMONALITY ANALYSIS FOR DEFINING SOFTWARE PRODUCT LINES
Mark Ardis, Bell Laboratories, Lucent Technologies
Tuesday, 6 June 2000, 9:00 - 12:30, EM009

A recent trend in both the software engineering research and industrial communities has been to seek ways to systematically engineer software domains. Rather than design and construct software products individually, a product line is designed that allows variation between products to meet individual requirements. This tutorial teaches the commonality analysis process, a systematic approach to analysing families of software systems. The result of the analysis forms the basis for designing reusable assets that facilitate rapid production of new products within the software product line. Participants will learn the principles underlying the approach and will perform a practice commonality analysis guided by an experienced user of the process.

T13: DESIGNING REAL-TIME AND DISTRIBUTED APPLICATIONS WITH THE UML
Hassan Gomaa, George Mason University
Tuesday, 6 June 2000, 9:00 - 12:30, EM010

This tutorial describes the object-oriented analysis and design of concurrent applications, in particular real-time, client/server, and distributed applications. Object-oriented concepts are crucial in software analysis and design because they address fundamental issues of adaptation and evolution. With the proliferation of notations and methods for the object-oriented analysis and design of software systems, the Unified Modelling Language (UML) has emerged to provide a standardised notation for describing object-oriented models. However, for the UML notation to be effectively applied, it needs to be used in conjunction with an object-oriented analysis and design method. Most books and courses on object-oriented analysis and design only address the design of sequential systems or omit the important design issues that need to be addressed when designing real-time and distributed applications. It is essential to blend object-oriented concepts with concurrency concepts to successfully design these applications. As the UML is now the standardised notation for describing object-oriented models, this tutorial uses the UML notation throughout.

T14: SYSTEM DEVELOPMENT USING APPLICATION SERVICES OVER THE NET
Kenji Takahashi, NTT
Anthony Finkelstein, University College, London
Wolfgang Emmerich, University College, London
Sofia Guerra, University College, London
Tuesday, 6 June 2000, 9:00 - 12:30, C1056

This tutorial explores how application service provision (ASP) is changing the software development landscape in terms of technology, methodology and economics. ASP is a type of commercial service offering that allows users to use computer applications hosted by the service providers at their premises over the network. The promise of ASP is that by using application services over the Net (internet, intranet, and/or extranet), a significant part of system requirements can be met with, it is suggested, little need for initial development and recurring in-house maintenance effort. The goal of this tutorial is to give participants a detailed understanding of the prospects for, and issues arising from, the emerging ASP industry. In this tutorial, we give an overview of the ASP industry and illustrate how ASP works in real settings using case studies. We discuss methods to make effective use of ASP (e.g. specification and selection of services), examine enabling technology (e.g. technical standards), and explore the key challenges (e.g. security and performance).

T15: SOFTWARE RELIABILITY: BASIC CONCEPTS AND ASSESSMENT METHODS
Bev Littlewood, Centre for Software Reliability, City Univ., London
Lorenzo Strigini, Centre for Software Reliability, City Univ., London
Tuesday, 6 June 2000, 9:00 - 12:30, C1062

This half-day tutorial introduces the need for probabilistic measures of software dependability, and the basic principles, capabilities and limitations of the methods for assessing and predicting them. The topics covered include the basic concepts of reliability; principles of statistically realistic testing; estimation of stable reliability; reliability growth models and ways for checking their trustworthiness; limits to reliability levels that can be effectively demonstrated before deployment of the software. The tutorial is designed for an audience with a software engineering background, with only a knowledge of the basic concepts of probability. It is meant to give an understanding of the meaning and value of claims about software reliability, and of the practical methods available for predicting it.

Tutorial Program (cont.)

Rooms are colour coded according to map on back of program.

T16: PRODUCT-LINE ARCHITECTURES, ASPECTS, AND REUSE
Don Batory, University of Texas at Austin
Tuesday, 6 June 2000, 14:00 - 17:30, EM009

This half-day tutorial explains practical techniques for designing and building product-line architectures (i.e. families of applications) from components and how to design architecturally evolvable software (i.e. software that evolves through the addition and removal of components). We present a model of product-line architectures that brings together significant areas of research in software design: aspect-oriented programming, Perry's lightweight semantics for validating architectural consistency, parameterized programming, and generative programming. Among the practical case-studies we review are building extensible Java compilers (i.e. a product-line of Java dialects) and extensible simulators for Army fire support. The tutorial assumes rudimentary knowledge of OO concepts, but no experience with the tutorial's subject.

T17: ADVANCED VISUAL MODELLING: BEYOND UML
Joseph Gil, Technion, The University, Canterbury
John Howse, University of Brighton
Stuart Kent, The University, Canterbury
Tuesday, 6 June 2000, 14:00 - 17:30, EM010

With the adoption of UML by the OMG and industry as the standard visual modelling notation, it is interesting to wonder what will come next in this field. This tutorial presents a vision for visual modelling beyond UML. It is based on a series of recent research papers, which have introduced some radical new notations and which have suggested the kinds of tools that could be available to support the modeler of the future. Highlights include: a crash critical overview in UML, stressing its weaknesses and strengths; a rich visual constraint language and an insight into subtle issues that arise when defining a visual language for applying the popular design-by-contract using a visual formalism; and a demonstration of a graphical editor for the constraint-diagrams language.

T18: UNDERSTANDING CODE MOBILITY
Gian Pietro Picco, Politecnico di Milano
Tuesday, 6 June 2000, 14:00 - 17:30, C1056

Code mobility allows a distributed application to relocate its components at run time. This approach, made popular by Java and a myriad of other languages and systems, exhibits the potential for changing radically the way distributed applications are developed and deployed. Nevertheless, the research area is still quite immature, and there is a strong need for a systematic approach towards understanding the key characteristics of code mobility as well as for a careful analysis of the benefits provided. The tutorial will provide a conceptual framework for code mobility by illustrating a taxonomy of mobile code technologies, architectural paradigms, and applications. The taxonomy will provide a terminological basis, as well as a precise characterisation of the founding concepts of the research area. As a final case study, the concepts developed in the taxonomy will be applied to a quantitative assessment of the benefits of mobile code technologies and paradigms in the network management application domain.

T19: FAULT TOLERANCE VIA DIVERSITY AGAINST DESIGN FAULTS: DESIGN PRINCIPLES AND RELIABILITY ASSESSMENT
Bev Littlewood, Centre for Software Reliability, City Univ., London
Lorenzo Strigini, Centre for Software Reliability, City Univ., London
Tuesday, 6 June 2000, 14:00 - 17:30, C1062

Design faults account for a large part of failures in mature software-based products. Fault tolerance, employing redundant, diverse software components, has been used for many years as a defence. Though commonly associated with safety-critical real-time systems, its principles are applicable to all kinds of software designs. While the idea is attractive, the use of diversity has been vehemently attacked. Most practitioners for whom software fault tolerance could be a viable choice have little information for decisions about its use. This half-day tutorial introduces both the methods of fault-tolerant software design, with examples from industrial practice and from research, and the problem of estimating the reliability gain from fault-tolerant design. This estimation is as difficult as for other software engineering methods. The tutorial explains the basic results so far, clarifies the terms of decisions about using software fault tolerance, and outlines recent progress in research. The tutorial is designed as an introduction to this topic for an audience with an experience in software engineering and an understanding of the basic concepts of software reliability.

T20: IMPROVING SOFTWARE INSPECTIONS BY USING READING TECHNIQUES
Victor Basili and Forrest Shul, Fraunhofer Center for Experimental Software Engineering, Maryland
Ioana Rus and Oliver Laitenberger, Fraunhofer Institute for Experimental Software Engineering (IESE), Germany
Tuesday, 6 June 2000, 14:00 - 17:30, D1049

This tutorial introduces Perspective-Based Reading (PBR), a specific reading technique used to review software requirements documents. The specific goals of this tutorial are: to describe a set of techniques that can increase the effectiveness of software inspections by providing individual inspectors with systematic techniques to read a software artifact and recognise defects; to provide participants with the opportunity to apply PBR while being able to ask questions or ask for assistance from the instructors; to demonstrate to the participants how to tailor the PBR technique to their development environment; and to discuss how participants can assess inspections in their own environments. This tutorial aims at industry practitioners, managers and developers alike, who want to learn more about ways to improve their software inspections with systematic reading techniques. Attending this tutorial will enable the participants to be more effective and more focused in looking for potential defects in software documents.

Workshop Program

Sunday - Monday, 4 - 5 June 2000

W01: 4TH INTERNATIONAL SOFTWARE ARCHITECTURE WORKSHOP (ISAW-4)
http://www.extra.research.philips.com/SAE/welcome.html
By invitation only
Sunday, 4 June 2000, 9:00 - 17:30
Monday, 5 June 2000, 9:00 - 17:30, SG17

The workshop brings together practitioners and researchers for two intense days of discussion and work in the emerging area of Software Architecture. Topics of interest include: Architecture Description Languages, Tools/Environments for Software Architects, the Art and Science of Software Architecting, Domain-Specific Architectures, Product Line Architectures, Dynamic Architectures and Reconfiguration, and Case Studies and Experience.

W02: SECOND ICSE WORKSHOP ON WEB ENGINEERING
http://fistserv.macarthur.uws.edu.au/icse2000-WebE
Sunday, 4 June 2000, 9:00 - 17:30
Monday, 5 June 2000, 9:00 - 17:30, S205

This workshop is in response to the increasing need to systematise the current ad hoc approaches to creating and maintaining Web-based applications. It focuses on successful development of large, complex Web-based systems and provides a mix of academic research and experience of industry practitioners to address the major problems in building and maintaining such systems. The workshop builds upon the previous one at ICSE '99 and incorporates the best practices from Software Engineering and other disciplines, which impact upon Web-based application development. It covers processes, methodologies, system design, life-cycle and management of large Web-based systems, and educational and research issues.

W03: WORKSHOP ON AUTOMATED PROGRAM ANALYSIS, TESTING, AND VERIFICATION
http://ase.arc.nasa.gov/icse2000
Sunday, 4 June 2000, 9:00 - 17:30
Monday, 5 June 2000, 9:00 - 17:30, CSG01

Due to recent developments in static analysis, automated testing and automated program verification, the boundaries between these fields have begun to blur. There are many open questions regarding the integration of automated testing and verification, the relationships between different algorithms, and the use of static analysis to assist testing and verification. The goal of this workshop is to bring together active researchers in these fields to foster collaboration. The workshop will span two days. The first day, several invited speakers from each community will give presentations and tool demonstrations, while the second day will consist of focused group discussions.

W04: CONTINUING COLLABORATIONS FOR SUCCESSFUL COTS DEVELOPMENT
http://www.sel.iit.nrc.ca/projects/cots/icse2000wkshp
By invitation only
Sunday, 4 June 2000, 9:00 - 17:30
Monday, 5 June 2000, 9:00 - 12:30, S206

This workshop is intended to be an extension of the work begun during the workshop entitled Ensuring Successful COTS Development, held in conjunction with ICSE '99. As a result of that workshop some collaborative work is being undertaken. One of the goals of this workshop is to refresh those collaborations and to instigate new efforts on a number of open questions that were not undertaken by last year's participants. A major objective is to attract new participants who would be interested in either joining an existing collaboration or establishing relationships in a different COTS research area.

Monday, 5 June 2000

W05: BEG, BORROW OR STEAL: USING MULTI-DISCIPLINARY APPROACHES IN EMPIRICAL SOFTWARE ENGINEERING RESEARCH
http://www.csr.uvic.ca/icse2000
Monday, 5 June 2000, 9:00 - 17:30, CSG28

This interactive workshop is for software engineers wishing to select and adapt methods and theoretical foundations from other fields. The workshop will be centred around real problems that are being experienced by software engineering researchers. Invited position papers will describe software engineering problems that could benefit from advice on empirical methods and theoretical approaches. Experts from other disciplines will participate in the workshop to respond to common problems.

W06: 2ND INTERNATIONAL SYMPOSIUM ON CONSTRUCTING SOFTWARE ENGINEERING TOOLS (COSET 2000)
http://www.itacs.uow.edu.au/~jon/coset2k
Monday, 5 June 2000, 9:00 - 17:30, CS246

Automated tools play an important role in the promotion and adoption of software engineering methods and processes. The development of these tools is itself a significant software engineering task, requiring a considerable investment of time and resources. The different strategies adopted by tool constructors in the development of these tools are the theme of this workshop. The workshop will be based around the participants' experience reports of constructing their tools, and it will focus principally on practical issues of tool development, deployment, and evaluation.

Monday - Tuesday, 5 - 6 June 2000

W08: DESIGN, SPECIFICATION, AND VERIFICATION OF INTERACTIVE SYSTEMS (DSV-IS 2000)
http://giove.cnuce.cnr.it/dsvis2000.html
Monday 5 June, 9:00 - 17:30,
Tuesday, 6 June 2000, 9:00 - 17:30, CSG28

DSV-IS (Design, Specification and Verification of Interactive Systems) is an international workshop that has been organised every year since 1994 in cooperation with Eurographics. As participation from people working in the software engineering field has been rather limited, we organize the 7th edition of DSV-IS in conjunction with ICSE 2000. The workshop will provide a forum for the exchange of ideas on diverse approaches to the design of interactive systems. The particular focus of this year's event is on models (e.g. of devices, users, tasks, context, etc.) and their role in supporting the design, development, and usability evaluation of interactive systems.

W09: 3RD INTERNATIONAL WORKSHOP ON COMPONENT-BASED SOFTWARE ENGINEERING: REFLECTION ON PRACTICE
http://www.sei.cmu.edu/cbs/cbse2000
By invitation only
Monday, 5 June 2000, 14:00 - 17:30, S116
Tuesday, 6 June 2000, 9:00 - 17:30, CSG25

Component-based software engineering (CBSE) is emerging as the software industry's response to dramatic increases in the demand for software and for reducing the time to market for software-intensive systems. As the number of fielded component-based systems grows, it becomes important to reflect upon what is working and where improvements are needed in practice. The third ICSE Workshop on CBSE will provide a forum for researchers and practitioners to present case studies in CBSE. The case studies will focus on the technologies and engineering practices that are unique to CBSE, and their evaluation will highlight best practices in CBSE as well as deficiencies that require engineering research.

Workshop Program (cont.)

Rooms are colour coded according to map on back of program.

Tuesday, 6 June 2000

W07: SOFTWARE ENGINEERING FOR WEARABLE AND PERVASIVE COMPUTING (SEWPC)
http://www.cs.washington.edu/sewpc
By invitation only, Tuesday, 6 June 2000, 9:00 - 17:30, C1061

Pervasive computing embraces a vision of information that is appropriate to the situation and environment, available continuously irrespective of location or circumstances, and connects the virtual and physical worlds allowing them to influence and inform one another. Wearable computing is the intimate apparel of pervasive computing - body-worn sensors, devices, and computing engines that interconnect personal and public information. Soon computing devices will be in the walls and floors of buildings, woven into the fabric of our clothing, and inside ordinary objects such as books or furniture. These deeply embedded, broadly networked, massively scaled systems require an enormous infusion of software whose architecture, algorithms, and languages are largely unknown. The workshop will explore the fundamental challenges that the widespread deployment of wearable and pervasive computing offers software engineering.

W10: WORKSHOP ON STANDARD EXCHANGE FORMAT (WOSEF)
http://www.cs.utoronto.ca/~simsuz/wosef
Tuesday, 6 June 2000, 9:00 - 17:30, C1058

A common exchange format for sharing data extracted from source code is necessary to advance the state of the art of tool development in software engineering. Such a format would allow us to share tools and compare results more easily. There are efforts already under way in some areas of the discipline to develop a standard exchange format. In this workshop, researchers and practitioners from across the discipline can discuss the form of the standard exchange format (i.e. syntax, schema, and semantics) and mechanisms for encouraging adoption of the format.

W11: 3RD WORKSHOP ON SOFTWARE ENGINEERING OVER THE INTERNET
http://sern.cpsc.ucalgary.ca/~maurer/icse2000ws/ICSE2000WS.html
Tuesday, 6 June 2000, 9:00 - 17:30, CS246

The 3rd ICSE workshop on Software Engineering over the Internet will bring together researchers and practitioners that try to use Internet technologies to overcome problems in distributed software development. The goal of the workshop is to exchange ideas on how distributed projects can utilize the Internet to overcome communication, collaboration, and coordination problems. Further, the workshop will discuss how standard SE practice can benefit from open-source approaches and vice versa. Case studies will present empirical evidence on the effectiveness of distributed software development.

W12: MULTI-DIMENSIONAL SEPARATION OF CONCERNS IN SOFTWARE ENGINEERING
http://www.research.ibm.com/hyperspace/workshops/icse2000
By invitation only, Tuesday, 6 June 2000, 9:00 - 17:30, SG21

Separation of concerns can provide a host of well-known and crucial benefits, but only if the concerns that are separated and modularized match the concerns one needs to deal with - which can be of dramatically different kinds in different development contexts. A multi-dimensional approach to separation of concerns must support:

- Multiple, arbitrary kinds (dimensions) of concerns.
- Separation according to these concerns simultaneously.
- Overlapping or interacting concerns, not simply independent or orthogonal ones.

Achieving these properties is challenging. This workshop is intended to bring together researchers and practitioners in this burgeoning area.

W13: 2ND INTERNATIONAL WORKSHOP ON ECONOMICS-DRIVEN SOFTWARE ENGINEERING RESEARCH (EDSER-2)
http://www.cs.virginia.edu/~sullivan/edser2
By invitation only, Tuesday, 6 June 2000, 9:00 - 17:30, SG15

The Second International Workshop on Economics-Driven Software Engineering Research (EDSER-2) will bring together leading researchers and practitioners in several fields who are concerned with understanding economic objectives, constraints, strategies, models and analysis methods in software engineering. The workshop will focus on developing key ideas identified in the first EDSER workshop as being especially important.

W14: THE THIRD WORKSHOP ON INTELLIGENT SOFTWARE ENGINEERING (WISE 3)
http://www.tim.menzies.com/wise3
By invitation only, Tuesday, 6 June 2000, 9:00 - 17:30, CSG01

Real-world software is now so complicated that manual browsing cannot reveal all its subtleties. Automatic tools are required. Many AI researchers now realise that software engineering is the best testbed for AI techniques. But which of these techniques, if any, are cost-effective? This workshop will bring together researchers working in the AI and SE fields who have tried to apply automated intelligence to software engineering. Future WISE workshops would then encourage different solutions to these challenge problems.

DOCTORAL WORKSHOP
http://www.dse.doc.ic.ac.uk/~jnm/doctoral.html
Tuesday, 6 June 2000, 9:00 - 17:30, C1059

This workshop is a forum for graduate students to present and discuss their doctoral research objectives, approaches, and preliminary results. Students will receive guidance and feedback on all aspects of their research from established researchers and the other graduate student attendees.

Doctoral Workshop Committee:
L. Dillon (Michigan State University, USA)
A. Finkelstein (University College London, UK)
A. Fuggetta (Politecnico di Milano, Italy)
J. Kramer (Imperial College, UK)
D. Le Metayer (INRIA/IRISA, France)
J. Magee (Imperial College, UK) (Co-chair)
L. Osterweil (University of Massacchussetts, USA)
M. Pezzè, (Politecnico di Milano, Italy) (Co-chair)
D. Rosenblum (University of California at Irvine, USA)
T. Tamai (University of Tokyo, Japan)

Saturday, 10 June 2000

W15: SOFTWARE PRODUCT LINES: ECONOMICS, ARCHITECTURES, AND IMPLICATIONS
http://www.spe.ucalgary.ca/icse2000pl
By invitation only, Saturday, 10 June 2000, 9:00 - 17:30, C1060

Product line engineering is a recent concept and one of the hottest topics in software engineering, aiming at synergy effects in software development. Diverse benefits like cost reduction, decreased time-to-market, and quality improvement can be expected from reuse of software assets. But also non-technical benefits can be expected as result of product branding, minimizing marginal costs, and sharing organisational costs. On the other hand, product lines introduce additional complexity: The planning and/or development of more than one product at a time has to be managed technically and organizationally. This workshop aims at sharing conceptual and practical experience by establishing contacts and starting the discussion between experts and practitioners from academia and industry.

W16: AGENT-ORIENTED SOFTWARE ENGINEERING (AOSE - 2000)
http://www.csc.liv.ac.uk/~mjw/aose2000/
Saturday, 10 June 2000, 9:00 - 17:30, C1058

Software agents and multi-agent systems have grown into one of the most active areas of research in computer science. The concept of an agent as an autonomous system, capable of interacting and coordinating with other agents in order to satisfy its design objectives, is a natural one for software designers. Just as we can understand many systems as being composed of passive objects, which have state, and upon which we can perform operations, so we can understand many others as being made up of interacting, coordinating agents. In this workshop we will examine the credentials of agent-based approaches as a software engineering paradigm. Both purely theoretical papers and papers that simply discuss agent applications will be presented.

Wednesday June 7th Main Program

TP: Technical Papers, **SEAT:** Software Engineering Education and Training, **FoSE:** Future of Software Engineering,
RD: Research Demos, **PETT:** Practical Experience and Technology Transfer

9:00 - 10:45 S1: Plenary Session

Chair: Carlo Ghezzi, FG061 CONCERT HALL

- Opening Ceremony:
  Dr. Roger Downer, President, University of Limerick
  Mr. Noel Treacy T.D., Minister for Science and Technology
- Invited Presentation: Is The New Economy Socially Sustainable? by Manuel Castells, Professor of Sociology, University of California at Berkeley, USA.

10:45 - 11:15 Break

11:15 - 12:45 S2: 5 Parallel Sessions

S2.1 (TP): Components and COTS
Chair: Clemens Szyperski, FG061 CONCERT HALL

- A Case Study: Demands on Component-based Development by Ivica Crnkovic (Mlardalen University), Magnus Larsson (ABB Automation Products AB).
- Investigating and Improving a COTS-Based Software Development Process by Maurizio Morisio (University of Maryland), Carolyn Seaman (University of Maryland, Baltimore County), Amy Parra (CSC), Victor Basili (University of Maryland and Fraunhofer Institute, Maryland), Steve Condon (CSC), Steve Kraft (NASA/GSFC).
- PPT: A COTS Integration Case Study by L. David Balk, Ann Kedia (T. Rowe Price Investment Technologies).

S2.2 (TP): Software Architectures and Product Families
Chair: Wilhelm Schaefer, DG016

- Supporting Diversity with Component Frameworks as Architectural Entities by Jan Gerben Wijnstra (Philips Research Laboratories).
- Requirements Engineering for Product Families by Juha Kuusela (Nokia Research Center), Juha Savolainen (Helsinki University of Technology).
- Extending Requirement Specifications Using Analogy by Yusuf Pisan (Macquarie University, Sydney).

S2.3 (FoSE): Reasoning and Representation
Chair: Anthony Finkelstein, CSG01

- Reasoning & Analysis: A Roadmap by Daniel Jackson, Martin Rinard (Massachusetts Institute of Technolgy, USA).
- Formal Specification: A Roadmap by Axel van Lamsweerde (Université Catholique de Louvain, Belgium).
- Mathematical Foundations of Software Engineering: A Roadmap by Tom Maibaum (Kings College, London, UK)

S2.4 (PETT): Technology Transfer "in the large"
Chair: TBA, FG042

- From MCC to CMM: Technology Transfer Wars Lost and Won by Bill Curtis (Teraquest, Inc).
- Fraunhofer: The German Model for Applied Research and Technology Transfer by Dieter Rombach (Fraunhofer Institute for Experimental Software Engineering - IESE).
- Discussion: What are the Success Criteria for Technology Transfer? – Does Culture Matter?

S2.5 (TP) Panel: Impact of Software Engineering Research upon Practice
Chair: Leon Osterweil, University of Massachusetts at Amherst, FB028
Panelists: Barry W. Boehm (University of Southern California) Michael Evangelist (National Science Foundation), Volker Gruhn (University of Dortmund), Edward F. Miller Jr. (Software Research Associates, Inc.)

12:45 - 14:15 Lunch

14:15 - 15:15 S3: Plenary Session

Chair: Alexander L. Wolf, FG061 CONCERT HALL

- Awards Presentation

SIGSOFT Outstanding Research Award: Victor R. Basili
SIGSOFT Distinguished Service Award: Marvin Zelkowitz
2000 Harlan D. Mills Practical Visionary award: Barry Boehm

15:30 - 16:30 S4: 5 Parallel Sessions

S4.1 (TP): New Perspectives on Software Engineering
Chair: Dan Hoffman, FG061 CONCERT HALL

- 'It's Engineering Jim but not as we know it': Software Engineering - solution to the software crisis, or part of the problem? by Antony Bryant (Leeds Metropolitan University).
- Producing More Reliable Software: Mature Software Engineering Process vs. State-of-the-Art Technology? by James Widmaier (National Security Agency), Carol Smidts, Xin Huang (University of Maryland).

S4.2 (TP): Data Analysis
Chair: Walcelio Melo, DG016

- Improving Problem-Oriented Mailing List Archives with MCS by Robert Brewer (University of Hawaii at Manoa).
- Broad-Spectrum Studies of Log File Analysis by James Andrews, Yingjun Zhang (University of Western Ontario).

S4.3 (FoSE): Infrastructure I
Chair: Jeff Kramer, CSG01

- Software Engineering Tools and Environments: A Roadmap by Willian Harrison, Harold Ossher, Peri Tarr (IBM Thomas J. Watson Research Centre, USA)
- Software Configuration Management: A Roadmap by Jacky Estublier (Dassault Systemes/LSR, Grenoble University, France).

S4.4 (PETT): Professionalization of Software Engineering
Chair: TBA, FG042

- Software Development Engineer - A Subjective View of Soft Skills Required by Martin Orsted (Microsoft).
- Software Needs Engineering by Jane B. Grimson (Trinity College Dublin), Hans-Jürgen Kugler (Q-Labs AB).
- Is Software Education Narrowminded? by Peter Morrogh (Lifetime Assurance).
- Discussion: Who Is at Fault: Education, Engineering, Business? Where Do We Go from Here?

Main Program

Rooms are colour coded according to map on back of program.

S4.5 (SEAT): Teaching Demos
Chair: Steve Fickas, FB028
- Lessons Learned from Teaching Reflective Software Engineering using the Leap Toolkit by Carleton A. Moore (University of Hawaii).
- Can Quality Graduate Software Engineering Courses Really Be Delivered Asynchronously On-line? by Stephen Edwards (Virginia Tech).

16:00 - 17:00 Break

17:00 - 18:30 S5: 5 Parallel Sessions

S5.1 (TP): Testing I
Chair: Antonia Bertolino, FG061 Concert Hall
- Multivariate Visualisation in Observation-Based Testing by David Leon, Andy Podgurski, Lee White (Case Western Reserve University).
- An Empirical Study of Regression Test Application Frequency by Jung-Min Kim, Adam Porter (University of Maryland), Gregg Rothermel (Oregon State University).
- Testing Levels for Object-Oriented Software by Yvan Labiche, Pascale Thevenod-Fosse, Helene Waeselynck (LAAS/CNRS), Marie-Helene Durand (Aerospatiale Matra Airbus).

S5.2 (TP): Evolution and Reuse
Chair: Martin Glintz, DG016
- Software Evolution in Componentware Using Requirements/Assurances Contracts by Andreas Rausch (Technische Universitaet Muenchen, FORSOFT).
- An Integrated Cost Model for Software Reuse by Senta Fowler Chmiel, Ravi Gottumukkala, Lisa Zhang, Ali Mili (West Virginia University).
- Data Mining Library Reuse Patterns using Generalised Association Rules by Amir Michail (University of Washington).

S5.3 (FoSE): Process I
Chair: Anthony Finkelstein, CSG01
- Software Process: A Roadmap by Alfonso Fuggetta (Politecnico di Milano, Italy).
- Requirements Engineering: A Roadmap by Bashar Nuseibeh (Imperial College of Science Technology and Medicine, UK), Steve Easterbrook (University of Toronto, Canada).
- Reverse Engineering: A Roadmap by Hausi Muller (University of Victoria, Canada), Jens Jahnke (University of Victoria, Canada), Dennis Smith (Carnegie Mellon University, USA), Margaret-Anne Storey (University of Victoria, Canada), Scott Tilley (University of California Riverside, USA) & Kenny Wong (University of Alberta, USA).

S5.4 (PETT): Panel: Component-Based Software Engineering and the Issue of Trust
Chair: George Heinemann (WPI), *Bill Councill* (Texas Quintessence Corporation), FG042
- Panelists: , Janet S. Flynt (Underwriters Laboratories, Inc), Éoin Redmond (Mannatech, Inc), John R. Speed (former Executive Director of Texas Board of Professional Engineers), Mary Shaw (Carnegie - Mellon University)

S5.5 (SEAT): Panel: Shortages of Qualified Software Engineering Faculty and Practitioners: Challenges in Breaking the Cycle,
Chairs: Nancy Mead (Software Engineering Institute, USA), Hossein Saiedian (University of Nebraska at Omaha, USA). FB028
- Panelists: Donald J. Bagert (Texas Tech. University), Helen Edwards (University of Sunderland), Günther Ruhe (Fraunhofer Institute for Experimental Software Engineering - IESE), Michael Ryan (Dublin City University).

Thursday June 8th Main Program

TP: Technical Papers,
RD: Research Demos,
SEAT: Software Engineering Education and Training,
PETT: Practical Experience and Technology Transfer
FoSE: Future of Software Engineering,

9:00 - 10:00 S6: Plenary Session

Chair: Alexander L. Wolf, FG061

- Invited Presentation: The Future of Software by Grady Booch, Chief Technical Officer at Catapulse, USA.

10:15 - 11:15 S7: 5 Parallel Sessions

S7.1 (TP): Component-Based Systems
Chair: Wolfgang Emmerich FG061

- Towards a Taxonomy of Software Connectors by Nikunj Mehta, Nenad Medvidovic, Sandeep Phadke (University of Southern California).
- A Formal Approach for Designing CORBA-Based Applications by Alberto Coen-Porisini (Università di Lecce), Matteo Pradella, Matteo Rossi, Dino Mandrioli (Politecnico di Milano).

S7.2 TP-SEAT: Software Engineering Training
Chair: Hossein Saeidian, DG016

- Simulation in Software Engineering Training by Anke Drappa, Jochen Ludewig (University of Stuttgart).
- Twenty Dirty Tricks to Train Software Engineers by Ray Dawson (Loughborough University).

S7.3 (FoSE): Methods I
Chair: Norman Fenton, CSG01

- Software Economics: A Roadmap by Barry Boehm (University of Southern California, USA), Kevin Sullivan (University of Virginia, USA).
- Empirical Studies of Software Engineering: A Roadmap by Dewayne Perry (University of Texas at Austin), Adam Porter (University of Maryland), Lawrence Votta (Motorola).

S7.4 (PETT): Experience with a Product Line and Family Approach
Chair: TBA, FG042

- An Approach to Architectural Analysis of Product Lines by Gerald Gannod (Arizona State University), Robyn Lutz (NASA Jet Propulsion Laboratory).
- Introducing a Software Modelling Concept in a Medium-Sized Company by Klaus Schmid, Ulrike Becker-Kornstaedt, Peter Knauber (Fraunhofer Institute for Experimental Software Engineering - IESE), Florian Bernauer (Markant Südwest Software und Dienstleistungs GmbH).

S7.5 (RD): Research Demos
Chair: Michael Goedicke, FB028

- Alcoa: The Alloy Constraint Analyzer by Daniel Jackson, Ian Schechter, Ilya Shlyakhter (MIT).
- Hyper/J™: Multi-Dimensional Separation of Concerns for Java™ by Harold Ossher, Peri Tarr (IBM Research).

11:15 - 11:45 Break

11:45 - 12:45 S8: 5 Parallel Sessions

S8.1 (TP): Testing II
Chair: Mathew Dwyer, FG061 Concert Hall

- Deriving Test Plans from Architectural Descriptions by Antonia Bertolino (IEI-CNR), Flavio Corradini, Paola Inverardi, Henry Muccini (Università di L'Aquila).
- WYSIWYT Testing in the Spreadsheet Paradigm: An Empirical Evaluation by Karen J. Rothermel, Curtis R. Cook, Margaret M. Burnett, Justin Schonfeld (Oregon State University), T.R.G. Green (University of Leeds), Gregg Rothermel (Oregon State University).

S8.2 (TP): Software Architecture
Chair: David Rosenblum, DG016

- Integrating UML Diagrams for Production Control Systems by Hans Koehler, Ulrich Nickel, Joerg Niere, Albert Zuendorf (University of Paderborn).
- Dragonfly: Linking Conceptual and Implementation Architectures of Multiuser Interactive Systems by Gary E. Anderson, T.C. Nicholas Graham, Timothy N. Wright (Queen's University).

S8.3 (FoSE): Methods II
Chair: Barry Boehm, CSG01

- Software Metrics: A Roadmap by Norman Fenton, Martin Neil (Queen Mary & Westfield College).
- Software Engineering Education: A Roadmap by Mary Shaw (Carnegie Mellon University).

S8.4 (PETT): Technology Transfer as an Explicit Business and Process Issue
Chair: TBA, FG042

- From Research to Reward: Challenges in Technology Transfer by Adrian M. Colyer (IBM).
- Technology Transfer Macro-Process - A Practical Guide for the Effective Introduction of Technology by Tetsuto Nishiyama, Kunihiko Ikeda, Toru Niwa (OMRON Corporation).

S8.5 (SEAT) Teaching Demos
Chair: Michael Goedicke, FB028

- Multibook's Test Environment by Nathalie Poerwantoro, Bernd Krämer (FemUniversität Hagen), Abdulmotaleb El Saddik,, Ralf Steinmetz (Darmstadt University of Technology).
- E-Slate: A Software Architectural Style for End-User Programming by George Birbilis, Manolis Koutlis, Kriton Kyrimis, George Tsironis, George Vasiliou (Computer Technology Institute, Greece).

Main Program

Rooms are colour coded according to map on back of program.

12:45 - 14:15 Lunch

14:15 - 15:15 S9: Plenary Session

Chair: Kevin Ryan, FG061 Concert Hall

- Invited Presentation: Dot.Com Versus Bricks and Mortar – The Impact of Portal Technology by Chris Horn, Founder and CEO of IONA Technologies plc., Ireland.

15:30 - 16:30 S10: 5 Parallel Sessions

S10.1 (TP): Open Source and Software Markets
Chair: Richard Kemmerer, FG061 Concert Hall

- A Case Study of Open Source Software Development: The Apache Server by Audris Mockus (Lucent Technologies, Bell Laboratories), Roy Fielding (University of California, Irvine), James Herbsleb (Lucent Technologies, Bell Laboratories).
- Multiple Mass Market Applications as Components by David Coppit, Kevin Sullivan (University of Virginia).

S10.2 (SEAT): Software Engineering Education
Chair: Bashar Nuseibeh, DG016

- Developing and Deploying Software Engineering Courseware in an Adaptable Curriculum Framework by W. Richard Adrion (University of Massachusetts, Amherst).
- Achieving Industrial Relevance with Academic Excellence: Lessons from the Oregon Master of Software Engineering by Stuart Faulk (University of Oregon).

S10.3 (FoSE): Process II
Chair: TBA, CSG01

- Testing: A Roadmap by Mary Jean Harrold (Georgia Institute of Technolgy)
- Software Maintenance and Evolution: A Roadmap by Keith Bennett (University of Durham) Vaclaw Rajlich (Wayne State University).

S10.4 (PETT): Support for Effective Project Estimation
Chair: TBA, FG042

- When the Project Absolutely Must Get Done: Marrying the Organisation Chart with the Precedence Diagram by Stan Rifkin (Master Systems, Inc).
- An Evaluation of the Paired Comparison Method for Software Sizing by Eduardo Miranda (Ericsson Research).

S10.5 (RD): Research Demo
Chair: Gian Pietro Picco, FB028

- A Software Engineering Approach and Tool Set for Developing Internet Applications by David A. Marca, Beth A. Perdue (Open Process Inc.).
- The FUJABA Environment by Ulrich Nickel, Jörg Niere, Albert Zündorf (University of Paderborn).

16:30 - 17:00 Break

17:00 - 18:30 S11: 5 Parallel Sessions

S11.1 (TP): System Model Derivation
Chair: Harald Gall, FG061 Concert Hall

- Inference of Message Sequence Charts by Rajeev Alur (University of Pennsylvania and Lucent Technologies, Bell Laboratories), Kousha Etessami, Mihalis Yannakakis (Lucent Technologies, Bell Laboratories).
- Generating Statechart Designs From Scenarios by Jon Whittle, Johann Schumann (NASA Ames Research Center).
- Object Model Resurrection - An Object-Oriented Software Maintenance Activity by Gokul Subramaniam (NORTEL Networks).

S11.2 (TP): Model Checking
Chair: William Griswold, DG016

- Action Language: A Specification Language for Model Checking Reactive Systems by Tevfik Bultan (University of California, Santa Barbara).
- Three Approximation Techniques for ASTRAL Symbolic Model Checking of Infinite State Real-time Systems by Zhe Dang, Richard Kemmerer (University of California, Santa Barbara).

S11.3 (FoSE): Components and Structure
Chair: Bashar Nuseibeh, CSG01

- Software Architecture: A Roadmap by David Garlan (Carnegie Mellon University)
- Object-Oriented Modelling: A Roadmap by Gregor Engels (University of Paderborn) & Luuk Groenewegen (Leiden University).
- Software Engineering for Middleware: A Roadmap by Wolfgang Emmerich (University College London).

S11.4 (PETT): Technology Transfer in the Internet World
Chair: TBA, FG042

- Technology Transfer: Leakage or Control? by Glorianna Davenport (MIT Media-Lab).
- Grow Fast - Grow Global: How the Irish Software Industry Evolved to this Business Model by Barry Murphy (Openet Telecom Ltd).
- Discussion: Research in Internet Time - Are We Endangering Innovation?

S11.5 (SEAT): PANEL: WHO NEEDS DOCTORS? Reporting from the doctoral workshop 17:00 – 18:30
Chairs: J. Magee (Imperial College, UK)
M. Pezzé (Politecnico di Milano, Italy), FB028
This panel will discuss the role of doctoral research in academia and industry. The doctoral workshop participants will summarize the discussion that took place at the doctoral workshop. The panel will question the relationship between academia and industry, and its impact on doctoral research. Among the questions to be addressed are: To what extent does industry need and support doctoral research? What is the impact of doctoral study on employability of graduates? How can industry contribute to doctoral research?

Friday June 9th Main Program

TP: Technical Papers, **SEAT:** Software Engineering Education and Training, **FoSE:** Future of Software Engineering,
RD: Research Demos, **PETT:** Practical Experience and Technology Transfer

9:00 - 10:00 S12: Plenary Session

Chair: Mehdi Jazayeri, **FG061 Concert Hall**

- Invited Presentation: Requirements Engineering in the Year 00: A Research Perspective by Axel van Lamsweerde, Professor of Computer Science, University of Louvain, Belgium.

10:15 - 11:15 S13: 5 Parallel Sessions

S13.1 (TP): Program Analysis I
Chair: Gary Leavens, **FG061 Concert Hall**

- Component Design of Retargetable Program Analysis Tools that Reuse Intermediate Representations by James Hayes, William Griswold, Stuart Moskovics (Motorola, BCS).
- Towards Efficient and Accurate Program Analysis Using Light-Weight Context Recovery by Donglin Liang, Mary Jean Harrold (Georgia Tech University).

S13.2 (TP): Empirical Studies
Chair: Prem Devanbu, **DG016**

- Using the European Space Agency Data Set: A Replicated Assessment of Common Software Cost Estimation Techniques by Lionel Briand (Carleton University), Tristen Langley (University of New South Wales), Isabella Wieczorek (Fraunhofer Institute for Experimental Software Engineering - IESE).
- Characterisation of Risky Projects Based on Project Managers' Evaluation by Osamu Mizuno, Tohru Kikuno (Osaka University), Yasunari Takagi, Keishi Sakamoto (OMRON Corporation).

S13.3 (FoSE): Properties I
Chair: Jeff Magee, **CSG01**

- Software Reliability & Dependability: A Roadmap by Bev Littlewood (City University), Lorenzo Strigini (City University).
- Software Engineering for Performance: A Roadmap by Rob Pooley (Heriot-Watt University).

S13.4 (PETT): From Research to Business Success
Chair: TBA, **FG042**

- The Making of Orbix and the iPortal Suite by Sean Baker (IONA Technologies plc).
- Daily Build and Feature Development in Large Distributed Projects by Even-André Karlsson (Q-Labs), Lars-Göran Andersson, Per Leion (Ericsson).
- Discussion: Professional Software Engineering and Extreme Programming - Is this a Domain or a Cultural Issue?

S13.5 (RD): Research Demos
Chair: Harald Gall, **FB028**

- Managing Software Artifacts on the Web with Labyrinth by Fabiano Cattaneo, Elisabetta Di Nitto, Alfonso Fuggetta, Luigi Lavazza, Giuseppe Valetto (Politecnico di Milano).
- Galileo: A Tool Built from Mass-Market Applications by David Coppit, Kevin J. Sullivan (University of Virginia).

11:15 - 11:45 Break

11:45 - 12:45 S14: 5 Parallel Sessions

S14.1 (TP): Web-Based Systems
Chair: Gruia-Catalin Roman **FG061**

- Implementing Incremental Code Migration with XML by Wolfgang Emmerich, Cecilia Mascolo, Anthony Finkelstein (University College London).
- Principled Design of the Modern Web Architecture by Roy Fielding, Richard Taylor (University of California at Irvine).

S14.2 (TP): Case Studies
Chair: Lionel Briand, **DG016**

- A Study on Exception Detection and Handling Using Aspect-Oriented Programming by Martin Lippert (University of Hamburg), Cristina Lopes (Xerox PARC).
- A Case Study in Root Cause Defect Analysis by Marek Leszak, Dewayne Perry, Dieter Stoll (Lucent Technologies, Bell Laboratories).

S14.3 (FoSE): Properties II
Chair: Tom Maibaum, **CSG01**

- Software Engineering for Real-time: A Roadmap by Hermann Kopetz (Technische Universität Wien).
- Software Engineering for Safety: A Roadmap by Robyn Lutz (Jet Propulsion Laboratory).

S14.4 (PETT) Practical Experience: Company Case Studies I
Chair: TBA, **FG042**

- Improvement of Configuration Management System by Frank Titze (CAD-UL Computer Aided Design Ulm GmbH).
- Applying and Adjusting a Software Process Improvement Model in Practice: The Use of the IDEAL Model in a Small Software Enterprise by Karlheinz Kautz, Henrik Westergaard Hansen, Kim Thaysen (Aalborg University).

S14.5 (SEAT): Teaching Demos
Chair: Bashar Nuseibeh, **FB028**

- An Interactive Multimedia Software House Simulation for Postgraduate Software Engineers by Helen Sharp (City University London), Pat Hall (The Open University, UK).
- LIGHTVIEWS - Visual Interactive Internet Environment for Learning OO Software Testing by Sita Ramakrishan (Monash University).

12:45 - 14:15 Lunch

Main Program

Rooms are colour coded according to map on back of program.

14:15 - 15:15 S15: 5 Parallel Sessions

S15.1 (TP): Program Analysis II
Chair: Mary Jean Harrold, FG061 Concert Hall
- Bandera: Extracting Finite-State Models from Java Source Code by James Corbett (University of Hawaii), Matthew Dwyer, John Hatcliff, Shawn Laubach, Corina S. Pasareanu, Robby, Hongjun Zheng (Kansas State University).
- Quickly Detecting Relevant Program Invariants by Michael Ernst, Adam Czeisler (University of Washington), William Griswold (University of California, San Diego), David Notkin (University of Washington).

S15.2 (TP): Review and Inspection Techniques
Chair: Pankaj Garg, DG016
- Characterising Implicit Information During Peer Review Meetings by Patrick d'Astous, Pierre N. Robillard (École Polytechnique de Montreal).
- Object-Oriented Inspection in the Face of Delocalization by Alastair Dunsmore, Marc Roper, Murray Wood (University of Strathclyde).

S15.3 (FoSE): Properties III
Chair: TBA, CSG01
- Software Engineering for Security: A Roadmap by Premkumar Devanbu (University of California Davis). Stuart Stubblebine (CertCo).
- Software Engineering for Mobility: A Roadmap by Gruia-Catalin Roman (Washington University), Amy Murphy (Washington University) Gian Pietro Picco (Politechnico di Milano).

S15.4 (PETT): Practical Experience:Company Case Studies II
Chair: TBA, FG042
- European Experiences with Software Process Improvement by Fran O'Hara (Insight Consulting Ltd).
- Software Process Improvement by Object Technology (ESSI PIE 27785 - SPOT) by Antonio Caliò, Massimo Autiero (Caliò Informatica, Tecnopolis), Giuseppe Bux (Tecnopolis CSATA Scrl).

S15.5 (RD): Research Demos
Chair: Gianpaolo Cugola, FB028
- Little-JIL/Juliette: A Process Definition Language and Interpreter by Aaron G. Cass, Barbara Staudt Lerner, Eric K. McCall, Leon J. Osterweil, Stanley M. Sutton Jr., Alexander Wise.
- Analysing Software Architectures with Argus-I by Marlon E. R. Vieira, Marcio S. Dias, Debra J. Richardson.

15:30 - 16:30 S16: 5 Parallel Sessions

S16.1 (TP): Verification and Proofs
Chair: Paola Inverardi, FG061
- An Inheritance-Based Technique for Building Simulation Proofs Incrementally by Idit Keidar, Roger Khazan, Nancy Lynch (Massachusetts Institute of Technology), Alex Shvartsman (University of Connecticut).
- Verification of Time Partitioning in the DEOS Scheduler Kernel by John Penix (NASA Ames Research Center), Willem Visser (RIACS/NASA Ames Research Center), Eric Engstrom, Aaron Larson, Nichloas Weininger (Honeywell).

S16.2 (TP): Visual Techniques
Chair: Richard Taylor, DG016
- Graphical Animation of Behavior Models by Jeff Magee, Nat Pryce, Dimitra Giannakopoulou, Jeff Kramer (Imperial College).
- On the Design and Use of Diagrams in Software Engineering by Corin Gurr, Konstantinos Tourlas (University of Edinburgh).

S16.3 (FoSE): Infrastructure II
Chair: Wolfgang Emmerich, CSG01
- Databases in Software Engineering: A Roadmap by Klaus Dittrich, Dimitrios Tombros, Andreas Geppert (University of Zurich).
- Software Engineering on the Internet: A Roadmap by Luca Bompani, Paolo Ciancarini, Fabio Vitali (University of Bolgona).

S16.5 (RD): Research Demos
Chair: David Rosenblum, FB028
- Bandera: A Source-level Interface for Model Checking Java Programs by James C. Corbett, Matthew B. Dwyer, John Hatcliff, Robby.
- Developing Mobile Computing Applications with LIME by Gian Pietro Picco, Amy L. Murphy, Gruia-Catalin Roman

16:45 - 17:15 S17: Plenary Session

Chair: Hausi Mueller, FG061 Concert Hall
- ICSE Summary and Prospects

Social Program

WHILE YOU ARE WITH US

Ireland will welcome you at any time of the year but June is an ideal time to visit. Days are long, the sun sets after 10pm, the tourist season is just starting and the weather - statistically - is at its best. Limerick is the gateway to Ireland's scenic west. An ancient city, with a charter predating that of London, Limerick epitomises Ireland's industrial and cultural renaissance. A resurgent economy, driven in large part by the IT sector, has resulted in a reinvigorated social and cultural scene that has, nonetheless, lost none of its traditional spontaneity and charm. You have to look very hard to find traces of "Angela's Ashes" – the Pulitzer Prize winning memoir by Limerick-born Frank McCourt.

The many attractions of Limerick and its hinterland will be in full swing in June. The city itself boasts intimate galleries, a small theatre, excellent restaurants and the world-class Hunt Museum housed in the restored Custom House. King John's Castle dominates the Shannon at one of its many bridges while St. Mary's Cathedral is a small-scale gothic masterpiece dating from the 12th century. Above all, Limerick is a haven for Irish traditional music and you can find a lively "session" every night of the week.

Limerick is the heart of the Shannon region. A short drive takes you to the wonders of the Burren, the Cliffs of Moher or the beauty of Galway Bay. An additional short boat trip reaches the Aran Islands, last outpost of Europe and a stronghold of the Irish (Gaelic) language. To the east flows the Shannon river and its hills and lakes; to the south lies Kerry and the legendary beauty of Dingle and Killarney.

The University of Limerick is proud to host ICSE. The conference center is the University Foundation Building, which houses research groups, faculty offices, lecture theatres, display areas and a 1,000 seat concert hall. For sports fans the university provides excellent facilities for swimming, tennis, squash, weights, running and quiet riverside jogging. Other essential services are also on hand, including banks, shops and, of course, two pubs.

The social program has been devised to fit around a full and busy ICSE. The welcome reception on Tuesday evening will be held in the Concert Hall atrium, where you can enjoy salmon salad and wine, to the accompaniment of a string quartet. On Wednesday there will be a barbeque on the university campus, with Guinness beer and Irish music and a chance to learn some Irish dancing. These events, and lunches on Wednesday, Thursday and Friday, are all included in your conference registration. The conference banquet on Thursday evening will be held in Bunratty, where the Folk Park will remain open especially for you to visit. After the gala dinner in Fitzpatrick's Hotel there will be a short, high-energy exhibition of the best in traditional Irish song, music and dance. In addition we will be pleased to recommend tours, sights, restaurants, golf courses and local sporting events. If you have any particular interests inquire at the registration / information desk and we will be happy to assist you. Ireland is proud to host ICSE 2000 and we wish you Céad Míle Fáilte—one hundred thousand welcomes.

Kevin Ryan,
Conference Organisation Chair, ICSE 2000

TUESDAY, 6 JUNE 2000

14:00 Optional half-day tour of Limerick City (extra cost).

18:00 Welcome reception Salmon salad and wine served in the Concert Hall atrium, with the accompaniment of the Dolmen String Quartet.

WEDNESDAY, 7 JUNE 2000

9:00 Optional full day tour of Killarney including lunch (extra cost).

19:30 Barbeque in the Stables Courtyard (inside the University campus). Beverages supplied by Guinness. Music by the Saosaire Group and a chance to learn some traditional Irish dance.

THURSDAY, 8 JUNE 2000

18:30 Coaches depart for visit to and drinks reception at Bunratty Folk Park.

20:30 Gala Banquet in Fitzpatrick's Bunratty Hotel followed by a short display of traditional Irish song, music, and dance by Limerick group Planxty O'Rourke.

FRIDAY, 9 JUNE 2000

9:30 Optional full-day tour of Galway, including lunch (extra cost).

Research and Teaching Demos, Posters

RESEARCH DEMOS: FB028

S7.5 **Daniel Jackson, Ian Schechter, Ilya Shlyakhter.** Alcoa: The Alloy Constraint Analyzer.

S7.5 **Harold Ossher, Peri Tarr.** Hyper/J(tm): Multi-Dimensional Separation of Concerns for Java(tm).

S10.5 **David A. Marca, Beth A. Perdue.** A Software Engineering Approach and Tool Set for Developing Internet Applications.

S10.5 **Ulrich Nickel, Jörg Niere, Albert Zündorf.** The FUJABA Environment.

S13.5 **Fabiano Cattaneo, Elisabetta Di Nitto, Alfonso Fuggetta, Luigi Lavazza,Giuseppe Valetto.** Managing Software Artifacts on the Web with Labyrinth.

S13.5 **David Coppit, Kevin J. Sullivan.** Galileo: A Tool Built from Mass-Market Applications.

S15.5 **Aaron G. Cass, Barbara Staudt Lerner, Eric K. McCall, Leon J.Osterweil, Stanley M. Sutton Jr., Alexander Wise.** Little-JIL/Juliette: A Process Definition Language and Interpreter.

S.15.5 **Marlon E. R. Vieira, Marcio S. Dias, Debra J. Richardson.** Analysing Software Architectures with Argus-I.

S16.5 **Gian Pietro Picco, Amy L. Murphy, Gruia-Catalin Roman.** Developing Mobile Computing Applications with LIME.

S16.5 **James C. Corbett, Matthew B. Dwyer, John Hatcliff, Robby.** Bandera: A Source-level Interface for Model Checking Java Programs.

TEACHING DEMOS: FB028

S4.5 **Carleton Moore.** Lessons Learned from Teaching Reflective Software Engineering using the Leap Toolkit.

S4.5 **Stephen Edwards.** Can Quality Graduate Software Engineering Courses Really Be Delivered Asynchronously On-line?

S8.5 **Nathalie Poerwantoro, Abdulmotaleb El Saddik, Bernd Kramer, Ralf Steinmetz.** Multibook's Test Environment.

S8.5 **George Birbilis, Manolis Koutlis, Kriton Kyrimis, George Tsironis,George Vasiliou.** E-Slate: A software architectural style for end-user programming.

S14.5 **Helen Sharp, Pat Hall.** An Interactive Multimedia Software House Simulation for Postgraduate Software Engineers.

S14.5 **Sita Ramakrishan.** LIGHTVIEWS - Visual Interactive Internet Environment for Learning OO Software Testing.

POSTERS FOUNDATION BUILDING (ATRIUM)

P1: **B. Michiels and B. Wydaeghe** (Vrije Universiteit Brussel). Component Composition.

P2: **R. Lencevicius, A. Ran and R. Yairi** (Nokia Research). Third Eye - Specification Based Analysis of Software Execution Traces.

P3: **S. Lei, M. Smith and G. Succi** (University of Calgary). Empirical Investigation of a Noval Approach to Check the Integrity of Software Engineering Measuring Processes.

P4: **R. O'Connor, R. Cochran and R. Moynihan** (Dublin City University and Catalyst Software Dublin). Prompter - A Project Planning Assistant.

P5: **L. Thomas** (University of Wales in Aberystwyth). The Implication of Different Learning Styles on the Modelling of Object-Oriented Systems.

P6: **S.T. Acuna, G.E. Barchini and M. Sosa** (Universidad Nacional de Santiago del Estero). A Culture-Centred Multilevel Software Process Cycle Model.

P7: **C. Liu and D.J. Richardson** (University of California, Irvine). Using Application States in Software Testing.

P8: **J.F. Ramil and M.M. Lehman** (Imperial College London). Effort Estimation from Change Records of Evolving Software

P9: **A.D. Sloane** (EMC Cork). Modelling Deployment and Configuration of CORBA Systems with UML.

P10: **T. Walsh, P. Nixon and S. Dobson** (Trinity College Dublin). As Strong As Possible Mobility.

P11: **K.S. Barber and S.R. Jernigan** (University of Texas at Austin). Hybrid Domain Representation Archive (HyDRA) for Requirements Model Synthesis across Viewpoints.

P12: **Y.H. Daabaj** (University of Salford). The Use of Task Analysis Methods in Support of the Development of Interactive Systems.

P13: **A. Liu, I. Gorton and P. Greenfield** (CSIRO Sydney). DeBOT - An Approach for Constructing High Performance, Scalable Distributed Object Systems.

P14: **G. Froelich, A. Kamel and P. Sorenson** (University of Alberta). Exploring O-O Framework Usage.

PDSE-2000

ROOM: C1063

Software Engineering for Parallel and Distributed Systems

The PDSE series of events provides a forum for exchange of information and publication of the latest technological and theoretical advances in software engineering for parallel and distributed systems. The Symposium will focus on the problems that are unique to the software engineer developing parallel and distributed systems. It will be of interest to both industrial and academic practitioners and researchers who have experience in developing software for parallel and distributed systems. It will also be relevant to practicing software engineers who are interested in developing expertise in the field for research and development purposes.

The PDSE symposium, now in its fifth successive year, will complement the experiences of previous years, the year 2000 event will focus more strongly on the areas of distributed systems design. It is also appropriate at this point in the development of the event to consider how research in the area is effecting industrial best practice.

IWPC2000

ROOMS:D1050 and C1059

International Workshop on Program Comprehension

Theme: "Combining Techniques and Experiments"

Comprehending programs written by others is at the heart of various software engineering activities. Program comprehension is performed when one reuses, reengineers, or enhances existing (or legacy) programs. It is also performed during review or code walk-through of new programs.
This workshop will gather together practitioners and researchers from academia, industry, and government to review the current state of the art and explore solutions to the program comprehension problem. The focus of the workshop is to explore the opportunities of both techniques and experiments to further improve program understanding: evaluating existing techniques by experiments and exploiting experimental results to revise existing or to develop new techniques.

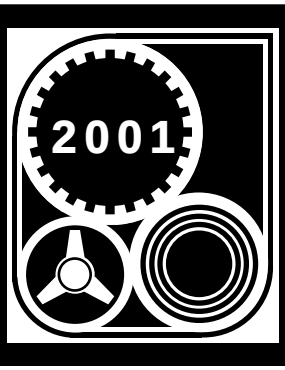
SPICE2000

ROOM:C1056

Software Process Improvement and Capability Determination

The SPICE Users Group is pleased to announce the first International SPICE Conference. This Conference builds upon the achievements of the International SPICE Symposia held on a regular basis since 1994, and provides an opportunity for participation by the wider software engineering community.
One of the basic intentions of this conference is to offer a forum for discussion and exchange of experience in the use of process assessment in the contexts of improvement and capability determination. Keynote addresses will explore the future evolution of ISO 15504 and the harmonisation of the CMM Integration models with the international standard. Invited presentations will focus on application of 15504-related approaches by key international organisations. A highlight of the conference will be a workshop for assessors presenting a range of experiences in application of ISO 15504.

ICSE 2001



May 2–9, 2001
Toronto, Canada

23rd International Conference on Software engineering

Westin Harbour Castle Hotel

Advance Call for Participation

Today, the engineering of software profoundly impacts world economics. Prominent software failures and widespread changes, such as the Year 2000 problem and the European currency change, demonstrate how much the world economy depends on software. ICSE 2001, the premier conference for software engineering, will review what we have learned from these experiences and, in particular, how we can prepare for future, mass changes involving diverse software systems. For example, the software industry is implementing mass changes to use Internet technology to streamline the way businesses gather and distribute information. Furthermore, because the increased popularity of the Internet has brought electronic commerce to small businesses and individual consumers, the software industry is working to adapt both legacy and new applications to a network-centric environment. These changes involve exploiting Web-based user interfaces, component-based systems, and distributed cooperative information systems. Some of these changes, such as those that provide solutions to the Year 2000 problem, have fixed deadlines. However, minimizing time to market is critical for all businesses. These, and other events on the horizon that are equally vital to the survival of large and small businesses, make it imperative that the software engineering community develop infrastructures, methods, and tools to help effect mass software change.

ICSE offers a wide spectrum of tutorials, workshops, industrial experiences, and research results on software construction and evolution from legacy systems to object technology. We invite you to participate in ICSE 2001 to help us build an exciting forum for exchanging ideas and experiences in this ever expanding and critical field of software engineering. We invite submissions of various types on all aspects of software engineering: high-quality research papers, experience papers and practitioner reports; tutorial, workshop, and panel proposals; and exhibits, demonstrations, and posters.

General Chair

Hausi Müller
University of Victoria, Canada
hausi@csr.uvic.ca

Program Chairs

Mary Jean Harrold
Georgia Institute of Technology, USA
harrold@cc.gatech.edu

Wilhelm Schäfer
University of Paderborn, Germany
wilhelm@uni-paderborn.de

Important Dates

Technical Papers	August 28, 2000
Case Study Papers	August 28, 2000
Tutorial Proposals	October 16, 2000
Workshop Proposals	October 16, 2000
Panel Proposals	October 16, 2000
Doctoral Symposium	October 16, 2000
Research Demos	October 16, 2000
Posters	October 16, 2000
Exhibits	January 8, 2001

http://www.csr.uvic.ca/icse2001/

ICSE 2002

International Conference on Software Engineering

May 19—25, 2002

Buenos Aires, Argentina

General Chair

Will Tracz
Lockheed Martin, USA
will.tracz@lmco.com

Program Chairs

Jeff Magee
Imperial College, UK
jnm@doc.ic.ac.uk

Michal Young
University of Oregon, USA
michal@cs.uoregon.edu

Campus Map

FOUNDATION BUILDING	FG016 (CONCERT HALL), FG042, FB028, F1030, F2030 MUSIC ROOM B
COMPUTER SCIENCE BUILDING	CSG01, CSG25, CSG27, CSG28, CS246, EMAIL/INTERNET ACCESS
SCHUMAN BUILDING	SG15 - SG21, S114 - S116, S205 - S206
MAIN BUILDING BLOCK C	C1056 - C1063
MAIN BUILDING BLOCK D	DG016, D1049, D1050
MAIN BUILDING BLOCK E	EM009, EM010
THE CAFETERIA	CAPACITY: 450
THE MAIN RESTAURANT	CAPACITY: 200
PRIVATE DINING ROOM	CAPACITY: 50
SCHUMAN RESTAURANT	CAPACITY: 100