



**26<sup>th</sup> International Conference on Software Engineering (ICSE 2004)**  
**May 23<sup>rd</sup> - 28<sup>th</sup>, Edinburgh, Scotland, UK.**

## Advance Program

ICSE is the premier software engineering conference, providing a forum for researchers, practitioners and educators to present and discuss the most recent innovations, trends, experiences and concerns in the field of software engineering.

The **ICSE 2004** program includes keynote talks by leaders in the field, presentations of refereed technical papers on innovative research and experience at the cutting edge of practice, a linkages track, tutorials by technology experts, workshops on important emerging topics, demonstrations of significant research prototypes, a doctoral symposium for PhD students and an informal poster/demo session. The main program will be complemented by an array of social events, providing further opportunities for informal networking.

In 2004, the ancient Scottish city of Edinburgh hosts the **ICSE** conference. With the conference hotels and venue all within easy reach of the Castle, Royal Mile and medieval old town, **ICSE 2004** promises to be to be a highly enjoyable as well as stimulating conference.

## Program Overview

Thursday 20<sup>th</sup> - Tuesday 25<sup>th</sup> May: Co-located Events

Monday 24<sup>th</sup> May: Doctoral Symposium

Monday 24<sup>th</sup> - Tuesday 25<sup>th</sup> May: Tutorials, Workshops and Adjunct (also co-located) Events

Wednesday 26<sup>th</sup> - Friday 28<sup>th</sup> May: Technical Papers, Demos and Posters, Linkages Tracks and Keynotes

Saturday 29<sup>th</sup> - Sunday 30<sup>th</sup> May: Co-located Events

## ICSE Week Preview

Sun 23 <sup>rd</sup> May	Mon 24 <sup>th</sup> May	Tues 25 <sup>th</sup> May	Weds 26 <sup>th</sup> May	Thurs 27 <sup>th</sup> May	Fri 28 <sup>th</sup> May
<i>ICSE 2004 Week</i>					
	Workshops				
	Tutorials				
			Main Conference and Technical Program		
		Doctoral Symposium			
	Adjunct Event: EASE 2004				
	Adjunct Event: CBSE7				

## Keynote Speakers

ICSE 2004 has a superb line-up of keynote speakers:

**Richard Stallman.** The founder of the GNU project, developer of GNU Emacs and much else besides. 'Father' of the free software movement and campaigner for civil liberties. The holder of numerous awards and distinctions, including the ACM Grace Hopper medal and membership of the National Academy of Engineering.

**Janet Thornton.** Director of the European Bioinformatics Institute (EBI). World leading researcher in bioinformatics and biomolecular structure. The effort to map the human genome sequence was unquestionably one of the extraordinary scientific achievements of the twentieth century. This revolution in biology brings profound challenges for computing.

**Karl Lieberherr.** Distinguished software development researcher. Pioneer of software component research. Responsible for innovative early work in aspect-oriented and adaptive programming.

## Technical Program Preview

Weds 26 <sup>th</sup> May	Thursday 27 <sup>th</sup> May	Fri 28 <sup>th</sup> May
Welcome and Keynote 1 - Richard Stallman	Keynote 2 - Karl Lieberherr	ICSE Future and Keynote 3 - Janet Thornton
Testing I	Testing II	Slicing
Patterns and Frameworks	Software Architecture I	Software Architecture II
Requirements	Process and Project Management	Software Configuration Management and Deployment
Linkages 1 - John Pinkus	Linkages 2 - Ian Foster	Linkages 3 - Mark Handley
Formal Demos - Analysis and Visualization	Formal Demos - Design Tools	Formal Demos - Requirements and Specification
Quality of Service	Decentralized Systems	Dynamic Configuration
Verification	Analysis Tools	Static Analysis
Unified Modeling Language	Dynamic Analysis	Object-Oriented Programming
Panel 1 - Design: Supporting Reflective Practitioners	Panel 2 - MDA in Practice	Panel 3 - Agile Development: Evaluation and Experience
	UK BOF	
		Farewell
Linkages 4 - Ken Birman	Awards and MIP	
Empirical Methods		
Feature-Based Software Engineering		
Whisky Tasting Reception	Reception - Royal Museum of Scotland	

## Workshops

ICSE 2004 is hosting an extensive series of 1 and 2 day workshops, spanning a wide range of important topics. Full details can be found online at <http://conferences.iee.org/icse2004/>, or at the workshop home pages listed below.

### **Monday 24<sup>th</sup> May – 1 Day Workshops**

W3S

#### **Software Engineering for High Performance Computing System (HPCS) Applications**

<http://csdl.ics.hawaii.edu/se-hp>

High performance computing systems are used to develop software for wide variety of domains including nuclear physics, crash simulation, satellite data processing, fluid dynamics, climate modelling, bioinformatics, and financial modelling. Recent initiatives in the HPCS community, such as the DARPA High Productivity Computing Systems program, recognize that dramatic increases in low-level HPCS benchmarks do not necessarily translate into high-level increases in actual development productivity. This creates an opportunity for the software engineering community to apply our techniques and knowledge to a new and important application domain. This interdisciplinary workshop will bring together researchers and practitioners from the SE and HPCS communities to share perspectives and define a research agenda for improving HPC development productivity.

W4S

#### **1st International Workshop on Advances and Applications of Problem Frames (IWAAPF 2004)**

<http://www.cse.unsw.edu.au/~karlc/ProblemFramesWorkshop.htm>

Michael Jackson's Problem Frames are a new and highly promising approach to problem description and requirements engineering. Their focus moves the engineer back to the problem to be solved rather than the software to solve an undefined problem. Problem Frames are patterns of known types of problem that are shaped to be easily referenced into the solution that best matches a problem or set of related problems; they are the design patterns of the problem world. The influence of the Problem Frames approach and related work is spreading in the fields of Domain Modelling, Process Modelling, Requirements Engineering and Software Architecture. It is widely recognised that the early lifecycle phases in a software project are crucial to its success. Poor requirements definition and process almost always leads to a poorer product or a cancelled project. The Problem Frames approach has the potential to have a major impact upon the way software engineering problems are thought about, described and later developed. The scope of this workshop is to address application and usage successes and issues with the Problem Frames approach to aspects of software engineering and to explore where new areas of research might arise.

W12S

#### **The Third International Workshop on Global Software Development (GSD2004)**

<http://gsd2004.uvic.ca>

The goal of this workshop is to provide an opportunity for researchers and industry practitioners to explore both the state-of-the art and the state-of-the-practice in GSD. The workshop will foster interaction between practitioners and researchers and help grow a community of interest in this area. Practitioners experiencing challenges in GSD will share their concerns and successful solutions and learn from research about current investigations. Researchers addressing GSD will gain a better understanding of the key issues facing practitioners and share their work in progress with others in the field.

W13S

### **The Second Workshop on Software Quality**

<http://attend.it.uts.edu.au/icse2004>

To develop software quickly, on time and within budget is not good enough if the product developed is full of defects and today, software stakeholders are demanding higher quality software than ever before. As the software market matures, users want to be assured of quality. They no longer accept the claims of the IT department at face value, but expect demonstrations of quality. In recent years, much of software engineering research has focussed on standards, methodologies and techniques for improving software quality, measuring software quality and software quality assurance. Most of this research is focused on an internal view of quality whereas few measures of the customer view of quality exist. The Second Workshop on Software Quality aims to bring together academic, industrial and commercial communities interested in software quality in order to discuss the different technologies that have been defined and used in the software quality area, with topics of interest to this discussion spanning the full range of software quality issues.

W15S

### **Second International Workshop on Remote Analysis and Measurement of Software Systems (RAMSS 04)**

<http://measure.cc.gt.atl.ga.us/ramss/>

The way software is produced and used is changing radically. Not so long ago software systems had only a few users, and ran on a limited number of mostly disconnected computers. Nowadays the number of software systems, computers, and users has dramatically increased. Moreover, most computers are connected through the Internet. Although these changes raise new issues for software engineers, they also represent new opportunities to greatly improve the quality and performance of software systems. The goal of this workshop is to bring together researchers and practitioners interested in exploring how the characteristics of today's computing environment can be leveraged to improve software quality and performance. In particular, the workshop aims to discuss how software engineers can shift substantial portions of their analysis and measurement activities to actual user environments, so as to leverage in-the-field computational power, human resources, and actual user data to investigate the behavior of their systems after deployment and to improve their quality and performance.

### ***Tuesday 25<sup>th</sup> May – 1 Day Workshops***

W2S

### **Workshop on Directions in Software Engineering Environments**

<http://www.cs.auckland.ac.nz/~herm/WoDiSEE2004/>

Software engineering environments are an essential part of practising software engineering, given the complexity of today's software systems and diversity of software processes. Much conceptual or methodological work in software engineering requires adequate tool support in order for researchers and practitioners to make use of these software engineering advances. The goal of this workshop is to provide a forum for exchange of ideas on new software tool construction techniques; innovative software engineering environments; automation, intelligence and integration issues for software tools; novel software engineering environment interfaces and application domains; and innovative tool experience reports. A key outcome will be a workshop summary of the state of the art in software engineering environments research and key directions for future tools research.

W5S

**Third International Workshop on Scenarios and State Machines: Models, Algorithms, and Tools (SCESM04)**

<http://scesm04.upb.de>

Scenarios and state machines have emerged as two important modeling perspectives on the reactive behavior of complex systems. Scenarios typically represent a partial view on the interactions between multiple components; state machines typically represent the complete behavior of individual components. The methodological potentials of the combination of partial and complete behavior perspectives have yet to be fully exploited in the development process for complex, reactive systems. Automated tool support - based on algorithms relating scenarios and state machines for analysis, design, implementation or validation - offers great promise for improving the current practice of software engineering. These issues will be explored at the workshop in the context of realistic case studies.

W6S

**ACSE 2004: Fourth International Workshop on Adoption-Centric Software Engineering**

<http://www.acse2004.cs.uvic.ca>

The key objective of this workshop is to explore innovative approaches to the adoption of software engineering tools and practices through the extension of Commercial-Off-The-Shelf Software (COTS) products and/or middleware technologies. The workshop aims to advance the understanding and evaluation of adoption of software engineering tools and practices by bringing together researchers and practitioners who investigate novel solutions to software engineering adoption issues.

W7S

**International Workshop on Models and Processes for the Evaluation of COTS Components (MPEC 2004)**

<http://www.lsi.upc.es/events/mpec>

Commercial Off-The-Shelf (COTS) evaluation is one of the main activities carried out during the selection and implementation of COTS components. Its importance and also its complexity are growing more and more due to the increasing number of COTS domains and products available. As a consequence, models for representing the evaluation criteria and the evaluations themselves, as well as process to conduct the evaluation activity, are needed.

W8S

**Collaboration, Conflict and Control: The 4th Workshop on Open Source Software Engineering**

<http://opensource.ucc.ie/icse2004/>

The main theme of the 4th workshop is "Collaboration, Conflict and Control." Specifically, the goal of the workshop is to bring together researchers and practitioners for the purpose of discussing the platforms and tools, the techniques and processes, and the organizational structures that are used to support and sustain:

- Communication and collaboration within and between developer and user communities
- The resolution of conflict within development projects (from deciding on improvements to be included in a particular release, to reconciling the goals of not-for-profit community organizations and their industrial collaborators)
- Effective leadership and control of development activities, and of the evolution of specific products (from the management of code commit privileges in a single project, to the roles played by non-governmental standards builders e.g. the Internet Engineering Task Force and the World Wide Web Consortium and governmental policy makers)

W10S

**Second International Workshop on Dynamic Analysis (WODA 2004)**

<http://www.cs.virginia.edu/woda2004>

WODA 2004 will bring together researchers and practitioners working in all areas of dynamic analysis. Dynamic analysis techniques reason over program executions and show promise in aiding the development of robust and reliable large-scale systems. It has become increasingly clear that limitations of static analysis can be overcome by integrating static and dynamic analyses, and that the performance and value of dynamic analyses can be improved by static analyses. Hence, a key focus of the workshop will be on hybrid analyses that involve both static and dynamic components.

W14S

**Software Engineering for Automotive Systems**

<http://www4.in.tum.de/workshops/ase-icse04/>

The next generation of premium cars is expected to host a cumulated amount of up to one gigabyte of binary code of software. To design, implement and manage the complexity of such a heterogeneous distributed system with increasingly short innovation cycles, the techniques and methods of classical embedded systems are not suitable, nor are the known ones in the desktop and business software domains. We see automotive software engineering as a massively emerging research field with heavy impact in industry. This workshop is intended to provide a discussion forum for researchers and practitioners working in the field. A particular goal is to discuss established software engineering concepts for their adoption to the automotive domain towards an explicit discipline of software engineering. Special focus of this workshop are models as well as specification and engineering techniques that support system integration in a field characterized by a sharp division of labor between original equipment manufacturers and different suppliers.

W17S

**International Workshop on Mining Software Repositories (MSR 2004)**

<http://msr.uwaterloo.ca>

The goal of this one-day workshop is to bring together researchers, and practitioners to consider methods to use the data stored in software repositories (such as source control systems, defect tracking systems, and archived project communications) to further understanding of software development practices. We expect the presentations and discussions in this workshop will facilitate the definition of challenges, ideas and approaches to transform software repositories from static record keeping repositories to active repositories used by researchers to gain empirically based understanding of software development, and by software practitioners to predict and plan various aspects of their project.

W19S

**Third Workshop on Architectures for Dependable Systems (WADS)**

<http://www.cs.kent.ac/wads>

Architectural representations of systems have shown to be effective in assisting the understanding of broader system concerns by abstracting away from details of the system. The dependability of systems is known as the reliance that can justifiably be placed on the service the system delivers. The architectural level reasoning about dependability is only just emerging as an important theme in software development, considering the current complexity of emerging applications and the trend of building trustworthy systems from existing untrustworthy systems. This is a twin workshop to another being organised during DSN (International Conference on Dependable Systems and Networks) 28 June to 1 July 2004, Florence, Italy (<http://www.dsn.org/>). The aim is to bring together researchers from both the software architectures and the dependability communities, and to have cross-fertilization from two different communities and to build strong collaboration possibilities among the participants.

further information available online at <http://conferences.iee.org/icse2004/>

## **Monday 24<sup>th</sup> and Tuesday 25<sup>th</sup> May – 2 Day Workshops**

W1L

**Bridging the Gaps II: Bridging the Gaps Between Software Engineering and Human-Computer Interaction**  
<http://www.se-hci.org/bridging/ICSE04>

Almost half of software in systems being developed today and 37 - 50 percent of efforts throughout the software life cycle are related to the system's user interface. For this reason problems and methods from the field of human-computer interaction affect the overall process of software engineering tremendously, and vice versa. Yet despite these seemingly powerful incentives to practice and apply effective SE and HCI method there still exist major gaps of understanding both between suggested practice and how software is actually developed in industry, and between the best practices of each of the fields. The theme of this workshop - the second at ICSE and the fourth in a series over the past two years -is to bring together practitioners and academics in the two fields in an attempt to enumerate and understand these gaps of understanding and communication, with an eventual goal of proposing practical means, shared processes, shared architectures, shared notations etc, to bridge these gaps.

W9L

**The Sixth International Workshop on Economics-Driven Software Engineering Research (EDSER-6)**  
<http://www.soberit.hut.fi/edser-6/>

Traditionally, the study of software engineering has been primarily a technical endeavour with minimal attention given to its economic context. Often technical decisions are made without adequate links to economic and business considerations. The goal of the EDSER workshops is to improve the quality of decision-making in software engineering based on sound economic justifications. EDSER-6 will provide an interactive forum to discuss and advance the state-of-the-art research and practice in economics-driven software engineering. Work in this area utilizes methods and theories from different disciplines, such as decision theory, game theory, economics, and finance to solve technical and managerial problems in software engineering. Participants are solicited in two categories: regular participants and student participants.

W11L

**The 5th International Workshop on Software Process Simulation and Modeling (ProSim 2004)**  
<http://www.prosim.pdx.edu/prosim2004/main.html>

Today, the software industry faces greater challenges than ever before. Customers are demanding more complex, fully functioned software that is easier to use. At the same time, customers want this software to be delivered more quickly and with higher levels of quality. These demands are set in a dynamic project environment of frequently changing technologies, short-staffed projects and globally distributed development teams. The goal of the workshop is to bring together academics and practitioners interested in the area of software process modeling and simulation as well as important industrial issues related to cost estimation and business process design. ProSim 2004 is an international forum for presenting current research themes and applications, for discussing various approaches and to discover underlying similarities at both the applied and theoretical levels. All areas related to software process modeling and simulation, using all applicable techniques and representations are encouraged.

W16L

**Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'04)**  
Workshop Home Page: <http://www.teccomm.les.inf.puc-rio.br/selmas2004/>

With advances in Internet technologies, multi-agent systems (MASs) are undergoing a transition from closed monolithic architectures into open architectures composed of a huge number of autonomous agents that operate and move across different environments. Large-scale open systems involve perhaps thousands of

*further information available online at <http://conferences.iee.org/icse2004/>*

agents not necessarily co-designed to share a common goal. Agents can dynamically leave and enter the system and as most agents are unknown a priori, cannot be supposed to be benevolent to each other. The dynamic arrival and exit of unknown agents, and the possibility of self-interested behavior in the course of the interactions must be taken into account. Moreover, as multiple software agents become collaborative and operate in networked environments, they must be context-aware and deal with environment uncertainty. It makes their coordination and management more difficult and increases the likelihood of the occurrence of exceptional situations, such as security holes, privacy violations, and unexpected global effects. In this context, the goal of the 3rd edition of SELMAS is to bring together researchers and practitioners to discuss the current state and future direction of research in software engineering for open MASs.

W18L

**International Workshop on Distributed Event-Based Systems (DEBS 2004)**

<http://www.cs.colorado.edu/~carzanig/debs04/>

DEBS provides a forum for the presentation and discussion of original and innovative ideas in the area of distributed event-based systems. Event-based systems are software systems that use the implicit invocation or publish/subscribe communication style. The focus of DEBS extends from the architectures and algorithms that implement distributed event-based infrastructures, to the principles and methodologies that support the design of event-based applications, to the issue of event correlation and analysis in the presence of high volumes of events from several sources.

## **Adjunct, Co-located Events**

ICSE 2004 is pleased to host the following adjunct events as part of the main conference program:

**24th - 25th May 2004**

[The 8th International Conference on Empirical Assessment in Software Engineering \(EASE 2004\)](#)

This conference is a unique forum dedicated to the presentation and discussion of research and practical experiences addressing all aspects of empirical assessment and evaluation in software engineering. We welcome papers addressing all aspects of the area, especially where the experiences relate to the problems that arise from the use of emerging technologies within new application domains.

**24th - 25th May 2004**

[International Symposium on Component-based Software Engineering \(CBSE7\)](#)

The premise of the CBSE workshops was that the long-term success of component-based development depends on the viability of an established science and technology foundation for achieving predictable quality in component-based systems. The intent of this symposium is to build on this premise, and to provide a forum for more in-depth and substantive treatment of topics pertaining to predictability. The symposium will bring together researchers and practitioners from a variety of disciplines related to CBSE to help establish cross-discipline insights and to provide a forum for presenting and discussing innovative approaches to CBSE.

## Tutorials

The ICSE 2004 program includes fifteen tutorials, covering a wide range of important topics from research and practice. Tutorials will be held on Monday 24th and Tuesday 25th May, and will be either half a day or a full day in length. Further details about individual tutorials can be found at <http://conferences.iee.org/icse2004/>.

### **Monday 24<sup>th</sup> May – Half Day Tutorials**

T2

#### **Getting Results from Search Based Approaches to Software Engineering**

Presenters: Mark Harman - Brunel University, Joachim Wegener - DaimlerChrysler,

Email: [Mark.Harman@brunel.ac.uk](mailto:Mark.Harman@brunel.ac.uk); [Joachim.Wegener@DaimlerChrysler.com](mailto:Joachim.Wegener@DaimlerChrysler.com)

Search Based Software Engineering uses adaptive search techniques such as evolutionary programming, genetic algorithms and simulated annealing to solve a large variety of software engineering problems right across the software development spectrum. These techniques have been successfully applied to provide automated solutions to difficult software engineering problems such as intelligent test case generation, design optimization, guiding change and release management and many others.

The tutorial includes demos, animations, real world examples, an overview of techniques and existing results and an optional exercise. It is suitable for both those with no prior exposure to search-based techniques and those with existing expertise. You will take away sufficient knowledge and expertise to easily and rapidly apply search based techniques.

T4

#### **Managing Commitments and Risks: Challenges in Distributed Agile Development**

Presenters: Dr. Jyrki Kontio (main contact) - Helsinki University of Technology, Magnus Hoglund - TietoEnator, Jan Ryden - TietoEnator

Email: [jyrki.kontio@hut.fi](mailto:jyrki.kontio@hut.fi); [magnus.hoglund@tietoenerator.com](mailto:magnus.hoglund@tietoenerator.com); [jan.ryden@tietoenerator.com](mailto:jan.ryden@tietoenerator.com)

The tutorial is an unconventional event that makes participants explore the challenges in commitment management and risk management in agile, distributed software development projects. In addition to providing concise and up-to-date briefings on risk management, commitment management, and agile software development fields, the tutorial will use teamwork and role-plays to make all participants experience and discover new insights and challenges modern software management contains. Much of the learning takes place through sharing of experiences and reflecting on them in discussions in teams, solving the problem scenarios presented to the teams. More information and examples of previous "COOLE" tutorials can be found at <http://www.soberit.hut.fi/COOLE/>

T5

#### **Formal Concept Analysis in Software Engineering**

Presenter: Paolo Tonella -ITC-IRST

Email: [tonella@itc.it](mailto:tonella@itc.it)

Concept analysis is a very general method to analyze a binary relationship between arbitrary objects and attributes. Its output is a lattice of so-called concepts, which offers non-trivial insights into the structure underlying the original relationship. Each lattice node (concept) contains maximal sets of objects sharing common attributes. The hierarchy of concepts in the lattice can be interpreted as the possibility to generalize or specialize a concept. In the analysis of software systems, several relationships among the composing entities emerge. For this reason, concept analysis found a very productive application area in software engineering. Static and dynamic relationships among software components can be subjected to

*further information available online at <http://conferences.iee.org/icse2004/>*

concept analysis to obtain information useful during maintenance, for program comprehension, and in the execution of reengineering tasks. The objective of this tutorial is to provide background and methodological knowledge on concept analysis and on its usage in software engineering. This will be achieved by describing three recent, representative applications of concept analysis in detail. They concern respectively the re-organization of a legacy system into cohesive units, the inference of design patterns without any a-priori information, and the decomposition of a software system into computational units (decomposition slices), that may be strongly dependent, weakly dependent or independent with each other. Other examples of applications, presented more succinctly, include the reengineering of class hierarchies, feature location by means of dynamic analysis, and the derivation of a software configuration structure.

T7

### **Software Variability Management**

Presenter: Jan Bosch - University of Groningen

Email: [Jan.Bosch@cs.rug.nl](mailto:Jan.Bosch@cs.rug.nl)

In a variety of approaches to software development, software artifacts are used in multiple contexts or for various purposes. The differences lead to so-called variation points in the software artifact. During recent years, the amount of variability supported by a software artifact is growing considerably and its management is developing as a main challenge in the development, usage and evolution of software artifacts. Examples of approaches where the management of variability is evolving as a challenge include software product families, component-based software development, object-oriented frameworks and configurable software products such as enterprise resource planning systems. The tutorial presents insights gained, techniques developed and lessons learned in the European IST project ConIPF (Configuration in Industrial Product Families) and in other research performed by the software engineering research group at the University of Groningen. The tutorial first establishes the importance of software variability management, defines the concept of variability, discusses notational and visualization aspects, assessment of software artifacts for variability, design of architectures and components for variability, usage of variation points while configuring instantiated software artefacts and, finally, some advanced issues including variation versus composition.

## ***Monday 24<sup>th</sup> May – Full Day Tutorials***

T1

### **Statistical Techniques for Software Engineering Practice**

Presenter: David N. Card - Software Productivity Consortium

Email: [card@software.org](mailto:card@software.org)

This tutorial is directed at software engineering practitioners and researchers seeking a refresher or introduction to basic quantitative and statistical analysis techniques. The course will focus on how these techniques are employed in software process management and quality control activities. It will describe the relationship of these techniques to commonly accepted software process maturity models (e.g., CMMI), approaches (e.g., Six Sigma), and standards (e.g., ISO/IEC 15939). Many factors are combining to promote the use of quantitative and statistical methods by practicing software engineers. While these techniques are not new to industry in general, they are relatively new to the software industry. Consequently, there is significant uncertainty in the community about their difficulty and applicability. This tutorial provides an introduction to basic concepts and provides real examples of how they can be applied to help solve common software engineering problems. The tutorial is organized into five topics as follows:

1. Basic Concepts: measurement information model of ISO/IEC 15939, descriptive statistics, statistical distributions, process stability, homogeneous populations, experiment design, hypothesis testing.
2. Quantitative Techniques: checksheets, pareto charts, histograms, run charts, scatter diagrams (with examples of use).

3. Control Charts: statistical assumptions, attribute charts, selection of control charts, interpretation, common problems (with examples of use).
4. Other Statistical Techniques: simple regression, chi-square, estimation of sample size, reliability models (with examples of use).
5. Models and strategies: high maturity requirements of the CMM and CMMI, relationship to Six Sigma, ISO 9001.

T3

### **An Introduction To Computing System Dependability**

Presenter: John C. Knight - University of Virginia

Email: [knight@cs.virginia.edu](mailto:knight@cs.virginia.edu)

This tutorial will be an enhanced and improved version of the tutorial that I gave at ICSE 2003 in Portland. I propose to use the same general title and basic approach. The tutorial I gave at ICSE 2003 was well attended, and the material seemed to be of interest and to be of value to the attendees. Since ICSE 2003, I have refined the material as part of a course at the University of Virginia for fourth year undergraduates and first year graduate students. That experience together with comments and ideas that arose during the tutorial at ICSE 2003 has enabled me to make substantial improvements to the tutorial. The aim of the tutorial will be to familiarize the software engineer with the basic elements of the theory and practice of dependability. This will include how software interacts with other elements of a system, how the dependability of software affects system dependability, and how software dependability can be achieved. The objectives of the tutorial will be to cover the essential material of the field so that both researchers and practicing software engineers will be able to understand software's role in system dependability. This will permit attendees to understand how their own work impacts the overall system and how they might introduce new ideas into their own work to enhance dependability.

T8

### **Software Architecture Reconstruction**

Presenters: Arie van Deursen - CWI, Claudio Riva - Nokia Research Center

A robust and clear software architecture is often the key discriminator for the success or failure of a software project. The description of software architecture should communicate the essential decisions that have been taken during the design of the software system. Architecture reconstruction is the reverse engineering activity that aims at recovering those decisions that either have been lost (because have not been documented or the original developers have left) or are unknown (because they originate from the system's evolution). The reconstruction is performed by examining the available artifacts (documentation, source code, experts), simulating the system behavior with dynamic analysis techniques and inferring new architectural information that is not immediately evident. This tutorial covers software architecture reconstruction. It addresses, amongst others, the following questions:

- How do we identify architecturally significant information?
- How can we extract, analyze and present it?
- What are the critical issues that have to be considered?
- How do we manage the reconstruction process in a product family?
- What tools and methods are available?

This tutorial will address these and other questions that are relevant for the development of large and complex software systems.

T13

### **Case Studies for Software Engineers**

Presenters: Dewayne E Perry, Susan Eliot Sim, Steve Easterbrook

Email: [perry@ece.utexas.edu](mailto:perry@ece.utexas.edu)

The purpose of this full-day tutorial is to delineate and illustrate the correct use and interpretation of case studies. It will help software engineers identify and avoid common mistakes by giving them a solid grounding in the fundamentals of case studies as a research method. Using an equal blend of lecture and discussion, it aims to provide software engineers with a foundation for conducting, reviewing, and reading case studies. For researchers, this tutorial will provide a starting point for learning how to conduct case studies. They will be able to find, assess, and apply appropriate resources at their home institution. For reviewers, the tutorial will provide guidance on how to judge the quality and validity of reported case studies. They will be able to use the criteria presented in this tutorial to assess whether research papers based on case studies are suitable for publication, allowing them to raise the quality of publications and give appropriate feedback to authors. For practitioners, the tutorial will provide a better awareness of how to interpret the claims made by researchers about new software engineering methods and tools. Practitioners will also gain deeper insights into the roles they can play in designing and conducting case studies in collaborative research projects. As well, they will read case studies more effectively and be better able to identify results suitable for use in their workplace.

### ***Tuesday 25<sup>th</sup> May – Half Day Tutorials***

T6

#### **An Overview of UML 2.0**

Presenter: Bran Selic - IBM Rational Software and Carleton University

Since its adoption as an industry standard in 1997, the Unified Modeling Language (UML) has been adopted widely by both industry and academia. This extensive experience has naturally led to demands for improvements and new capabilities. In September 2000, the Object Management Group-the industrial consortium that controls the UML standard-issued a request for proposal for the first major revision of UML, UML 2.0. This new version was conceived as the basis for the coming generation of model-based development methods, such as OMG's Model-Driven Architecture (MDA). The distinguishing characteristic of these methods is that their primary focus is on the definition and evolution of models rather than programs-with programs being automatically generated from such models. The combination of higher-level abstractions defined in UML and the use of automation provide the potential for a dramatic improvement in productivity and software reliability. Attendees of this half-day tutorial will learn the salient aspects of UML 2.0-from the perspective of one of its primary authors.

T11

### **Architectures and Technologies for Enterprise Application Integration**

Presenters: Ian Gorton - Pacific Northwest National Lab, Anna Liu - Microsoft Australia

Email: [ian.gorton@pnl.gov](mailto:ian.gorton@pnl.gov); [annali@microsoft.com](mailto:annali@microsoft.com)

Architects are faced with the problem of building enterprise scale information systems, with streamlined, automated internal business processes and web-enabled business functions, all across multiple legacy applications. The underlying architectures for such systems are embodied in a range of diverse products known as Enterprise Application Integration (EAI) technologies. In this tutorial, we will highlight some of the major problems, approaches and issues in designing EAI architectures and selecting appropriate supporting technology. The tutorial will present a range of the common architectural styles frequently used for EAI applications. It will also explain Service Oriented Architectures as the current best practise architectural framework for EAI. It will then describe the state-of-the-art in EAI technologies that support these architectural styles, and discuss some of the key design trade-offs involved when selecting an appropriate integration technology (including buy versus build decisions). We will then discuss some of the specific technologies that are available for EAI applications. The focus will be on J2EE, .NET. and Web Services based technologies.

T15

### **Object-Oriented Reengineering Patterns**

Presenters: Serge Demeyer - University of Antwerp, Oscar Nierstrasz - University of Berne

The rapid growth of object-oriented development over the past twenty years has given rise to many object-oriented systems that are large, complex and hard to maintain. These systems exhibit a range of problems, effectively preventing them from satisfying the evolving requirements imposed by their customers. This tutorial addresses the problem of understanding and reengineering such object-oriented legacy systems. The material is presented as a set of "reengineering patterns" -recurring solutions that experts apply while reengineering and maintaining object-oriented systems. The patterns distil successful techniques in planning a reengineering project, reverse-engineering, problem detection, migration strategies and software redesign. The principles and techniques described have been observed and validated in a number of industrial projects, and reflect best practice in object-oriented reengineering. Details about the book can be found at <http://www.iam.unibe.ch/~scg/OORP/>

## ***Tuesday 25<sup>th</sup> May – Full Day Tutorials***

T9

### **Software Modeling Techniques and the Semantic Web**

Presenter: Jin Song Dong - National University of Singapore

Email: [dongjs@comp.nus.edu.sg](mailto:dongjs@comp.nus.edu.sg)

Following the success of XML, W3C envisions the Semantic Web (SW) as the next generation of web in which data are given well-defined and machine-understandable semantics so that they can be processed by intelligent software agents. SW can be regarded as an emerging area from the Knowledge Representation and the Web Communities. The Software Engineering community can also play an important role in the SW development. Modeling and verification techniques can be useful at many stages during the design, maintenance and deployment of SW ontology. We believe SW will be a new research and application domain for software modeling techniques and tools. For example, recent research results have shown that UML, Z and Alloy can provide modeling, reasoning and consistency checking services for SW. On the other hand, the diversity of various software specification techniques and the need for their effective combinations requires an extensible and integrated supporting environment. The success of the Semantic Web may have profound impact on the web environment for software design methods, especially for extending and integrating

different software modeling techniques. This full-day tutorial (with no specific prerequisite) is aimed at both industrial and academic participants. The tutorial will include:

- A detailed introduction to Semantic Web languages (DAML+OIL and OWL) and Semantic Web tools (FaCT and RACER)
- An introduction to software modeling techniques Z, Alloy and UML and a demonstration on how they can facilitate modeling, checking and reasoning about web ontologies
- A military plan ontology case study where we discovered a number of errors in the original ontologies with the help of the combination of RACER, Alloy Analyser and Z/EVES tools
- An illustration on how DAML+OIL can be used to build a Semantic Web environment for supporting, extending and integrating various software specification languages

T10

### **Usability Supporting Architectural Patterns**

Presenters: Len Bass (lead presenter) - CMU, Bonnie E. John - CMU, Natalia Juristo - Universidad Politecnica de Madrid, Maribel Sanchez-Segura - Carlos III University of Madrid

Software architects have techniques to deal with many quality attributes such as performance, reliability, and maintainability. Usability, however, has traditionally been concerned primarily with presentation and not been a concern of software architects beyond separating the user interface from the remainder of the application. A usability-supporting architectural patterns (USAP) describes a usability concern that is not supported by separation alone. For each concern, a USAP provides the forces from the characteristics of the task and environment, the human, and the state of the software to motivate an implementation independent solution cast in terms of the responsibilities that must be fulfilled to satisfy the forces. Furthermore, each pattern includes a sample solution implemented in the context of an overriding separation based pattern such as J2EE Model View Controller. During the tutorial, the instructors will present the concept of a USAP and several examples. The instructors will also facilitate an exercise where attendees will develop their own USAP.

T12

### **Testing Object Oriented Software**

Presenters: Mauro Pezze - University of Milan Bicocca, Michal Young - University of Oregon

Email: [pezze@disco.unimib.it](mailto:pezze@disco.unimib.it); [michal@cs.uoregon.edu](mailto:michal@cs.uoregon.edu)

Object-oriented software requires reconsidering and adapting approaches to software test and analysis. Some traditional test and analysis techniques are easily adjusted to object-oriented software, but others require substantial revision, and yet others need to be introduced to cope with new problems of object-oriented software. This tutorial brings together process and technical aspects of testing object-oriented software in an overall coherent framework that considers what can be simply adapted from conventional test practices and what new and extended techniques are required. Topics include test planning, test design from specification and design documentation, adapting design and code inspection to object oriented software development, intra- and inter-class structural testing, testing programs with exception-handling and threading, test oracles for object-oriented programs, regression testing, and process improvement.

T14

### **Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods**

Presenters: Prof. Barry Boehm - USC, Prof. Richard Turner - George Washington U. and Dept. of Defense

Email: [boehm@usc.edu](mailto:boehm@usc.edu); [rich.turner.CTR@osd.mil](mailto:rich.turner.CTR@osd.mil)

Rapid change and increasing software criticality are driving successful development and acquisition organizations to balance the agility and discipline of their key processes. The emergence of agile methods in the software community is raising the expectations of customers and management, but the methods have shortfalls and their compatibility with traditional plan-driven methods such as those represented by CMMI, ISO-15288, and UK-DefStan-00-55 is largely unexplored. Multiple sources of perplexity -- inconsistent

definitions and interpretations, overgeneralization of successes and failures, confusing of methods' usage and misuse -- complicate the search for clarity of understanding. This tutorial pragmatically examines the aspects of agile and plan-driven methods through examples and case studies. We characterize "home grounds" where the approaches are most likely to succeed, identifying five critical dimensions that describe the agile/plan-driven spectrum. We present a risk-based method for developing balanced strategies that take advantage of the strengths and mitigate the weaknesses of both agile and plan-driven approaches, and that fit the objectives, constraints, and priorities of a particular project or organization. Step-by-step walkthroughs of several example projects show how the method is applied. Finally, we involve participants in an exercise involving hands-on evaluation of their current organizational balance of agility and discipline, identification of their likely directions of change, and development of strategies for evolving their balance of agility and discipline to meet their future objectives and challenges.

## Doctoral Symposium

### *Monday 24<sup>th</sup> May*

The Doctoral Symposium is a closed forum that provides an opportunity for Ph.D. students to discuss their research goals, methods, and results at an early stage in their research, in a critical but supportive environment. The Symposium aims to provide useful guidance for completion of the dissertation research and initiation of a research career. The Symposium and ICSE will also provide an opportunity for student participants to interact with other students at a similar stage in their careers, established researchers, and the broader software engineering community.

#### Invited Participants:

*Genáina Nunes Rodrigues (University College London)*  
A Model Driven Approach for Software Systems Reliability

*Sascha Alda (University of Bonn)*  
Component-based Self-Adaptability in Peer-to-Peer Architectures

*Hridesh Rajan (University of Virginia)*  
One More Step in the Direction of Modularized Integration Concerns

*EunYoung Kang (Delft University of Technology)*  
Parametric Analysis of Real-Time Embedded Systems with Abstract Approximation Interpretation

*Annabella Loconsole (Umeå University)*  
Empirical studies on Requirement Management Activities

*Frank Keenan (University of Ulster at Coleraine)*  
Agile Process Tailoring and problem analysis (APTLY)

*Mika Mäntylä (Helsinki University of Technology)*  
Subjective Evaluation of Software Design Quality

*Steffen Zschaler (Dresden University of Technology)*  
Research Abstract: Semantic Concepts for the Specification of Non-functional Properties of Component-Based Software

*James Jones (Georgia Institute of Technology)*  
Visualization of Test Information to Assist Fault Localization

further information available online at <http://conferences.iee.org/icse2004/>

*Diane Kirk (University of Auckland)*

A Flexible Model for Software Process Simulation

*Nachiappan Nagappan (North Carolina State University)*

Toward a Software Testing and Reliability Early Warning Metric Suite

*Michael Pacione (University of Strathclyde)*

Software Visualisation for Object-Oriented Program Comprehension

*Ciaran O'Reilly (University of Ulster Coleraine)*

A Weakly Constrained Approach to Software Change Coordination

*Roshanak Roshandel (University of Southern California)*

Calculating Architectural Reliability via Modeling and Analysis

*Ana Belén Barragáns Martínez (University of Vigo)*

chi-SCTL/MUS: A Formal Methodology to Evolve Multi-Perspective Software Requirements Specifications

*Jennifer Tenzer (University of Edinburgh)*

Improving UML Design Tools by Formal Games

*Leonardo Mariani (Università di Milano Bicocca)*

Behavior Capture and Test for Verifying Evolving Component-Based Systems

*Pakorn Waewsawangwong (Imperial College London)*

A Constraint Architectural Description Approach to Self-Organising Component-Based Software Systems -- an Extended Abstract

## **Technical Sessions**

***Wednesday 26th May***

### **Testing I**

Using Simulation to Empirically Investigate Test Coverage Criteria

*Lionel Briand, Yvan Labiche & Yunlan Wang*

Automated Generation of Test Programs From Closed Specifications of Classes and Test Cases

*Wee Kheng Leow, Siau Cheng Khoo & Yi Sun*

Bi-Criteria Models for All-Uses Test Suite Reduction

*Jennifer Black, Emanuel Melachrinoudis & David Kaeli*

### **Patterns and Frameworks**

The Dublo Architecture Pattern for Smooth Migration of Business Information Systems

*Wilhelm Hasselbring, Ralf Reussner, Holger Jaekel, Jürgen Schlegelmilch, Thorsten Teschke & Stefan Krieghoff*

Comparison of Software Product Line Design Methods: COPA, FAST, FORM, KobrA and QADA

*Mari Matinlassi*

Oil and Water? High Performance Garbage Collection in Java with JMTk  
*Stephen Blackburn, Perry Cheng & Kathryn McKinley*

### **Requirements**

Eliciting Security Requirements by Construction of Intentional Anti-Models  
*Axel van Lamsweerde*

Theme: An Approach for Aspect-Oriented Analysis and Design  
*Elisa Baniassad & Siobhan Clarke*

Visual Timed Event Scenarios  
*Victor Braberman, Alejandra Alfonso, Nicolas Kisillof & Alfredo Olivero*

### **Quality of Service**

Design Level Performance Modeling of Component-based Applications  
*Yan Liu, Alan Fekete & Ian Gorton*

Precise Service Level Agreements  
*James Skene, Davide Lamanna & Wolfgang Emmerich*

GlueQoS: Middleware to Sweeten Quality-of-Service Policy Interactions  
*Eric Wohlstadter, Stefan Tai, Thomas Mikalsen, Isabelle Rouvellou & Premkumar Devanbu*

### **Verification**

Verifying DAML+OIL and Beyond in Z/EVES  
*Jin Song Dong, Chew Hung Lee, Yuan Fang Li & Hai Wang*

Assume-guarantee Verification of Source Code with Design-Level Assumptions  
*Dimitra Giannakopoulou, Corina Pasareanu & Jamieson Cobleigh*

Compositional Verification of Middleware-Based Software Architecture Descriptions  
*Patrizio Pelliccione, Paola Inverardi & Mauro Caporuscio*

### **Unified Modeling Language**

The Evaluation of Large, Complex UML Analysis and Design Models  
*Brian Berenbach*

Revisiting Statechart Synthesis with an Algebraic Approach  
*Tewfik Ziadi, Loic Helouet & Jean-Marc Jézéquel*

Precise Specification of Design Patterns in UML  
*Jeffrey K. H. Mak, Clifford S. T. Choy & Daniel P. K. Lun*

### **Empirical Methods**

Team-based Fault Content Estimation in the Software Inspection Process  
*Thomas Thelin*

Evidence-based Software Engineering  
*Barbara Kitchenham, Tore Dybå & Magne Jørgensen*

An Empirical Study of Software Reuse vs. Reliability and Stability  
*Parastoo Mohagheghi, Reidar Conradi, Ole M. Killi & Henrik Schwarz*

### **Feature-Based Software Engineering**

SNIAFL: Towards a Static Non-Interactive Approach to Feature Location  
*Wei Zhao, Lu Zhang, Yin Liu, Jiasu Sun & Fuqing Yang*

Feature-Based Decomposition of Inductive Proofs Applied to Real-Time Avionics Software  
*Vu Ha, Murali Rangarajan, Darren Cofer, Harald Ruess & Bruno Dutertre*

### **Thursday 27th May**

#### **Testing II**

Applications of Data Versioning in Database Application Development  
*Ramkrishna Chatterjee, Gopalan Arun, Sanjay Agarwal, Ben Speckhard & Ramesh Vasudevan*

Generating Tests from Counterexamples  
*Adam Chlipala, Thomas Henzinger, Ranjit Jhala & Rupak Majumdar*

Automated Support for Development, Maintenance, and Testing in the Presence of Implicit Control Flow  
*Saurabh Sinha, Alessandro Orso & Mary Jean Harrold*

#### **Software Architecture I**

Towards Safe Distributed Application Development  
*Patrick Eugster, Christian Damm & Rachid Guerraoui*

Design of Large-Scale Polylingual Systems  
*Mark Grechanik, Don Batory & Dewayne Perry*

A Hybrid Architectural Style for Distributed Parallel Processing of Generic Data Streams  
*Alexandre Francois*

#### **Process and Project Management**

Breaking the Ice for Agile Development of Embedded Software  
*Peter Manhart & Kurt Schneider*

Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams  
*Jon Froehlich & Paul Dourish*

Making Resource Decisions for Software Projects  
*Norman Fenton, William Marsh, Martin Neil, Patrick Cates, Simon Forey & Manesh Tailor*

#### **Decentralized Systems**

Using Event-Based Translation to Support Dynamic Protocol Evolution  
*Nathan Ryan & Alexander Wolf*

Efficient Decentralized Monitoring of Safety in Distributed Systems  
*Koushik Sen, Abhay Vardhan, Gul Agha & Grigore Rosu*

Extending the Representational State Transfer (REST) Architectural Style for Decentralized Systems  
*Rohit Khare & Richard Taylor*

### **Analysis Tools**

Validating the Unit Correctness of Spreadsheet Programs

*Tudor Antoniu, Paul Steckler, Shriram Krishnamurthi, Erich Neuwirth & Matthias Felleisen*

A Tool for Writing and Debugging Algebraic Specifications

*Johannes Henkel & Amer Diwan*

Skoll: Distributed Continuous Quality Assurance

*Atif M. Memon, Adam Porter, Cemal Yilmaz, Adithya Nagarajan, Douglas C. Schmidt & Bala Natarajan*

### **Dynamic Analysis**

DiscoTect: A System for Discovering Architectures from Running Systems

*Hong Yan, David Garlan & Bradley Schmerl*

Finding Latent Code Errors via Machine Learning over Program Executions

*Yuriy Brun & Michael Ernst*

An Empirical Comparison of Dynamic Impact Analysis Algorithms

*Alessandro Orso, Taweewat Apiwattanapong, James Law, Gregg Rothermel & Mary Jean Harrold*

### **Friday 28th May**

#### **Slicing**

Efficient Forward Computation of Dynamic Slices Using Reduced Ordered Binary Decision Diagrams

*Xiangyu Zhang, Rajiv Gupta & Youtao Zhang*

Using Compressed Bytecode Traces for Slicing Java Programs

*Tao Wang & Abhik Roychoudhury*

A Fast Assembly Level Reverse Execution Method via Dynamic Slicing

*Tankut Akgul, Vincent Mooney & Santosh Pande*

#### **Software Architecture II**

Polyphony in Architecture

*Bas van der Raadt, Jasper Soetendal, Michiel Perdeck & Hans van Vliet*

Architecting in the Face of Uncertainty: An Experience Report

*Ian Gorton & Jereme Haack*

Using Web Service Technologies to Create an Information Broker

*Mark Turner, Fujun Zhu, Ioannis Kotsiopoulos, Michelle Russell, David Budgen, Keith Bennett, Pearl Brereton, John Keane, Paul Layzell & Michael Rigby*

#### **Software Configuration Management and Deployment**

Mining Version Histories to Guide Software Changes

*Thomas Zimmermann, Peter Weissgerber, Stephan Diehl & Andreas Zeller*

An Experimental, Pluggable Infrastructure for Modular Configuration Management Policy Composition  
*Ronald van der Lingen & Andre van der Hoek*

Imposing a Memory Management Discipline on Software Deployment  
*Eelco Dolstra, Eelco Visser & Merijn de Jonge*

### **Dynamic Reconfiguration**

An Open Framework for Dynamic Reconfiguration  
*Jamie Hillman & Ian Warren*

Dynamic Configuration of Resource-Aware Services  
*Vahe Poladian, David Garlan, Mary Shaw & Joao Sousa*

Autonomous Adaptation to Dynamic Availability Using a Service-Oriented Component Model  
*Humberto Cervantes & Richard Hall*

### **Static Analysis**

DMS: Program Transformations for Practical Scalable Software Evolution  
*Ira Baxter, Christopher Pidgeon & Michael Mehlich*

Heuristic-Based Model Refinement for FLAVERS  
*Jianbin Tan, George Avrunin & Lori Clarke*

Static Checking of Dynamically Generated Queries in Database Applications  
*Carl Gould & Zhendong Su & Premkumar Devanbu*

### **Object-Oriented Programming**

Evaluating Object-Oriented Designs with Link Analysis  
*Alexander Chatzigeorgiou, Spiros Xanthos & George Stephanides*

Responsibilities and Rewards: Reasoning About Design Patterns  
*Neelam Soundarajan & Jason Hallstrom*

Programming with Traits  
*Andrew Black & Nathanael Schärli*

## **Research (Formal) Demos**

***Wednesday 26<sup>th</sup> May***

### **Analysis and Visualisation**

JDBC Checker: A Static Analysis Tool for SQL/JDBC Applications  
*Carl Gould, Zhendong Su, Premkumar Devanbu, University of California, Davis*

GAMMATELLA: Visualization of Program-Execution Data for Deployed Software  
*Alessandro Orso, James Jones, Mary Jean Harrold, Georgia Institute of Technology*

## ***Thursday 27<sup>th</sup> May***

### **Design Tools**

Feature-Oriented Programming and the AHEAD Tool Suite  
*Don Batory, University of Texas at Austin*

AcmeStudio: Supporting Style-Centered Architecture Development  
*Bradley Schmerl, David Garlan, Carnegie Mellon University, [garlan@cs.cmu.edu](mailto:garlan@cs.cmu.edu)*

Design and Implementation of Distributed Crosscutting Features with DADO  
*Eric Wohlstadter, Stoney Jackson, Premkumar Devanbu, University of California, Davis*

## ***Friday 28<sup>th</sup> May***

### **Requirements and Specifications**

The CommUnity Workbench  
*Cristsvco Oliveira, Michel Wermelinger, Universidade Nova de Lisboa*

Statestep: A Tool for Systematic, Incremental Specification  
*Michael Breen, independent consultant,*

Requirements Engineering Tools Go Mobile  
*Norbert Seyff, Paul Grunbacher, Johannes Kepler University Linz,  
Neil Maiden, Amit Tosar, City University London*

## **Panels**

ICSE 2004 Panels will be highly interactive events that emphasise question and answer sessions and encourage lively debate.

## ***Wednesday 26<sup>th</sup> May***

### **Design: Supporting Reflective Practitioners**

Headed by David Redmiles

## ***Thursday 27<sup>th</sup> May***

### **MDA in practice**

Headed by Jean-Marc Jezequel and Wolfgang Emmerich

## ***Friday 28<sup>th</sup> May***

### **Agile Development: Evaluation & Experience**

Headed by Walter Tichy

## Linkages

A series of presentations by renowned experts on some current and future challenges for Software Engineering.

### ***Wednesday 26<sup>th</sup> May***

#### **Graceless Degradation, Measurement, and Other Challenges in Security and Privacy**

*Jon Pincus - Microsoft Research*

Security and privacy issues shine an intense spotlight on areas that are problematic for software engineering in general. One major problem is the lack of graceful degradation; today, it often appears that virtually any security or privacy vulnerability can lead to catastrophic consequences. Obviously, a system that degrades more gracefully - where only a small amount of security and privacy is lost - is preferable; while there have been several successful applications of this principle, it is still hard to generalize. Another significant challenge is measurement of security and privacy; even post-release statistics such as the number of vulnerabilities, patches, or exploits are difficult to interpret meaningfully, but software vendors really need metrics that can be computed much earlier in the engineering process. In both of these areas, approaches focused on attacking subsets of the overall problem show promise, but significant work is needed at both the engineering and research level.

#### **Adding High Availability and Autonomic Behavior to Web Services**

*Ken Birman - Department of Computer Science, Cornell University.*

A new wave of Web Services systems are soon to be rolled out, and when this happens, the software engineering community will experience a sea-change. For the first time, we are building distributed, Web-based applications that truly interoperate and that are likely to play very sensitive roles for the organizations that deploy them. Yet the Web Services architecture inherits a legacy from the Internet: one of best-effort message delivery, inconsistent and unreliable failure detection, ad-hoc end-to-end fault-tolerance mechanisms, and a pervasive lack of information about the state of the network. Internet applications routinely operate in the dark with respect to even the most elementary properties of the environment! In this talk, we'll ask whether it might not be possible to "light up the dark", enabling applications on the client side of a Web Services system to share state, to sense the global state of the system and data centers, and to use this information to greatly improve availability, reliability, self-configuration and management. The Astrolabe system, a novel peer-to-peer technology, could help open the door to a new way of thinking about the client side, and in so-doing contribute to a radical reduction in cost of ownership for large Web Services applications and big advances in autonomic behavior. Astrolabe is part of QuickSilver, a platform tackling many aspects of Web Services availability and "autonomic behavior."

### ***Thursday 27<sup>th</sup> May***

#### **Grid Small and Large: Distributed Systems and Global Communities**

*Ian Foster - Argonne National Laboratory & Department of Computing Science, University of Chicago*

Grid technologies seek to enable collaborative problem solving and resource sharing within distributed, multi-organizational "virtual organizations." Two characteristics of Grid environments make the engineering of systems and applications particularly challenging. First, we face the familiar difficulties that arise when developing software that must provide reliability, performance, and security in an environment that may be heterogeneous, unpredictable, unreliable, and hostile; second, we must allow this software to be deployed, operated, and evolved in an environment characterized by multiple participants with different and perhaps conflicting views on system function and design. I introduce work that is being done to address these challenges. I speak first to "Grids in the small," and describe the work being performed within the Open Grid

further information available online at <http://conferences.iee.org/icse2004/>

Services Architecture framework to define a standard set of Grid protocols layered on Web Services. I explain the relationship of OGSA to Web Services, the evolution of OGSA to better exploit emerging Web Services standards, requirements Grid is placing on those emerging Web Services standards, and the landscape of protocols that are being defined upon Web Services to meet Grid requirements. I then turn to problems associated with "Grids in the large" and discuss how Grid technologies can evolve to address the challenges associated with community development of complex software systems.

**Friday 28<sup>th</sup> May**

### **Evolving the Internet: Changing the Engines in Mid-Flight**

*Mark Handley - Department of Computer Science, University College London*

The Internet has grown very rapidly in the last decade; this phenomenal growth continues today despite the bursting of the .com bubble. At the same time, greater reliability and performance are being demanded, as the Internet becomes mission-critical for many businesses. In this talk I will discuss the hard research problems currently being faced by the Internet, and speculate on some possible solutions. Key to many of these problems is the difficulty of evolving a system of 200 million machines whilst simultaneously keeping it running. The analogy of attempting to change the engines on an aircraft in mid flight is unfortunately an apt one. It is worth noting that Software Engineering researchers and Networking researchers rarely pay much attention to each other's problems and potential solutions. In particular, networking protocol design often lacks the rigour that good software engineering methods could bring to the process. At the same time, much distributed systems middleware attempts to abstract away the fundamental limitations of the network. In passing, this talk will touch on why this may be and what we can do about it.

## **Registration**

Registration for ICSE 2004 will open early in February 2004. Online registration and forms for postal registration will be available through the conference website: <http://conferences.iee.org/icse2004/>.

## **Venue, Location and Accommodation**

**ICSE 2004** will be held at the prestigious Edinburgh International Conference Centre, a perfect venue for ICSE. The EICC is a world class, self-contained venue. Delegates will be able to use the local wireless network to access the Internet at their convenience.

**ICSE 2004** is pleased to be able to recommend a diversity of hotel accommodation, in and around the city centre. The conference hotel is The Point, adjacent to the EICC. Other options have been selected to suit all tastes and budgets. Accommodation can be viewed and booked through the conference website.

The EICC, the conference hotel and a variety of alternative accommodation are all within easy reach of Princes Street, the ancient Castle and the Royal Mile in the centre of the medieval old town. **ICSE 2004** delegates will be ideally situated to explore the attractions of this delightful city.

Edinburgh International airport has direct flights from all major European cities and the USA, with shuttle flights from London every 30 minutes and a short taxi ride into the city centre. The City also has excellent road and rail links.

## Co-located Events

**20th - 21st May**

[The 2nd International Working Conference on Component Deployment \(CD 2004\)](#)

What do we do with components after they have been built? How do we deploy them into their execution environment? How can we evolve them once they have been deployed? While several tools exist to support deployment, the underlying principles are only beginning to emerge. CD 2004 will bring together researchers and practitioners with the goal of developing a better understanding of how deployment takes place in the wider context. Papers relating theory to practice are particularly welcome.

**21st - 22nd May**

[The First European Workshop on Software Architecture \(EWSA 2004\)](#)

The role of software architecture in the engineering of software-intensive applications has become more and more important and widespread. Dynamic software architectures are key to the design, development and evolution of large applications. The purpose of the workshop is to bring together researchers and practitioners from academia and industry who are interested in software architecture technology. It addresses both practical and theoretical advances.

**22nd - 23rd May 2004**

Program Committee Meeting for

[The ACM SIGSOFT 12th International Symposium on the Foundations of Software Engineering \(FSE-12\)](#)

**29th - 30th May 2004**

Program Committee Meeting for

[The 19th IEEE International Conference on Automated Software Engineering 2004 \(ASE 2004\)](#)

## Sponsors, Donors and Supporters

ICSE 2004 is being organised under the auspices of the *Institution of Electrical Engineers (IEE)* and is supported by the *British Computer Society (BCS)*. The *ACM Special Interest Group on Software Engineering (SIGSOFT)* and the *IEEE Computer Society Technical Council on Software Engineering* are the permanent sponsors of ICSE. ICSE 2004 is also grateful for the generous support of *Northrop Grumman*, the *Nokia Corporation*, *Microsoft Research* and *Zühlke Engineering*.

