Westin Harbour Castle Hotel

All paper sessions, tutorials, workshops, and exhibits are held entirely within the Westin Harbour Castle Hotel Complex and Conference Center. This beautiful hotel property is situated directly on the lake front overlooking Lake Ontario, and provides ample meeting space for the entire conference. The hotel property is also located just a few blocks away from the heart of downtown Toronto, which abounds in restaurants, theaters, shopping, and other activities. There is also very convenient and efficient public transportation within a few blocks of the hotel which places the entire city within easy reach.

All guest rooms feature individual climate control, two telephones, remote control cable TV with free and pay-per-view movies, minibars, and many other amenities. Smoking and non-smoking rooms are available. The hotel also has excellent indoor recreational facilities, including a pool, sauna, and exercise room. It also features three fine restaurants. The Toulà is a newly opened, fine Italian restaurant on the 38th floor atop the south tower of the hotel, which provides a panoramic view of the city and lake. The Mizzen offers first class service and opens seven days a week for breakfast, lunch, and dinner. The Chartroom Bar and Lounge are located adjacent to the main lobby.

Westin Harbour Castle Hotel
One Harbour Square
Toronto, Ontario, Canada
M5J 1A6
Voice: +1-416-869-1600
Fax: +1-416-869-0573

Editor and Layout: Anke Weber, University of Victoria, Canada
Contributors:
Hausi A. Müller, University of Victoria, Canada; Mary Jean Harrold, Georgia Institute of Technology, USA; Wilhelm Schäfer, University of Paderborn, Germany; Anke Weber, University of Victoria, Canada; Kenny Wong, University of Alberta, Canada

Toronto photography courtesy by Kenny Wong.
ICSE 2002 and ICSE 2003 ads by Kenneth M. Anderson, University of Colorado, USA.
The ICSE 2001 Final Program was printed by the National Research Council of Canada (NRC).

We would like to thank all the people who helped to proofread this document. In particular, we would like to acknowledge several University of Victoria graduate students for their helpful comments.
Welcome to ICSE 2001 Software Engineering Week in Toronto!

Today, the engineering of software profoundly impacts world economics. For example, the desperate demands by all information technology sectors to adapt their information systems to the Web has generated a tremendous need for methods, tools, processes, and infrastructure to develop and evolve applications efficiently and cost-effectively.

ICSE 2001, the premier conference for software engineering will feature the latest inventions, achievements, and experiences in software engineering research and practice, and will give researchers, practitioners, and educators the opportunity to present, discuss, and learn. The ICSE 2001 Software Engineering Week, May 12–19, 2001 consists of the main ICSE conference and over 50 tutorials, workshops, collocated conferences, and symposia. The conference venue is the Westin Harbour Castle overlooking Lake Ontario in downtown Toronto, with restaurants, theaters, shopping, and plenty of other activities.

The main ICSE 2001 program includes 47 technical papers, eight case-study reports, six education papers, an invited industry track, nine formal research demonstrations, and four panels. The program also contains six plenary sessions with outstanding invited keynote speakers. The main ICSE 2001 program also contains two new features: Challenges and Achievements in Software Engineering (CHASE), in which each session offers both research and industrial views of the same topic; and Frontiers of Software Practice (FoSP), which provides mini-tutorials on new and promising software technologies. Throughout the conference, there are also exhibits, posters, and informal research demonstrations. Finally, the conference features three casual receptions with great food and entertainment to give all an opportunity to meet and mingle with old and new friends.

Prior to the main ICSE 2001 program, there are 22 tutorials (full day and half day) on a variety of topics and 18 workshops that offer a forum for interaction. There are also three special symposia: the David L. Parnas Symposium, the New Software Engineering Faculty Symposium, and the Doctoral Symposium. Finally, both prior to and immediately following the main ICSE 2001 program, there are four collocated conferences: International Workshop on Program Comprehension (IWPC 2001); Engineering for Human-Computer Interaction (EHCI 2001); Symposium on Software Reusability (SSR 2001); and Spin Workshop on Model Checking of Software (SPIN 2001).

We hope that you will find time during this busy Software Engineering Week to explore the beautiful city of Toronto with its restaurants, theaters, shopping, and plenty of other activities.

We cordially invite you to participate in ICSE 2001 to help us exchange ideas and experiences in this ever expanding and critical field of software engineering.

Hausi A. Müller  Mary Jean Harrold  Wilhelm Schäfer
City of Toronto

Toronto is a clean, safe, cosmopolitan city with a wonderful network of parks, recreational, and cultural facilities. Toronto is the home of four professional sports teams and the third largest English-speaking theater district in the world, behind New York and London. One of the world's most ethnically diverse cities, it is home to more than 80 ethnic communities from Africa, Asia, and Europe. Toronto is also the business center of Canada. Exciting, vibrant, and cosmopolitan, Metropolitan Toronto, with a population of approximately 2.5 million, reigns as Canada's largest city, the capital of the Province of Ontario, and Canada's cultural, financial, and transportation hub. Metro Toronto is within a one hour drive for about five million Canadians, and is 90 minutes by air for about 60 per cent of the population of the United States. Toronto is home to the CN Tower, the world's tallest free standing structure; two-time World Champion Blue Jays baseball team; SkyDome, the world's first retractable roof stadium; Ontario Place, which features the first permanent giant-screen IMAX theatre, a Canadian invention; and is in close proximity to the Niagara region featuring Niagara Falls. Please find more details at http://www.csr.uvic.ca/icse2001/toronto.html.

Entering Canada

A passport is preferred, but is not necessary, for U.S. visitors entering Canada. U.S. citizens should have their birth certificate or citizenship certificate as well as a picture ID; permanent residents (who are not citizens) need their alien-registration card and passport. Citizens from all countries other than the US must have a valid passport, and in some cases a visitor visa may be required. Canadian customs regulations apply for all personal and business travel into Canada. Most travelers may clear customs with their personal goods and belongings at the airport in Toronto. Please check your local customs regulations if you are planning on taking Canadian purchases home with you.

Air Transportation and Discount Airfare

Air Canada and many US and international carriers provide direct flights to Toronto's Lester B. Pearson International Airport (YYZ) from major cities in North and South America, Europe, the Orient, and the Pacific Rim. Conference discounted airfares are currently available with:

- Air Canada, +1-800-361-7585, convention number CV931642

Ground Transportation

The downtown Toronto hotels are approximately 13 km (8 miles) from Lester B. Pearson International Airport. The Pacific Western Airport Express bus to downtown hotels usually operates every 20 minutes and costs CAD$13.75. The fixed-rate taxi/limo fares to downtown are about CAD$36.

Currency

Money may be exchanged at a favorable rate at the Westin Harbour Castle Hotel or at nearby banks. Shops and restaurants will often accept U.S. currency, but at a less favorable exchange rate. Note, however, that all registration, short course, tours and luncheon or dinner fees paid at the conference will be collected in U.S. dollars.

Sales Taxes and Visitor Tax Refund

Most items in Ontario, except food, are subject to both a 5% Provincial Sales Tax (PST) and a 7% federal Goods and Services Tax (GST). Non-residents of Canada, however, may claim a rebate for the GST paid on accommodation and most consumer goods taken out of the country. The GST Rebate Guide for Visitors, which includes the rebate form, will be supplied in the registration package. Please ensure that you retain all your receipts so that you can claim your rebate at a Duty Free Shop as you leave the country.

More Information

Visit http://www.csr.uvic.ca/icse2001/travel.html and find out more details about all the above topics as well as many more informative hyperlinks.
Sponsoring Organizations

IEEE Computer Society
http://www.computer.org/
The IEEE Computer Society is the oldest and largest association of computer professionals in the world. It offers over 90,000 members a comprehensive program of publications, meetings, and technical and educational activities, fostering an active exchange of information, ideas, and innovation. The society is the world’s leading publisher of technical material in the computing field. No other professional or commercial organization comes close to matching the Computer Society in terms of the quality, quantity, or diversity of its publications. Headquartered in Washington, DC, the society serves its members from offices in Los Alamitos, CA; Tokyo, Japan; and Brussels, Belgium. The society is the largest technical society within the Institute of Electrical and Electronics Engineers.

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

Association for Computing Machinery
http://www.acm.org/
The Association for Computing Machinery (ACM), is the world’s oldest and largest educational and scientific computing society. With a worldwide membership of 80,000 IT practitioners and academics, the ACM is the premier forum for all those that wish to keep abreast of the latest information, trends and developments in the IT industry. ACM offers its members an unprecedented number of publications, conferences, tutorials, and special interest groups.

ACM SIGSOFT Special Interest Group on Software Engineering
http://www.acm.org/sigsoft/
ACM SIGSOFT focuses on issues relating to all aspects of software engineering, providing a forum for computing professionals from industry, government and academia to examine principles, practices, education, and new research results in software engineering. In addition to ICSE, SIGSOFT sponsors the Foundations of Software Engineering conference and a variety of one-time and on-going workshops that bring practitioners, researchers, and educators together to discuss and debate timely issues. SIGSOFT publishes a bimonthly newsletter, Software Engineering Notes, which includes articles submitted by members as well as the popular forum ”The Risks Digest”, which describes software safety mishaps and concerns.

Cooperating Organizations

ACM SIGPLAN Special Interest Group on Programming Languages
http://www.acm.org/sigplan/

Corporate Sponsors

ACD Systems
http://www.acdsystems.com/
Alberta Software Engineering Research Consortium (ASERC)
http://www.aserc.ab.ca/
Avaya Communication
http://www.avaya.com/
Bell Canada
http://www.bell.ca/
National Research Council of Canada
http://www.nrc.ca/
Cisco Systems
http://www.cisco.com/
Communications and Information Technology Ontario (CITO)
http://www.cito.ca/
IBM Toronto Laboratory
http://www.cas.ibm.ca/
Alberta Informatics Circle of Research Excellence (iCore)
http://www.icore.ca/
MacDonald Dettwiler
http://www.mda.ca/
Microsoft Research
http://www.research.microsoft.com/
Mitel Networks
http://www.mitel.com/
Nokia
Rational Software
http://www.rational.com/
Software Productivity Center Inc. (SPC)
http://www.spc.ca/
Sun Microsystems
http://www.sun.ca/

Supporting Organizations

Consortium for Software Engineering Research, Canada
http://www.cser.ca/
Carnegie Mellon Software Engineering Institute, USA
http://www.sei.cmu.edu/
Georgia Institute of Technology, USA
http://www.cc.gatech.edu/
University of Alberta, Canada
http://www.cs.ualberta.ca/
University of Paderborn, Germany
http://www.uni-paderborn.de/
University of Toronto, Canada
http://www.cs.utoronto.ca/
University of Victoria, Canada
http://www.csc.uvic.ca/
University of Waterloo, Canada
http://www.math.uwaterloo.ca/
ICSE 2001 Contact
Professor Hausi A. Müller
Department of Computer Science
University of Victoria
EOW Room 337
P.O. Box 3055 STN CSC
Victoria, BC, Canada
V8W 3P6
Fax: +1-250-721-7292
E-mail: icse2001@csr.uvic.ca

Registration
ICSE 2001 registration started on January 23, 2001. The advance registration deadline was April 12, but you can still register online, by fax using a PDF registration form, or at the conference registration desk. For more details and for downloading the PDF registration form or online registration, please visit http://www.csr.uvic.ca/icse2001/registration.html

Room Assignments
Meeting rooms have been assigned for most of the conference events. Room assignments are marked by a leading “@” (e.g. “@Frontenac”). However, meeting room assignments are subject to change. Please confirm the final room assignments at the conference.

ICSE 2001 at a Glance

<table>
<thead>
<tr>
<th>Event</th>
<th>Sun May 13</th>
<th>Mon May 14</th>
<th>Tue May 15</th>
<th>Wed May 16</th>
<th>Thu May 17</th>
<th>Fri May 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keynotes</td>
<td></td>
<td></td>
<td>@Frontenac</td>
<td>@Frontenac</td>
<td>@Frontenac</td>
<td>@Frontenac</td>
</tr>
<tr>
<td>Tutorials</td>
<td>see schedule on page 8–10</td>
<td>see schedule on page 8–10</td>
<td>see schedule on page 11–13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workshops</td>
<td>see schedule on page 8–10</td>
<td>see schedule on page 11–13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical Papers (TP)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Education Papers (EDU)</td>
<td></td>
<td></td>
<td></td>
<td>@Pier 7 &amp; 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Invited Industry Presentations (IIP)</td>
<td></td>
<td></td>
<td></td>
<td>@Frontenac</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frontiers of Software Practice (FoSP)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>see schedule on Page 11</td>
<td>see schedule on Page 13</td>
</tr>
<tr>
<td>Challenges and Achievements in SE (CHASE)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>@Queen’s Quay</td>
<td></td>
</tr>
<tr>
<td>Case Study Reports (CSR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>@Frontenac</td>
<td></td>
</tr>
<tr>
<td>Formal Demos (FD)</td>
<td></td>
<td></td>
<td></td>
<td>@Pier 7 &amp; 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Awards Presentations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>@Frontenac</td>
<td></td>
</tr>
<tr>
<td>Doctoral Symposium</td>
<td></td>
<td>@Dockside II</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>David Lorge Parnas Symposium</td>
<td></td>
<td></td>
<td>@Pier 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New SE Faculty Symposium</td>
<td></td>
<td></td>
<td>@Pier 2 &amp; 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perspectives on SE Panel (PoSE)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>@Frontenac</td>
</tr>
<tr>
<td>SE Research Agendas Panel (SERA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>@Frontenac</td>
</tr>
<tr>
<td>Speaker’s Breakfast and Ready Room</td>
<td></td>
<td></td>
<td></td>
<td>@Pier 6</td>
<td>@Pier 6</td>
<td>@Pier 6</td>
</tr>
<tr>
<td>Student's Breakfast</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>@Pier 6</td>
<td></td>
</tr>
<tr>
<td>Lunches</td>
<td></td>
<td></td>
<td>@Metro East</td>
<td>@Harbour Ballroom</td>
<td>@Harbour Ballroom</td>
<td>@Metro East provided</td>
</tr>
<tr>
<td>Receptions</td>
<td></td>
<td></td>
<td>@Harbour Ballroom, @Frontenac, Foyer, and @Pier 2 &amp; 3</td>
<td>@Harbour Ballroom, @Harbour Foyer, and @Pier 2 &amp; 3</td>
<td>@Harbour Ballroom, @Harbour Foyer, and @Pier 2 &amp; 3</td>
<td>@Frontenac</td>
</tr>
<tr>
<td>Internet Cafés</td>
<td></td>
<td></td>
<td></td>
<td>@Bay</td>
<td>@Bay and @Pier 2 &amp; 3 (see schedule on Page 40)</td>
<td></td>
</tr>
<tr>
<td>Exhibits, Posters, and Informal Demos</td>
<td></td>
<td></td>
<td></td>
<td>@Pier 2 &amp; 3 (see schedule on Page 40)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registration Desk</td>
<td></td>
<td></td>
<td></td>
<td>@CL-STF Foyer and @Frontenac Foyer alternately (see schedule on Page 8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conference Office</td>
<td></td>
<td></td>
<td></td>
<td>@Wellington</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Plenary Sessions
ICSE 2001 welcomes several outstanding plenary speakers:
- Robert Balzer
- Frederick P. Brooks, Jr.
- Linda M. Northrop
- Daniel Sabbah
- Mary Shaw
- Bernd Voigt
- David M. Weiss

Tutorials
ICSE 2001 offers 22 tutorials with a broad range of topics. The full-day or half-day tutorials provide the opportunity to gain new insights, knowledge, and skills in a broad range of areas of software engineering.

Workshops
18 workshops serve as formal forums to exchange ideas on topics in research and practice of software engineering. Workshops are offered as one-day or two-day events.

Doctoral Symposium
Ph.D. students present their research objectives, methods, and preliminary results at an early enough stage to allow useful guidance for their further work and future careers.

David Lorge Parnas Symposium (DLPS)
Come and help “Thinking Hard About Software”: A symposium in recognition of the work of David Lorge Parnas in honor of his 60th birthday.

New SE Faculty Symposium
This morning symposium brings together faculty who have survived their early years with new and junior faculty. The main goal is to share and exchange ideas on practical methods for having a successful and fulfilling academic career.

Panels
ICSE 2001 panel topics include pressing issues in theory and practice of software engineering, emerging trends, enabling technologies, and professional, organizational, and social issues associated with software engineering.
- SE Body of Knowledge Panel
- Impact Project Panel
- Perspectives on SE Panel
- SE Research Agendas Panel

Technical Papers (TP)
The main ICSE 2001 program contains 47 technical papers that report on a full range of topics, including software engineering principles, theories, techniques, tools, and empirical evaluations.

Invited Industry Presentations (IIP)
ICSE 2001 features an industrial track with invited presentations to discuss leading-edge software technology in practice.

Frontiers of Software Practice (FoSP)
The FoSP sessions, a new feature of ICSE 2001, are mini-tutorials on new and promising software technologies.

Challenges and Achievements in SE (CHASE)
The CHASE sessions, another new feature of ICSE 2001, offer both research and industrial views of the same topic.

Case Study Reports (CSR)
Case study reports give an account of a significant project, with a critical review, and summary of lessons learned. These reports provide deep insights useful for future projects.

Education Papers (EDU)
Proper software engineering education and training can significantly improve the current state of software development. ICSE 2001 themes include Ph.D. programs, distance education and training, and curriculum issues such as the SWEBOK project.

Formal Research Demos (FD)
Formal research demonstrations show research systems in action. Use the opportunity to discuss the systems with their creators.

Internet Cafés
The ICSE 2001 Internet Cafés provide central zones for meetings, discussions, and collaboration of researchers and developers from around the world.

Exhibits
ICSE 2001 provides a hall for publishers, commercial exhibits, posters, and academic research demonstrations. The commercial exhibits feature leading companies offering software engineering capabilities and tools, education and training, books and journals, and consulting services.

Collocated Events
Both prior to and immediately following the main ICSE 2001 program, there are four collocated conferences:
- IWPC 2001
- EHCI 2001
- SSR 2001
- SPIN 2001

Receptions
ICSE 2001 features three casual receptions in and around the exhibits area with great food and entertainment to give all an opportunity to meet and mingle with old and new friends.
Conference Office
If you need assistance during the conference, the ICSE 2001 organization team in the Wellington room will be happy to help you.

Registration Desk
During the conference, you can register at the Registration Desk, which is located in the Frontenac Foyer or the Convention Level South Tower Foyer (CL-STF).

<table>
<thead>
<tr>
<th>Schedule for Registration Desk</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sunday, May 13</strong></td>
<td>7:30 am–5:30 pm @CL-STF</td>
</tr>
<tr>
<td><strong>Monday, May 14</strong></td>
<td>7:30 am–5:30 pm @CL-STF</td>
</tr>
<tr>
<td><strong>Tuesday, May 15</strong></td>
<td>7:30 am–3:30 pm &amp; 3:30 pm–5:30 pm @CL-STF</td>
</tr>
<tr>
<td><strong>Wednesday, May 16</strong></td>
<td>7:30 am–5:30 pm @Frontenac Foyer</td>
</tr>
<tr>
<td><strong>Thursday, May 17</strong></td>
<td>7:30 am–5:30 pm @Frontenac Foyer</td>
</tr>
<tr>
<td><strong>Friday, May 18</strong></td>
<td>7:30 am–5:00 pm @Frontenac Foyer</td>
</tr>
</tbody>
</table>

Child Care
Custom Comfort Care Inc. is a nursing resource that offers Care in a Crunch service, childcare for conference attendees who plan to bring their dependants with them. The caregivers are from nursing and ECE professions and have been trained to handle any health emergency. Basic services for a conference includes a minimum of two trained careproviders. The Day Nurseries Act requires staffing allocations for care of children according to the ages of the children in the group. Information required from conference attendees includes: name, hotel, room number, phone number, number of children and their ages, and pool supervisor. To book care, please call +1-416-698-3356 at least 4 hours prior.

**Conference Services and Schedule**

---

Sunday, May 13, 2001

<table>
<thead>
<tr>
<th>Time</th>
<th>Activities</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:30–8:30</td>
<td>Breakfast</td>
<td>Harbour Foyer</td>
</tr>
<tr>
<td>8:30–10:00</td>
<td>Full Day Tutorials</td>
<td>Harbour Foyer</td>
</tr>
<tr>
<td></td>
<td>T1 UML for Software Engineers</td>
<td>Pier 3</td>
</tr>
<tr>
<td></td>
<td>T2 The Intertwining between Risk and Project Management</td>
<td>Harbour Foyer</td>
</tr>
<tr>
<td></td>
<td>Morning Tutorials</td>
<td>CL-STF Foyer</td>
</tr>
<tr>
<td></td>
<td>T3 Methods of Component-Based Software Engineering: Essential Concepts and Classroom Experience</td>
<td>Bay</td>
</tr>
<tr>
<td>10:00–10:30</td>
<td>Nutrition Break</td>
<td>Harbour Foyer</td>
</tr>
<tr>
<td>10:30–12:00</td>
<td>Full Day Tutorials continued</td>
<td>Harbour Foyer</td>
</tr>
<tr>
<td>12:00–2:00</td>
<td>Lunch</td>
<td>Harbour Ballroom A &amp; B</td>
</tr>
<tr>
<td>2:00–3:30</td>
<td>Full Day Tutorials continued</td>
<td>Harbour Foyer</td>
</tr>
<tr>
<td>3:30–4:00</td>
<td>Nutrition Break</td>
<td>Harbour Foyer</td>
</tr>
<tr>
<td>4:00–5:30</td>
<td>Full Day Tutorials continued</td>
<td>Harbour Foyer</td>
</tr>
</tbody>
</table>

Workshops

- W1 2nd International Workshop on Living with Inconsistency @Pier 2
- W2 4th ICSE Workshop on Software Engineering over the Internet @Queen’s Quay 2
- W3 2nd ICSE Workshop on Software Product Lines: Economics, Architectures, and Implications @Dockside IV
- W4 2nd International Workshop on Automated Program Analysis, Testing, and Verification (WAPATV) @Pier 4
- W5 Software Engineering and Mobility @Pier 9
- W6 Software Visualization @Queen’s Quay 1

---

<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:30–8:30</td>
<td>Breakfast @Harbour and Dockside Foyers</td>
</tr>
<tr>
<td>8:30–10:00</td>
<td><strong>Full Day Tutorials</strong>&lt;br&gt; T5 OPEX: A Flexible OO/COB Process for Software-Intensive Systems Development, a UML Exposition @Pier 7 &lt;br&gt; T6 Describing Software Architecture with UML @Pier 9 &lt;br&gt; T7 Software Product Lines and Software Architecture Design @Pier 8</td>
</tr>
<tr>
<td></td>
<td><strong>Morning Tutorials</strong>&lt;br&gt; T8 How to Do Inspections When There is No Time @Pier 2 &lt;br&gt; T9 Easy WinWin: A Groupware-Supported Methodology for Requirements Negotiation @Pier 3 &lt;br&gt; T10 Fundamental Concepts for Practical Software Architecture @Pier 6 &lt;br&gt; T11 Introduction to Java 2 Micro Edition (J2ME): Java in Small Things @Dockside III</td>
</tr>
<tr>
<td></td>
<td><strong>Registration Desk</strong>&lt;br&gt; Workshops W5 continued @Harbour B &lt;br&gt; W6 continued @Dockside I &lt;br&gt; W7 Generative Techniques for Product Lines @Yonge &lt;br&gt; W8 SEMINAL: Software Engineering using Metaheuristic InNoative Algorithms @Queen’s Quay 1 &lt;br&gt; W9 From Software Requirements to Architectures (STRAW 2001) @Queen’s Quay 2 &lt;br&gt; W10 3rd Intl. Workshop on Net-Centric Computing: Migrating to the Web (NCC 2001) @Pier 5 &lt;br&gt; W11 Global Aspects of Software Engineering Professionalism @Dockside IV &lt;br&gt; W12 3rd Intl. Workshop on Economics-Driven Software Engineering Research (EDSER) @Pier 4 &lt;br&gt; W13 4th ICSE Workshop on Component-Based Software Engineering: Component Certification and System Prediction @Harbour A &lt;br&gt; W14 10th Intl. Workshop on Software Configuration Management: New Practices, New Challenges, and New Boundaries (SCM 10) @Harbour C</td>
</tr>
<tr>
<td>10:00–10:30</td>
<td>Nutrition Break @Harbour and Dockside Foyers</td>
</tr>
<tr>
<td>10:30–12:00</td>
<td><strong>Full Day Tutorials</strong> continued</td>
</tr>
<tr>
<td></td>
<td><strong>Morning Tutorials</strong> continued</td>
</tr>
<tr>
<td>12:00–2:00</td>
<td>Lunch @Metro East</td>
</tr>
<tr>
<td>2:00–3:30</td>
<td><strong>Full Day Tutorials</strong> continued</td>
</tr>
<tr>
<td></td>
<td><strong>Afternoon Tutorials</strong>&lt;br&gt; T12 Improving Software Inspections by Using Reading Techniques @Pier 2 &lt;br&gt; T13 Mining Components for a Software Architecture and a Product Line: The Options Analysis for Reengineering (OAR) Method @Pier 3 &lt;br&gt; T14 HyperJ™: Multidimensional Separation of Concerns for Java™ @Pier 6 &lt;br&gt; T15 Enterprise JavaBean Architecture and Design Issues: Avoiding JavaBean Soup @Dockside III</td>
</tr>
<tr>
<td>3:30–4:00</td>
<td>Nutrition Break @Harbour and Dockside Foyers</td>
</tr>
<tr>
<td>4:00–5:30</td>
<td><strong>Full Day Tutorials</strong> continued</td>
</tr>
<tr>
<td></td>
<td><strong>Afternoon Tutorials</strong> continued</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00–10:30</td>
<td>Nutrition Break @Harbour and Dockside Foyers</td>
</tr>
<tr>
<td>10:30–12:00</td>
<td><strong>Full Day Tutorials</strong> continued</td>
</tr>
<tr>
<td>12:00–2:00</td>
<td>Lunch @Metro East</td>
</tr>
<tr>
<td>2:00–3:30</td>
<td><strong>Full Day Tutorials</strong> continued</td>
</tr>
<tr>
<td>3:30–4:00</td>
<td>Nutrition Break @Harbour and Dockside Foyers</td>
</tr>
<tr>
<td>4:00–5:30</td>
<td><strong>Full Day Tutorials</strong> continued</td>
</tr>
</tbody>
</table>

**CONFERENCE SCHEDULE**

Internet Café @Bay, 12:00 pm – 9:00 pm
<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
<th>Location</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:30–8:30</td>
<td>Breakfast</td>
<td>@Harbour and Dockside Foyers</td>
<td>Registration Desk @CL-STF Foyer</td>
</tr>
<tr>
<td>8:30–10:00</td>
<td>Full Day Tutorials&lt;br&gt;T16 State, Event, Time, and Diagram in System Modeling&lt;br&gt;T17 From UML to Java: Building a 3-Tier Architecture&lt;br&gt;T18 Designing Concurrent, Distributed, and Real-Time Applications with UML&lt;br&gt;T19 Using Transformation Systems for Software Maintenance and Reengineering&lt;br&gt;T20 Effective Software Architecture Design: From Global Analysis to UML Descriptions</td>
<td>@Pier 8&lt;br&gt;@Pier 6&lt;br&gt;@Pier 7&lt;br&gt;@Pier 8&lt;br&gt;@Pier 6</td>
<td>Workshops&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued</td>
</tr>
<tr>
<td></td>
<td>Morning Tutorials&lt;br&gt;T21 Bridging the Requirements/Design Gap in Dynamic Systems with Use Case Maps (UCMS)</td>
<td>@Pier 9</td>
<td>Symposia&lt;br&gt;Symposia continued&lt;br&gt;Symposia continued&lt;br&gt;Symposia continued&lt;br&gt;Symposia continued</td>
</tr>
<tr>
<td></td>
<td>Workshops&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued</td>
<td>@Dockside V&lt;br&gt;@Dockside II&lt;br&gt;@Dockside IV&lt;br&gt;@Queen’s Quay 2&lt;br&gt;@Dockside 1</td>
<td>A2 David Lorge Parnas Symposium (DLPS)&lt;br&gt;@Pier 4&lt;br&gt;Symposium chairs: D. Hoffman, University of Victoria, Canada; D. Weiss, Avaya Communication, USA&lt;br&gt;A3 New Software Engineering Faculty Symposium (morning only)&lt;br&gt;@Pier 2 &amp; 3&lt;br&gt;Symposium chairs: L. Osterweil, University of Massachusetts, Amherst, USA; G. Silberman, Centre for Advanced Studies, Canada; K. Wong, University of Alberta, Canada</td>
</tr>
<tr>
<td>10:00–10:30</td>
<td>Nutrition Break</td>
<td>@Harbour and Dockside Foyers</td>
<td>Registration Desk @CL-STF Foyer</td>
</tr>
<tr>
<td>10:30–12:00</td>
<td>Full Day Tutorials continued</td>
<td>Morning Tutorials continued</td>
<td>Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;workshops continued&lt;br&gt;Workshops continued</td>
</tr>
<tr>
<td>12:00–2:00</td>
<td>Lunch</td>
<td>@Harbour Ballroom</td>
<td>Registration Desk @CL-STF Foyer</td>
</tr>
<tr>
<td>2:00–3:30</td>
<td>Full Day Tutorials continued</td>
<td>Afternoon Tutorials continued</td>
<td>Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued</td>
</tr>
<tr>
<td>3:30–4:00</td>
<td>Nutrition Break</td>
<td>@Harbour and Dockside Foyers</td>
<td>Registration Desk @Frontenac Foyer</td>
</tr>
<tr>
<td>4:00–5:30</td>
<td>Full Day Tutorials continued</td>
<td>Afternoon Tutorials continued</td>
<td>Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued&lt;br&gt;Workshops continued</td>
</tr>
<tr>
<td>5:30–5:45</td>
<td>Break</td>
<td>@Frontenac Foyer</td>
<td>Registration Desk @Frontenac Foyer</td>
</tr>
<tr>
<td>5:45–7:00</td>
<td>S1 Closing for David L. Parnas Symposium (Open to all ICSE attendees)</td>
<td>@Harbour Ballroom</td>
<td>Registration Desk @Frontenac Foyer</td>
</tr>
<tr>
<td></td>
<td>Software Fundamentals: The Ideas of David L. Parnas&lt;br&gt;&quot;Diogenes, Where Are You?&quot;&lt;br&gt;Frederick P. Brooks, Jr., University of North Carolina, Chapel Hill, USA</td>
<td>Session chair: D. Hoffman, University of Victoria, Canada</td>
<td></td>
</tr>
<tr>
<td>7:00–9:00</td>
<td>S2 Reception – sponsored by ACD Systems, Canada</td>
<td>@Harbour Ballroom, @Harbour Foyer, and @Pier 2 &amp; 3</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Event</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:30–8:30</td>
<td>Breakfast @Pier and @Frontenac Foyers</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Student’s Breakfast @Harbour Ballroom B</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Speaker’s Breakfast @Pier 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Registration Desk @Frontenac Foyer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:30–10:00</td>
<td><strong>S3 Welcome and Keynote</strong> @Frontenac</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Engineering and the Internet</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Daniel Sabbah, IBM Corporation, USA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session chair: H. Muller, University of Victoria, Canada</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00–10:30</td>
<td><strong>Nutrition Break</strong> @Pier and @Frontenac Foyers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:30–12:00</td>
<td><strong>S4 Technical Papers</strong> @Pier 4 &amp; 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design and Specification of Distributed Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session chair: A. Zündorf, University of Paderborn, Germany</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:00–1:10</td>
<td>Lunch @Harbour Ballroom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:10–1:50</td>
<td><strong>S8 Frontiers of Software Practice</strong> @Pier 4 &amp; 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enabling Technologies for the Future of Voice-Based Web Access</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S. Woods, Quack.com, USA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session chair: S. Tilley, University of California, Riverside, USA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:50–2:00</td>
<td><strong>Break</strong> @Pier and @Frontenac Foyers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:00–3:30</td>
<td><strong>S12 Technical Papers</strong> @Pier 4 &amp; 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Process Improvement for Multi-Site Environments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session chair: D. Berry, University of Waterloo, Canada</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3:30–4:00</td>
<td><strong>Nutrition Break</strong> @Pier and @Frontenac Foyers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4:00–5:30</td>
<td><strong>S16 Technical Papers</strong> @Pier 4 &amp; 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Effective Uses of Inspections</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session chair: L. Briand, Carleton University, Canada</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5:30–9:00</td>
<td><strong>S20 Reception</strong> – sponsored by IBM Canada Ltd. @Harbour Ballroom, @Harbour Foyer, and @Pier 2 &amp; 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Conference Schedule

**Thursday, May 17, 2001**

<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:30–8:30</td>
<td>Breakfast @Pier and Frontenac Foyers</td>
</tr>
<tr>
<td>8:30–10:00</td>
<td><strong>S21 Technical Papers</strong> @Pier 4&lt;br&gt;Dynamic Analysis and Testing Session chair: D. Rosenblum, University of California, Irvine, USA</td>
</tr>
<tr>
<td></td>
<td><strong>S22 Technical Papers</strong> @Pier 5&lt;br&gt;Construction of Component-Based Systems Session chair: D. Batory, University of Texas at Austin, USA</td>
</tr>
<tr>
<td></td>
<td><strong>S23 Case Study Reports</strong> @Frontenac&lt;br&gt;Infrastructure Support Session chair: J. Kramer, Imperial College, UK</td>
</tr>
<tr>
<td></td>
<td><strong>S24 Challenges and Achievements in SE</strong> @Queen's Quay&lt;br&gt;Process-Centered Software Engineering Environments R. Balzer, Teknowledge Corporation, USA; and V. Crum, University of Dortmund, Germany Session chairs: W. Schäfer, C. Engels, University of Paderborn, Germany</td>
</tr>
<tr>
<td></td>
<td><strong>S25 Formal Demos</strong> @Pier 7 &amp; 8&lt;br&gt;Software Architecture Session chair: R. Keller, University of Montréal, Canada</td>
</tr>
<tr>
<td>10:00–10:30</td>
<td>Nutrition Break @Frontenac Foyer</td>
</tr>
<tr>
<td>10:30–11:30</td>
<td><strong>S26 Keynote</strong> @Frontenac&lt;br&gt;The Coming-of-Age of Software Architecture Research Mary Shaw, Carnegie Mellon University, USA Session chair: M. J. Harrold, Georgia Institute of Technology, USA</td>
</tr>
<tr>
<td>11:30–12:00</td>
<td><strong>S27 ACM SIGSOFT and IEEE TCSE Award: Most Influential Paper from ICSE</strong> @Frontenac&lt;br&gt;&quot;Tolerating Inconsistency&quot; Revisited Robert Balzer, Teknowledge Corporation, USA Session chairs: M. J. Harrold, Georgia Institute of Technology, USA; W. Schäfer, University of Paderborn, Germany</td>
</tr>
<tr>
<td>12:00–1:10</td>
<td>Lunch @Metro East</td>
</tr>
<tr>
<td>1:10–1:50</td>
<td><strong>S28 Awards Presentations:</strong> ACM Service, ACM Research, IEEE Computer Society Harlan D. Mills Award @Frontenac&lt;br&gt;Session chairs: ACM SIGSOFT and IEEE TCSE Chairs</td>
</tr>
<tr>
<td>1:50–2:00</td>
<td>Break @Pier and Frontenac Foyers</td>
</tr>
<tr>
<td>2:00–3:30</td>
<td><strong>S29 Technical Papers</strong> @Pier 4&lt;br&gt;Reengineering and Software Evolution Session chair: J. Johnke, University of Victoria, Canada</td>
</tr>
<tr>
<td></td>
<td><strong>S30 Technical Papers</strong> @Pier 5&lt;br&gt;Analysis of Architectures Session chair: N. Medvidovic, University of Southern California, USA</td>
</tr>
<tr>
<td></td>
<td><strong>S31 Case Study Reports</strong> @Frontenac&lt;br&gt;Applications of New Paradigms in Software Development Session chair: D. Smith, Carnegie Mellon Software Engineering Institute, USA</td>
</tr>
<tr>
<td></td>
<td><strong>S32 Challenges and Achievements in SE</strong> @Queen's Quay&lt;br&gt;Specification and Modeling M. Boy, University of Munich, Germany, and B. Selle, Rational Inc., Canada Session chairs: W. Schäfer, C. Engels, University of Paderborn, Germany</td>
</tr>
<tr>
<td></td>
<td><strong>S33 Formal Demos</strong> @Pier 7 &amp; 8&lt;br&gt;Reuse and Integration Session chair: C. Heinemann, Worcester Polytechnic Institute, USA</td>
</tr>
<tr>
<td>3:30–4:00</td>
<td>Nutrition Break @Pier and Frontenac Foyers</td>
</tr>
<tr>
<td>4:00–5:30</td>
<td><strong>S34 Technical Papers</strong> @Pier 4&lt;br&gt;Improving the Testing Process Session chair: K. Inoue, Osaka University, Japan</td>
</tr>
<tr>
<td></td>
<td><strong>S35 Technical Papers</strong> @Pier 5&lt;br&gt;Mobile Agents Session chair: I. Ben-Shaul, Technion-Israel Institute of Technology, VersEdge Technologies, Israel</td>
</tr>
<tr>
<td></td>
<td><strong>S36 Case Study Reports</strong> @Frontenac&lt;br&gt;Software Evolution Session chair: T. Lethbridge, University of Ottawa, Canada</td>
</tr>
<tr>
<td></td>
<td><strong>S37 Challenges and Achievements in SE</strong> @Queen's Quay&lt;br&gt;Impact Project Panel Session chair: L. Osterweil, University of Massachusetts, Amherst, USA</td>
</tr>
<tr>
<td></td>
<td><strong>S38 Formal Demos</strong> @Pier 7 &amp; 8&lt;br&gt;Verification and Maintenance Session chair: E. Stouila, University of Alberta, Canada</td>
</tr>
<tr>
<td>7:00–10:00</td>
<td><strong>S39 Reception – sponsored by National Research Council of Canada (NRC)</strong> @Frontenac</td>
</tr>
</tbody>
</table>
## Conference Schedule

### Friday, May 18, 2001

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Location</th>
<th>Speaker’s Room</th>
<th>Registration Desk</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:30–8:30</td>
<td>Breakfast</td>
<td>@Pier and @Dockside Foyers</td>
<td>@Pier 6</td>
<td>@Frontenac Foyer</td>
</tr>
<tr>
<td>8:30–10:30</td>
<td><strong>$40 Technical Papers</strong> @Pier 4 &amp; 5</td>
<td>@Queen’s Quay</td>
<td>Formal Frameworks</td>
<td><strong>$42 Perspectives on SE Panel (PoSE)</strong> @Frontenac</td>
</tr>
<tr>
<td></td>
<td>Analysis of Requirements</td>
<td>Session chair: P. Tang, IBM T.J. Watson Research Center, USA</td>
<td>Session chair: K. Futate, JAIST, Japan</td>
<td>The Future of Software Engineering Panel chair: D. Notkin, University of Washington, USA</td>
</tr>
<tr>
<td>10:30–11:00</td>
<td>Nutrition Break</td>
<td>@Frontenac Foyer</td>
<td>Speaker’s Room</td>
<td>Registration Desk</td>
</tr>
<tr>
<td>11:00–12:00</td>
<td><strong>$43 Keynote</strong></td>
<td>@Frontenac</td>
<td>Speaker’s Room</td>
<td>Registration Desk</td>
</tr>
<tr>
<td></td>
<td>Software Engineering Challenges: A CIO’s Perspective</td>
<td>Session chair: W. Schäfer, University of Paderborn, Germany</td>
<td>Registration Desk</td>
<td>Registration Desk</td>
</tr>
<tr>
<td>12:00–1:10</td>
<td>Lunch provided</td>
<td></td>
<td>Registration Desk</td>
<td>Registration Desk</td>
</tr>
<tr>
<td>1:10–1:50</td>
<td><strong>$44 Frontiers of Software Practice @Pier 4 &amp; 5</strong></td>
<td>@Queen’s Quay</td>
<td>Dependability of Embedded Systems</td>
<td><strong>$46 Frontiers of Software Practice @Frontenac</strong></td>
</tr>
<tr>
<td></td>
<td>Anti-Patterns in Software Architecture</td>
<td>I. Knight, University of Virginia, USA</td>
<td>Session chair: A. Kark, National Research Council, Canada</td>
<td>Inter-Language Object Sharing with the Common Language Runtime: Infrastructure for MS .NET Panel chair: H. Müller, University of Victoria, Canada</td>
</tr>
<tr>
<td></td>
<td>T. Mohay, World Wide Institute of Software Architects (WWISA), USA</td>
<td>Session chair: M. A. Storoy, University of Victoria, Canada</td>
<td>Session chair: P. Sorensen, University of Alberta, Canada</td>
<td>Corporation, USA</td>
</tr>
<tr>
<td>1:50–2:00</td>
<td>Break</td>
<td>@Pier and @Frontenac Foyers</td>
<td>Speaker’s Room</td>
<td>Registration Desk</td>
</tr>
<tr>
<td>2:00–3:00</td>
<td><strong>$48 Keynote</strong></td>
<td>@Frontenac</td>
<td>Speaker’s Room</td>
<td>Registration Desk</td>
</tr>
<tr>
<td></td>
<td>Reuse That Pays</td>
<td>Linda M. Northrop, Carnegie Mellon Software Engineering Institute, USA</td>
<td>Registration Desk</td>
<td>Registration Desk</td>
</tr>
<tr>
<td>3:00–3:15</td>
<td>Nutrition Break</td>
<td>@Frontenac Foyer</td>
<td>Speaker’s Room</td>
<td>Registration Desk</td>
</tr>
<tr>
<td>3:15–4:45</td>
<td><strong>$49 Technical Papers</strong> @Queen’s Quay</td>
<td>Architecture for Emerging Applications</td>
<td>Panel chair: D. Rombach, Fraunhofer IESE, Germany</td>
<td><strong>$50 SE Research Agendas Panel (SERA)</strong> @Frontenac</td>
</tr>
<tr>
<td></td>
<td>Session chair: E. Di Nito, Politecnico di Milano, Italy</td>
<td></td>
<td>Session chair: W. Tracz, Lockheed Martin Federal Systems, USA</td>
<td></td>
</tr>
<tr>
<td>4:45–5:00</td>
<td><strong>$51 ICSE 2001 Closing</strong></td>
<td>@Frontenac</td>
<td>Registration Desk</td>
<td>Registration Desk</td>
</tr>
</tbody>
</table>

### Exhibits, Informal Demos, and Posters

- Internet Café @Pier 2 & 3, 7:30 am–2:00 pm
- Bay, 7:30 am–2:00 pm
- Pier 2 & 3, 3:30–7:30 pm
Daniel Sabbah  
Vice President Application and Integration Middleware Division  
IBM Corporation, USA  
Session S3 @Frontenac  
“Software Engineering and the Internet”  
Wednesday, May 16, 8:30 am–10:00 am  
Photo courtesy of IBM

Dr. Sabbah is the Vice President, Development, Application and Integration Middleware (AIM) Division of IBM. He manages software development for all application server, messaging, and development tools in IBM’s portfolio. This group spans over 9 locations worldwide with over 4000 software developers that produce the WebSphere platform, CICS, all MQ Series products and the VisualAge tools across a base of 9 or more operating system platforms. Prior to that, he was vice president architecture and tools development where he was responsible for the architecture and strategy for IBM’s application development tools as well as IBM’s overall web application servers and business integration software. He began his career at IBM in 1974 in telecommunications software (VTAM) in Kingston, New York. He received his Ph.D. in Computer Science from the University of Rochester in 1981, specializing in artificial intelligence and computer vision. He returned to the IBM Research Division and was responsible for the artificial intelligence effort, then programming languages, and finally for software technology. Dr. Sabbah has direct experience in both product development and in software research.

Mary Shaw  
Alan J. Perlis Professor of Computer Science  
Carnegie Mellon University, USA  
http://www.cs.cmu.edu/~shaw/  
Session S26 @Frontenac  
“The Coming-of-Age of Software Architecture Research”  
Thursday, May 17, 10:30 am–11:30 am  
Photo courtesy of Carnegie Mellon University

Mary Shaw is the Alan J. Perlis Professor of Computer Science at Carnegie Mellon University. She has been a member of this faculty since completing the Ph.D. degree at Carnegie-Mellon in 1972. She had previously earned a B.A (cum laude) from Rice University and worked in systems programming and research at the Research Analysis Corporation and Rice University. Her research interests in computer science lie primarily in the areas of programming systems and software engineering, particularly software architecture, programming languages, specifications, and abstraction techniques. She has participated in developing innovative curricula in Computer Science from the introductory to the doctoral level. Dr. Shaw has received the Warnier prize for contributions to software engineering and is a Fellow of the Association for Computing Machinery, the Institute for Electrical and Electronics Engineers, and the American Association for the Advancement of Science.

Bernd Voigt  
Senior Vice President and  
Chief Information Officer  
Lufthansa, Germany  
Session S43 @Frontenac  
“Software Engineering Challenges: A CIO’s Perspective”  
Friday, May 18, 11:00 am–12:00 pm  
Photo courtesy of Lufthansa

Before joining Lufthansa in 1992, Bernd Voigt was doing research and teaching mathematics at various German universities. At Lufthansa, he directed the Frankfurt office of Lufthansa Informationstechnik und Software GmbH Berlin, and formed the new Competence Center Decision Support Technology at Lufthansa Systems GmbH. In 1995, he founded in Budapest, Hungary, the Lufthansa subsidiary Lufthansa Systems Hungary Kft. From 1996 to 1998, he was managing director of Lufthansa Systems Berlin GmbH. Since the beginning of 1999, he has been Senior Vice President and Chief Information Officer of Lufthansa German Airlines.

Linda M. Northrop  
Director, Product Line Systems Program  
Carnegie Mellon Software Engineering Institute, USA  
http://www.sei.cmu.edu/staff/lmn/  
Session S48 @Frontenac  
“Reuse That Pays”  
Friday, May 18, 2:00 pm–3:00 pm  
Photo courtesy of the Carnegie Mellon Software Engineering Institute

Linda Northrop has over 30 years of experience in the software development field as practitioner, manager, consultant, and educator. She is currently director of the Product Line Systems Program at the Software Engineering Institute (SEI). The Product Line Systems Program works in the areas of software architecture, reengineering, component and product line engineering. Her current publications are in the areas of software product lines, software architecture, and object technology. She is a frequently invited speaker at technical conferences and was featured in a television special on object technology aired by the British Broadcasting Company. Before joining the SEI, she was associated with both the United States Air Force Academy and the State University of New York as professor of computer science and with both Eastman Kodak and IBM as a software engineer. As a private consultant, Linda also worked for an assortment of companies covering a wide range of software systems. She recently chaired the first Software Product Line Conference. Linda is an eight-year member of the OOPSLA Organizing Committee, was OOPSLA ’99 Technical Program Chair, and is OOPSLA ’2001 Conference Chair. She is also a member of the ACM and the IEEE Computer Society, and the Computer Science Accreditation Commission.
Technology Research at Avaya Laboratories, where he conducts and guides research into ways of improving the effectiveness of software development. Formerly he was Director of Software Production Research at Bell Laboratories. He has also served as CTO of PaceLine Technologies and as Director of Reuse and Measurement at the Software Productivity Consortium. At the Congressional Office of Technology Assessment he was co-author of an assessment of the Strategic Defense Initiative, and he was a visiting scholar at The Wang Institute. He originated the CQM approach to software measurement as his PhD thesis at the University of Maryland, was a member of the A-7 project at the Naval Research Laboratory, and devised the FAST process for product-line engineering. He has also worked as a programmer and a mathematician.

Frederick P. Brooks, Jr., is Kenan Professor of Computer Science at the University of North Carolina at Chapel Hill. He was an architect of the IBM Stretch and Harvest computers. He was corporate project Manager for the System/360, including development of the System/360 computer family hardware, and the Operating System/360 software. He founded the Department of Computer Science, University of North Carolina, in 1964 and chaired it for 20 years. His research includes computer architecture, software engineering, and interactive 3-D computer graphics (virtual reality). His best-known books are The Mythical Man-Month: Essays on Software Engineering, and Computer Architecture: Concepts and Evolution (with G.A. Blaauw, 1997). Dr. Brooks has received the National Medal of Technology, the Bower Award and Prize of the Franklin Institute, the John von Neumann Medal of the IEEE, and the Allen Newell and Distinguished service awards of the ACM.
**Full-Day Tutorials (8:30 am–5:30 pm)**

### T1 @Pier 3
**UML for Software Engineers**  
Robert France, Colorado State University, USA; and Cris Kobryn, Telelogic, Inc., USA; france@cs.colostate.edu  
This tutorial assists participants in developing an understanding of UML with respect to its use for modeling software requirements and designs. The tutorial targets attendees familiar with basic object-oriented modeling principles and concepts. It covers both basic and some advanced features of the UML and gives an overview of efforts directed at restructuring UML to support the view of UML as a family of languages. A roadmap outlining the planned evolution of the UML is presented.

### T2 @Yonge
**The Intertwining between Risk and Project Management**  
Karol Frühauf, INFOGEM AG, Switzerland; Karol_Fruehauf@compuserve.com  
This is a tutorial from a practitioner for software practitioners, especially software project leaders, software managers and software quality engineers. The participants should have some project experience in order to benefit fully from the material. Familiarity with process models for software development and with project management techniques is advantageous. Participants will learn how to integrate risk assessment principles and techniques with software project planning and controlling. The fundamental problems of software project management such that what is a project (and not a process or a product) and what are the tasks of project management in this context are discussed. A thorough, practical approach to risk management presented. The main focus is the intertwining of project and risk management into a coherent planning and controlling process.

### T4 @Bay
**From Use Cases to Code—Rigorous Software Development with UML**  
Albert Zündorf, University of Paderborn, Germany; zuendorf@uni-paderborn.de  
The Rational Unified Process lacks technical guidance for the development of OO applications. This tutorial fills this gap. We first use UML scenario diagrams to analyze use-cases. Next, we show a method to analyze scenarios and to derive UML class diagrams and UML behavior modeling for active classes and methods. We show how to choose and embed design patterns in a design and how to employ different architectural styles. From such a precise design, smart CASE tools generate fully functional implementations. We explain state-of-the-art code generation concepts for UML and assess current CASE tools for their code generation capabilities and for their support through all software development phases more generally.

### At a Glance

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:30</td>
<td>Breakfast</td>
</tr>
<tr>
<td>8:30–10:00</td>
<td>Full Day Tutorials</td>
</tr>
<tr>
<td>10:30–12:00</td>
<td>Full Day Tutorials continued</td>
</tr>
<tr>
<td>2:00–3:30</td>
<td>Full Day Tutorials continued</td>
</tr>
<tr>
<td>4:00–5:30</td>
<td>Full Day Tutorials continued</td>
</tr>
</tbody>
</table>

**Morning Tutorial (8:30 am–12:00 pm)**

### T3 @Bay
**Methods of Component-Based Software Engineering: Essential Concepts and Classroom Experience**  
Kurt Wallnau, Carnegie Mellon Software Engineering Institute, USA; kcw@sei.cmu.edu  
The need for component-based development methods has become more pronounced as industry adopts software component technology. However, the idea of component is not monolithic; different forms of a software component introduce different engineering challenges. This tutorial provides an overview of “component space” and the methods that populate this space. Two classes of component-based design methods are discussed in detail: one where the task is to specify component interfaces, the other where components exist a priori from a commercial provider and must be integrated. The tutorial also describes a CMU Master of Software Engineering course focused on component-based methods and taught by the presenter. The design of team projects and their results are presented, along with lessons learned that may prove useful to other instructors.

**Afternoon Tutorial (2:00 pm–5:30 pm)**

### T4 @Bay
**From Use Cases to Code—Rigorous Software Development with UML**  
Albert Zündorf, University of Paderborn, Germany; zuendorf@uni-paderborn.de  
The Rational Unified Process lacks technical guidance for the development of OO applications. This tutorial fills this gap. We first use UML scenario diagrams to analyze use-cases. Next, we show a method to analyze scenarios and to derive UML class diagrams and UML behavior modeling for active classes and methods. We show how to choose and embed design patterns in a design and how to employ different architectural styles. From such a precise design, smart CASE tools generate fully functional implementations. We explain state-of-the-art code generation concepts for UML and assess current CASE tools for their code generation capabilities and for their support through all software development phases more generally.
In software engineering, there has long been a recognition that inconsistency is a fact of life. Evolving descriptions of software artefacts are frequently inconsistent, and tolerating this inconsistency is important if flexible collaborative working is to be supported. This workshop will focus on reasoning in the presence of inconsistency, for a wide range of software engineering activities, such as checking, theorem proving, logic programming, and model-based reasoning can still be applied in the presence of inconsistency.

A product line is a set of products designed within a single strategy to exploit mutual synergies. Especially in areas where the competition is very high, companies adhering to a product line approach can benefit from cost reduction, decreased time-to-market, and quality improvement. These and other advantages often lead to better results in the market than the development of individual products alone. Following the remarkable success of the "First International Workshop on Software product lines: Economics, architectures, and implications" held at ICSE 2000 in Limerick, the objective of this workshop is to bring together again people from industry and academia to investigate the proposals and the practices that pertain to the domain of software product lines.
2-Day Workshops (Sunday–Monday; 8:30 am–5:30 pm)

W5 Sunday @Pier 9; Monday @Harbour B
Software Engineering and Mobility
Gruia-Catalin Roman, Washington University, St. Louis, USA; and Gian Pietro Picco, Politecnico di Milano, Italy
Contact: G.-C. Roman; roman@cs.wustl.edu; http://www.elet.polimi.it/~picco/icse01mobility/

Mobility is redefining the hardware and software fabric of distributed systems. Wireless communication allows network hosts to participate in a distributed computation while on the move. Novel middleware technologies allow software components to migrate across hosts for enhanced flexibility or performance. The software engineering implications of this wave of technological changes still await a thorough understanding. Researchers are invited to discuss fundamental models, emerging themes, research opportunities, technological trends, and market forces in the field of mobile computing and communication. The immediate objective is to provide a forum for intellectual debate. The ultimate goal is to define an influential research agenda for the area as a whole and to generate advocacy for it by stimulating new research initiatives.

W6 Sunday @Queen's Quay 1; Monday @Dockside I
Software Visualization
Wim De Pauw, IBM T.J. Watson Research Center, USA; Steven P. Reiss, Brown University, USA; and John T. Stasko, Georgia Institute of Technology, USA
Contact: Steven P. Reiss; spr@cs.brown.edu; http://www.cs.brown.edu/research/softvis/icse2001.html

This workshop will look at current work in the area of software visualization with an emphasis on software understanding through visualization. It will explore new visualization techniques, addressing software problems through visualization, frameworks for gathering and analyzing data for software visualization, software visualization systems, and experiments and experiences with software visualization. In addition to providing an overview of current research in the area, it will provide a forum for discussions and cooperation among researchers in this and related areas.
At a Glance

7:30  Breakfast  
   @Harbour and  
   @Dockside Foyers

8:30–
10:00  Full Day Tutorials  
   T5–T7  
   Morning Tutorials  
   T8–T11  
   Workshops  
   W5, W6 continued  
   W7–W14  
   Symposia  
   A1  
   Nutrition Break  
   @Harbour and  
   @Dockside Foyers

10:30–
12:00  Full Day Tutorials continued  
   Morning Tutorials continued  
   Workshops continued  
   Symposia continued  
   Lunch  
   @Metro East

2:00–
3:30  Full Day Tutorials continued  
   Afternoon Tutorials  
   T12–T15  
   Workshops continued  
   Symposia continued  
   Nutrition Break  
   @Harbour and  
   @Dockside Foyers

4:00–
5:30  Full Day Tutorials continued  
   Afternoon Tutorials continued  
   Workshops continued  
   Symposia continued

Full-Day Symposium (8:30 am–5:30 pm)

T5 @Pier 7
Houman Younesi, Rensselaer Polytechnic Institute at Hartford, USA; and Brian Henderson-Sellers, Swinburn University of Technology, Australia; houman@rh.edu

The increased complexity associated with large-scale software-intensive systems development requires an increase in the sophistication of the methodology utilized. Following a general discussion on the value of processes, one specific OO/CBD example, OPEN (Object-oriented Process, Environment and Notation) is described in detail. Emphasis will be placed upon the need for flexibility of processes and how they can be constructed and configured to individual circumstances. Some advice on how to transition to OO/CBD and deploy this process for the first time will be given. In addition, and using OPEN as the background process, the UML (Unified Modeling Language) is introduced through a number of case study examples. Modeling and notational issues of the UML are discussed at an introductory level, set in the relevant context of OPEN’s modeling tasks and techniques.

T6 @Pier 9
Describing Software Architecture with UML
Phillipe Kruchten and Bran Selic, Rational Software Canada Corp., Canada; and Wojtek Kozaczyinski, Rational Software Inc., USA; pbk@rational.com

The presence of a solid architectural vision is a key discriminator in the success or failure of a software project. This tutorial examines what software architecture is and what it is not. It discusses and illustrates how to describe architecture through a set of design viewpoints and views and how to express these views in the UML, in the spirit of the new IEEE Standard 1471:2000: recommended practice for architectural description. The tutorial shows how architectures drive the development process and how to capture architectural design patterns using the UML. It is illustrated by several widely applicable architectural patterns in different domains.

T7 @Pier 8
Software Product Lines and Software Architecture Design
Jan Bosch, University of Groningen, The Netherlands; Jan.Bosch@cs.rug.nl

One can identify two important developments in software architecture, i.e. software product lines and software architecture design. This tutorial addresses these topics by presenting a method for architectural design explicitly focussing on assessment of and transformation for quality attributes. In addition, the notion of software product lines and the process, technology, business and organizational issues of adopting the approach are presented. Examples and experiences from numerous industrial cases are used to extensively illustrate both design and product-lines. This intermediate-level tutorial is based on a recently published book entitled Design and Use of Software Architectures: Adopting and Evolving a Product Line Approach, authored by the presenter, and published by Addison-Wesley. The expected audience is of two types: software engineers and technical managers considering the introduction of software architecture design and, potentially, product-line architectures in their organization, and researchers interested in the experiences of the presenter on this subject.
Morning Tutorials (8:30 am–12:00 pm)

T8 @Pier 2
How to Do Inspections When There Is No Time
Terry Shepard and Diane Kelly, Royal Military College, Canada; shepard@rmc.ca

Research shows that software inspections are an effective, essential part of software development. Yet, according to some industry practitioners, inspections are difficult, costly, ineffective, and excessively time consuming. So what’s gone wrong? The most likely answer is extreme pressure on resources, and above all, on schedules. Organized around the 3Ms of inspection, Management, Mechanics, and Metrics, this tutorial provides background information for the novice and discusses new material of interest to both novices and experienced inspection practitioners. It will include a case study and an analysis of how the case study can be adapted to other situations. Opportunities for questions and discussions will be provided throughout.

T9 @Pier 3
EasyWinWin: A Groupware-Supported Methodology For Requirements Negotiation
Barry Boehm, University of Southern California, USA; Paul Grünbacher, Johannes Kepler University, Austria; and Robert O. Briggs, GroupSystems.com, USA; pg@sea.uni-linz.ac.at

EasyWinWin is a requirements definition methodology that builds on the win-win negotiation approach and leverages collaborative technology to improve the involvement and interaction of key stakeholders. With EasyWinWin, the stakeholders move through a step-by-step win-win negotiation where they collect, elaborate, and prioritize their requirements, and then surface and resolve issues. This tutorial introduces the EasyWinWin negotiation approach and situates it with respect to other leading requirements determination approaches, and within the spiral model of software development. We explain the objectives and deliverables of each step in the methodology, and offer tips and pitfalls from the field. We give a live demonstration of the collaborative tools and the methodology in action, and demonstrate facilitation techniques that keep the process moving forward. Throughout the tutorial, we will present the highlights from several real-world EasyWinWin projects. The intended audience are those professionals involved in requirements definition (e.g., as project managers, engineers, executives, users, customers); and software requirements engineering researchers. The level of presentation is introductory with no specific background knowledge required.

T10 @Pier 6
Fundamental Concepts for Practical Software Architecture
Alexander Ran, Nokia Research Center, USA; alexander.ran@nokia.com

Architecture of software is a collection of design decisions that are expensive to change. How can we identify which design decisions are expensive to change? What are architecture views and which views are needed to adequately describe the architecture of a specific system? How can we create and manage software architecture for a product family? This tutorial will provide an answer to these and other questions that arise in the context of complex software development. We introduce a system of concepts useful to understand, design, and evaluate architecture of software intensive systems and system families. Our approach utilizes different software structures in order to control important system qualities related to its development, performance, and evolution. We draw our experience primarily from software embedded in voice and data communication systems. However the same principles can be applied to software architecture in other domains. This tutorial should be useful to software engineers and technical managers involved in construction or evaluation of complex software.

T11 @Dockside III
An Introduction to Java 2 Micro Edition (J2ME): Java in Small Things
James White, Catapult Technologies, Inc., USA; jwhite@catapult-technologies.com

This tutorial is intended to be an introduction to the Java 2 Micro Edition (J2ME) for software engineers. J2ME is Sun's latest Java 2 platform and is meant for consumer electronics and embedded devices. An exploration of the technology as well as an introduction to programming in the J2ME environment is undertaken. The tutorial is divided into two major sections. The first section provides a general understanding of J2ME. Included in this half are discussions on J2ME's reason for being, architecture, and configurations/profiles --the specifications that really define J2ME. The second half focuses on developing applications in the various J2ME configurations/profiles. The various J2ME APIs are compared and contrasted with programming examples to the more familiar Java 2 Standard Edition (J2SE). To understand this tutorial, some knowledge of Java programming is helpful but not required.
This tutorial discusses the problem of identifying components from legacy systems and determining which components will be useful for insertion in a new architecture, particularly in a product line architecture. Specifically, it outlines Options Analysis for Reengineering (OAR), a systematic method for evaluating the feasibility and benefits of mining existing components for a product line. OAR operates like a funnel in which a large set of potential assets is screened out so that the effort can most efficiently focus on a smaller set that most effectively meet the technical and programmatic needs of the product line or target architecture. The method provides a set of scalable techniques and exercises to collaboratively analyze existing components, determine viable mining options, and select the most promising option. It provides a structured approach to determine the cost, effort, and risk of mining a set of components from legacy systems. The tutorial is appropriate for researchers and advanced practitioners who are faced with making decisions on mining legacy components.

Hyper/J™: Multi-Dimensional Separation of Concerns for Java™
Peri Tarr and Harold Ossher, IBM T. J. Watson Research Center, USA; tarr@watson.ibm.com
Hyper/J™ supports a new approach to constructing, integrating and evolving software, called Multi-Dimensional Separation of Concerns (MDSOC). Developers can decompose and organize code and other artifacts according to multiple, arbitrary criteria (concerns) simultaneously—to some extent even after the software has been implemented—and synthesize or integrate the pieces into larger-scale components and systems. This tutorial describes how to accomplish MDSOC with Hyper/J™, a tool available for free download, in the context of standard Java™ development. The focus is on addressing some real, pervasive problems, especially in the areas of evolution, integration, reuse, and (re)use of design patterns, and showing how to encapsulate concerns like features, variants, roles and business rules. The tutorial is at an intermediate-level and combines presentation with interactive exploration of examples. Participants have the opportunity to select examples from a prepared set, and to contribute their own. Attendees must have experience with object-oriented software engineering, and some familiarity with Java™.

Enterprise JavaBeans (EJB) have become a staple in distributed object and component architectures. However, like most technologies, EJB provides technology tradeoffs that must be weighted by the architect contemplating its use. Furthermore, like most technologies, EJB cannot save a poor system design. This tutorial is separated into two general sessions. The first half of the tutorial covers the larger “macro” architectural decisions surrounding EJB, such as decisions about whether EJB technology is right for a project and selecting an appropriate EJB server. The second half of the tutorial focuses on issues in bean design. These “micro” issues include: where to use entity versus session beans, when to use container versus bean managed persistence, the granularity of EJBs, and the impact of the underlying database on bean design. This session is intended for architects, analyst and developers examining or working with EJB technology. Attendees are expected to have some familiarity with EJB terminology and technology.

Enterprise JavaBean Architecture and Design Issues: Avoiding JavaBean Soup
James White, Catapult Technologies, Inc., USA; jwhite@catapult-technologies.com

Concerns (MDSOC). Developers can decompose and organize code and other artifacts according to multiple, arbitrary criteria (concerns) simultaneously—to some extent even after the software has been implemented—and synthesize or integrate the pieces into larger-scale components and systems. This tutorial describes how to accomplish MDSOC with Hyper/J™, a tool available for free download, in the context of standard Java™ development. The focus is on addressing some real, pervasive problems, especially in the areas of evolution, integration, reuse, and (re)use of design patterns, and showing how to encapsulate concerns like features, variants, roles and business rules. The tutorial is at an intermediate-level and combines presentation with interactive exploration of examples. Participants have the opportunity to select examples from a prepared set, and to contribute their own. Attendees must have experience with object-oriented software engineering, and some familiarity with Java™.
Generative Techniques for Product Lines

Greg Butler, Concordia University, Canada; Don Batory, University of Texas at Austin, USA; Krzysztof Czarnecki, Daimler Chrysler Research, Germany; and Ulrich Eisenecker, University of Applied Sciences, Kaiserslautern, Germany

Contact: G. Butler, gregb@cs.concordia.ca; http://www.cs.concordia.ca/~gregb/icse-workshop/

A software product line leverages the knowledge of one or more domains in order to achieve short time-to-market, cost savings, and high quality software. The highest level of reuse comes by using domain-specific languages or visual builders to describe a member of the product line, and to generate the member from the description. Generative techniques can help us to capture the configuration knowledge for a product line and use it to generate concrete family members. This workshop focuses on technical issues of product lines, rather than economic issues.

SEMINAL: Software Engineering using Metaheuristic INnovative ALgorithms

Mark Harman, Brunel University, UK; Bryan Jones, University of Glamorgan, UK; and Nigel Tracey, University of York, UK

Contact: M. Harman, mark.harman@brunel.ac.uk; http://www.brunel.ac.uk/~csrmmmb2/seminal2001/

Metaheuristic algorithms, such as genetic algorithms and simulated annealing, have been applied successfully to a number of engineering problems ranging from load balancing in the process industries, through electromagnetic system design, to aircraft control and aerodynamics. It is surprising that these essentially software driven technologies have not yet fully penetrated the software engineering research community and are not widely applied when compared to the more traditional engineering disciplines. Software engineers often face problems associated with the balancing of competing constraints, trade-offs between concerns and requirement imprecision. Perfect solutions are often either impossible or impractical. Therefore, like other engineering disciplines, software engineering is typically concerned with near optimal solutions or those which fall within a specified tolerance. It is precisely these factors which make robust metaheuristic search-based optimization techniques readily applicable. The goal of the workshop is to broaden awareness within the software engineering community of metaheuristic algorithms and their application to software engineering problems and to bring together researchers and practitioners in software engineering and metaheuristics to build upon the embryonic community which currently occupies the intersection of the two fields.

From Software Requirements to Architectures (STRAW 2001)

Jaelson Castro, Universidade Federal de Pernambuco, Brazil; and Jeff Kramer, Imperial College, UK

Contact: J. Castro, jbc@cs.toronto.edu; http://www.cin.ufpe.br/~straw01/

Requirements Engineering and Software Architecture have become established areas of research, education and practice within the software engineering community for a considerable time. Despite the advances on both fronts we still need frameworks, techniques and tools to support the systematic achievement of architectural objectives in the context of complex stakeholders relationships. For example, little effort has been devoted to date to techniques for deriving architectural descriptions together with the requirement specifications. It also remains very difficult to show that a given software architecture satisfies a set of functional and non-functional requirements. This is somewhat surprising, as software architecture has long been recognized to have a profound impact on the achievement of non-functional goals (“ilities”) such as availability, reliability, maintainability, safety, confidentiality, evolvability, and so forth. Therefore greater effort should be devoted to bridging the gap between Requirements Engineering research and Software Architecture research.

3rd International Workshop on Net-Centric Computing: Migrating to the Web (NCC 2001)

Jens H. Jahnke, University of Victoria, Canada; Kostas Kontogiannis, University of Waterloo, Canada; Eleni Stroulia, University of Alberta, Canada; Scott R. Tilley, University of California, Riverside, USA; and Kenny Wong, University of Alberta, Canada

Contact: S. R. Tilley, stilley@cs.ucr.edu; http://mulford.cs.ucr.edu/stilley/ncc2001/

The underlying principle of Net-Centric Computing (NCC) is a distributed environment where applications and data are downloaded from servers and exchanged with peers across a network as an-as-needed basis. NCC relies on portable applications running on multiple platforms, mobile data accessed via high-speed network connections, and low-cost appliances for local processing. In keeping with the theme of “Migrating to the Web,” the 3rd International Workshop in Net-Centric Computing (NCC 2001) will focus on issues related to reengineering legacy systems for use in an NCC environment. Of particular interest are holistic techniques for Web-enabling existing applications that integrate various reengineering aspects (e.g., code, data, and user interface reengineering) into a “whole system” modernization process. The workshop will be structured around three central issues: decomposing legacy systems to identify logical components representing essential functionality, developing a new Web-enabled system using these components, and deploying the new system in an NCC environment.
W11 Monday @Dockside IV
Global Aspects of Software Engineering Professionalism
J. Barrie Thompson and Helen M. Edwards, University of Sunderland, UK
Contact: J. B. Thompson, barrie.thompson@sunderland.ac.uk; http://www.cet.sunderland.ac.uk/seis/icse2001workshop/

The workshop will provide a forum to consider the global dimensions of a Software Engineering Profession and determine the relevance and usefulness to the software engineering community of an initiative by the International Federation of Information Processing (IFIP) concerning harmonization of professional standards. The prime objectives of the workshop will be to examine:
- The current situation with regard to certification of licensing in different countries
- The drivers and constraints regarding Software Engineering Professionalism and to determine:
- The extent to which world-wide Software Engineering Professionalism is achievable (particularly in the light of IFIP’s proposals)
- Possible strategies that would assist in achieving such world-wide Software Engineering Professionalism.

W12 Monday @Pier 4; Tuesday @Dockside V
3rd International Workshop on Economics-Driven Software Engineering Research (EDSER)
Kevin J. Sullivan, University of Virginia, USA; Mary Shaw, Carnegie Mellon University, USA; Barry Boehm, University of Southern California, USA; David Notkin, University of Washington, USA; and Warren Harrison, Portland State University, USA
Contact: K. J. Sullivan, sullivan@virginia.edu; http://www.cs.virginia.edu/~sullivan/edser3/

The overall objective of the Third International Workshop on Economics-Driven Software Engineering Research (EDSER-3) is to advance the theory and the practice of software design and engineering by viewing them as value-seeking activities. The term value is construed broadly. It includes but is not limited to meaning monetary value in capital markets. Other dimensions in which value can be defined include national security, solution of major social problems, and the advance of democratic society and values. To date, the field of software economics has focused largely on cost estimation, largely ignoring benefits and their realization. We now need a theory and practice of software design and engineering that is based on modeling of, and design and dynamic management for, net value creation, including costs and benefits, risks and opportunities. Sources for theoretical insights and advances can be found in many quarters: finance, strategy, decision theory, game theory, politics, ethics, aesthetics. EDSER-3 seeks leading-edge contributions that can advance the emerging discussion in this area.

W13 Monday @Harbour A; Tuesday @Dockside II
4th ICSE Workshop on Component-Based Software Engineering: Component Certification and System Prediction
Ivica Crnkovic, Malardalens University, Sweden; Heinz Schmidt, Monash University, Australia; and Judith Stafford and Kurt Wallnau, Carnegie Mellon Software Engineering Institute, USA
Contact: J. Stafford, jas@sei.cmu.edu; http://www.sei.cmu.edu/pacc/workshop_call.html

Components play a critical role in many software systems. Such systems suffer from (1) a lack of information about component behaviour (or lack of confidence in the information that is available), and (2) an inability to determine properties of the whole system from properties of the parts. CBSE4 will bring together researchers and practitioners from the areas of component trust and certification, component technology, and software architecture to ensure that work in the areas of certification of software components and architectural analyses for prediction of system quality attributes will be mutually aware, if not mutually reinforcing. The output of the workshop will be a defined set of community model problems that reflects this intersection of interests.

W14 Monday @Harbour C; Tuesday @Queen’s Quay 2
10th International Workshop on Software Configuration Management: New Practices, New Challenges, and New Boundaries (SCM 10)
André van der Hoek, University of California, Irvine, USA
Contact: A. van der Hoek, andre@ics.uci.edu; http://www.ics.uci.edu/~andre/scm10/

The goal of SCM-10 is to bring together industrial and academic researchers from a variety of disciplines to investigate and determine the future of configuration management. SCM-10 specifically aims to step outside of its traditional boundary of just managing source code, and is intended to be a rich, open forum for discussing such issues as component-based CM, CM for Open Source projects, Web-site management, hypermedia, deployment, the relation between dynamism, run-time change, and CM, and other such relevant topics. Discussion will be fueled via several invited presentations and presentations based on materials drawn from submitted position papers.
State, Event, Time, and Diagram in System Modeling
Jin Song Dong, National University of Singapore, Singapore; dongjs@comp.nus.edu.sg

The design of complex systems requires powerful mechanisms for modeling data, state, concurrency, and real-time behavior, as well as for structuring and decomposing systems in order to control local complexity. Method integration has become a recent research trend in software specification and design. In the graphical area, many object-oriented methods have merged into one, the Unified Modeling Language (UML) which combines various diagrammatic modeling techniques to model static and dynamic aspects of software systems. Although traditional formal methods have not scale-up well, new integrated formal methods show great promise. This tutorial will present the state of the art in formal modeling techniques (state-based Object-Z and event-based Timed CSP), their integration (TCOZ), and transformation techniques from the integrated formalism to UML. An XML web environment for projecting integrated formal models to UML diagrams will also be demonstrated. The tutorial material contains both introductory and intermediate level and should be of interest to software designers, software engineers, as well as software engineering researchers and graduate students.
T17 @Pier 6
From UML to Java: Building a 3-Tier Architecture
Timothy Korson, Southern Adventist University, USA; korson@southern.edu
The successful use of object technology requires far more than simply the adoption of UML, Java, CORBA or COM. What is crucial, is knowing how to use these technologies to build commercially robust software systems. In this session the speaker draws on his experience at NASA, AT&T, IBM, and other leading companies to illustrate the pitfalls and best practices of component-based software development. The case study will explore the recent experience of the author in developing a multi-currency, multi-lingual financial application for an international organization. The focus of the tutorial will be on exploring the modeling, architecture, Java implementation, and database issues as well as the other design tradeoffs that were considered. Each design tradeoff is related to the system requirements and business goals.

T18 @Pier 7
Designing Concurrent, Distributed, and Real-Time Applications with UML
Hassan Gomaa, George Mason University, USA; hgomaa@gmu.edu
Object-oriented analysis and design of concurrent applications are described with particular emphasis on 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. To successfully design real-time and distributed applications, it is essential to blend object-oriented concepts with concurrent processing concepts. Although the Unified Modeling Language (UML) has emerged to provide a standardized notation for describing object-oriented models, for the UML notation to be effectively applied, it needs to be used in conjunction with an object-oriented analysis and design method, as presented in this tutorial. This tutorial is based on the instructor’s book: Designing Concurrent, Distributed, and Real-Time Applications with UML, Addison Wesley, 2000, http://www.aw.com/cseng/titles/0-201-65793-7. Topics covered include OO requirements, analysis, and design modeling; distributed software architecture; component-based distributed design; concurrent task structuring; design of message communication interfaces, including synchronous, asynchronous, brokered, and group communication; performance analysis of real-time designs. This tutorial should be of interest to software professionals responsible for designing, implementing, or managing the design of concurrent applications. The material is presented for an intermediate level audience who have some familiarity with object-oriented and concurrency concepts.

T19 @Yonge
Using Transformation Systems for Software Maintenance and Reengineering
Ira Baxter, Semantic Designs, Inc., USA; idbaxter@semdesigs.com
Software maintenance costs dominate software engineering costs, partly because most such engineering is done manually. Program transformation tools leverage an engineer-provided base of “transforms” (a kind of generative reuse of programming knowledge), to automate analysis, modification, and generation of software, enhancing productivity and quality over conventional methods. This tutorial provides a complete overview of Program Transformation, from theory to implementation to application. Several real transformation systems are examined, with application examples including automated detection and removal of duplicate code from large systems, and the potential for semi-automated refactoring of large object frameworks. The tutorial progresses from introductory to intermediate, but all the necessary background will be provided. Attendees need only have basic software engineering knowledge and motivating experience in modifying software. The presenter has been designing and using transformation systems for 20 years, and is the principal behind the DMS transformation system for large scale software reengineering.

T20 @Dockside III
Effective Software Architecture Design: From Global Analysis to UML Descriptions
Robert L. Nord, Daniel J. Paulish, and Dilip Soni, Siemens Corporate Research, USA; and Christine Hofmeister, Lehigh University, USA; rm@sei.cmu.edu
It is now generally accepted that separating software architecture into multiple views can help in reducing complexity and in making sound decisions about design trade-offs. Our four views are based on current practice; they are loosely coupled, and address different engineering concerns. This tutorial will teach you how global analysis can improve your design, and how to use UML to describe these four views: the conceptual, module, execution, and code architecture views. You will also learn:

- the purpose of having separate software architecture views.
- the difference between using UML for software architecture and the use of UML for designing OO implementations.
- how to apply global analysis to analyze factors that influence the architecture and to develop strategies that guide the design.
- the importance of designing for anticipated change to produce more maintainable architectures.
- how to incorporate software architecture design and its artifacts in your software process.

The tutorial is aimed at experienced software engineers, architects, project and technical managers. It is assumed that participants know the basic UML diagram types. Experience in developing models and software design is helpful.
**Bridging the Requirements/Design Gap in Dynamic Systems with Use Case Maps (UCMs)**

Daniel Amyot and Gunter Mussbacher, Mitel Corp., Canada; gunter_mussbacher@mitel.com

Use Case Maps (UCMs) concepts, the UCM notation, and how UCMs fit into the software development process are presented. UCMs help bridge the requirements/design gap and provide dynamic (run-time) refinement capabilities for scenario/structure variations often characteristic of dynamic software systems (e.g. e-commerce applications). The tutorial will show how UCMs address functional requirements expressed in use cases and performance requirements as well as high-level design and testing. Exercises are provided for the participants along with a brief demonstration of the freely available UCM Navigator tool.

The intended audience includes requirements engineers, system architects, test engineers, and software engineers with an interest in scenario-based system development and/or responsible for the development of dynamic systems. Participants will be assumed to have some experience in system modeling and be familiar with software development processes.

---

**Introduction to the Attribute Driven Design Method**

Felix Bachmann and Len Bass, Carnegie Mellon Software Engineering Institute, USA; ljb@sei.cmu.edu

The Attribute Driven Design (ADD) method is introduced. ADD is a method for designing the conceptual architecture for a product line. The conceptual architecture includes the decomposition of function for the final systems, identification of possible threads of parallelism, identification of possible physical network configurations and allocation of the functional decomposition to processors. The ADD method is based on: 1) explicit identification of quality attribute goals for a system and 2) coupling these goals to architectural mechanisms and styles that are intended to achieve those goals. It treats quality goals as first class requirements on the level of functional requirements and makes explicit the dependence of architecture on quality and business requirements. The ADD method can begin once architectural drivers (those requirements that shape the architecture) have been identified and this allows the architecture design to be carried on in parallel to the requirements process.
1-Day Workshops (8:30 am–5:30 pm)

W15 @Pier 5
XML Technologies and Software Engineering (XSE 2001)
Cecilia Mascolo, Wolfgang Emmerich, and Anthony Finkelstein, University College London, UK
Contact: C. Mascolo, c.mascolo@cs.ucl.ac.uk; http://pizza.cs.ucl.ac.uk/xse01/xse01.xml
The aim of this workshop is to bring together the efforts in the areas of software engineering with XML and engineering software with XML. The first stream addresses the use of XML and related technologies for the construction of software engineering tools and environments. The second stream focuses on the exploitation of XML for the development of new generations of distributed software architectures and middleware in order to provide new degrees of flexibility in terms of integration, security, and interoperability. The focus of the workshopstretches from pure research on XML and related technologies to applications and reports of industrial experience with mark-up languages and tools. Technologies relevant to the workshop include XML, XSL, DOM, XLink, XPath, Schema, SOAP, Biztalk, XML Query, XMI, XPL, and RDF.

W16 @Dockside IV
Describing Software Architecture with UML
Alan Brown, Catapulse, Inc., USA; Wojtek Kozaczynski, Rational Software, Inc., USA; Philippe Kruchten, Rational Software Canada Corp., Canada; and Grant Larsen, Catapulse, Inc., USA
Contact: P. Kruchten, pbk@rational.com; http://www.rational.com/events/ICSE2001/
This workshop focuses on the use the UML for representing and modeling architectures of software-intensive systems. We will look at experience reports, discuss UML subsets and patterns of use, and examine identified difficulties and limitations. How can UML support IEEE 1471? How does it map to ADLs? How can we use a model-based architectural representation fo to ADLs? How can we use a model-based architectural representation for reasoning about various architectural qualities: performance, scalability, robustness, fault-tolerance, safety, or security? We will look at extending UML via stereotypes and UML profiles to support representation of architectures and possible extensions or improvements in the future UML 2.0 to help reason about architectural issues.

W17 @Dockside I
Advanced Separation of Concerns in Software Engineering
Peri Tarr, IBM T. J. Watson Research Center, USA
Contact: P. Tarr, tarr@watson.ibm.com; http://www.research.ibm.com/hyperspace/workshops/icse2001/
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. Traditional modularization approaches have proven inadequate. Work in the growing area of advanced separation of concerns seeks to provide more powerful and flexible modularization, capable of encapsulating multiple kinds of overlapping, interacting and crosscutting concerns. This workshop is intended to bring together researchers and practitioners in this and related areas, to explore some of the many open issues.

W18 @Queen’s Quay 1
1st Workshop on Open Source Software Engineering
Joseph Feller, Brian Fitzgerald, University College Cork, Ireland; and André van der Hoek, University of California, Irvine, USA
Contact: J. Feller, jfeller@afis.ucc.ie; http://opensource.ucc.ie/icse2001/CFP.htm
Open Source Software (OSS) has recently become the focus of considerable interest, yet there remains a need for rigorous analytical inquiry into the subject. This workshop seeks to articulate OSS as an SE paradigm, addressing the requirements of OSS in terms of methodology and process, tools and enabling technologies, and human resources and project management. Format: Round-Table Discussion. Size: Maximum 40 participants. Position Papers Required. The Workshop Report will be published in a special issue of IEE Proceedings - Software on Open Source Software Engineering, and workshop participants will be encouraged to submit full research papers based on their position papers for possible inclusion in the special issue.
Closing for David L. Parnas Symposium — Open to all ICSE Attendees

Session Chair: Dan Hoffman, University of Victoria, Canada

This keynote and personal retrospective is part of and concludes the David L. Parnas Symposium, but is open to all ICSE conference attendees. David M. Weiss will deliver the keynote and then Frederick P. Brooks, Jr. will give a personal retrospective.

Software Fundamentals: The Ideas of David L. Parnas

David M. Weiss, Director of Software Technology Research, Avaya Laboratories, USA

At ICSE 2001 we are honoring the work of one of the grandmasters of our field, highlighting the fundamental ideas that David L. Parnas invented and expounded, including such ideas as information hiding, abstract interfaces, the uses relation, program families, explicit layered exception handling, and deterministic scheduling for hard real-timesystems. Do you need to understand how to organize your software into modules so that it can be easily maintained and your modules are reusable, whether they are expressed as classes, packages, or other forms? Dave Parnas identified the information hiding principle and showed how to use it to construct workable, reusable modular structures that are stable over time. Are you struggling to create APIs to make your software useful to application programmers? Dave Parnas devised the idea (and coined the term) for abstract interfaces, and showed how to design interfaces that provided services without revealing their implementations. Languages like C++ and Java directly support this idea with abstract classes. Are you wondering how to create your software as a set of layers that define a hierarchical structure that meets your requirements, lets you build your system a few layers at a time, and lets others add to the structure that you have created? Dave Parnas clearly explained what a hierarchical structure is, what some of the important hierarchical structures that we use are, why people often confuse them, and how to create a layered structure that meets your needs. Do you know that your software is going to exist in many different versions, but are having difficulty designing your software not just to accommodate the different versions, but to take advantage of your situation to make your development process more efficient? Dave Parnas defined program families to help with just this situation and showed how to create them in a cost-effective way. However, Dave has been busy in more than just technical areas. His work includes commentary on the social responsibility of software engineers, both by exposition and by example, and on how we should educate our students so that software development becomes an engineering profession. His stance on our inability to create trustworthy software for the Strategic Defense Initiative, as well as his thoughts on how to teach software engineering have influenced how we think, act, and teach, as well as how the public perceives us. David Parnas is both a clear and creative thinker and an extraordinary expositor of seminal ideas. The issues that he addresses are at the heart of software engineering today; his explanations are still relevant and his solutions, trialed on real systems, transfer well to today’s software development organizations and environments.

“Diogenes, Where Are You?”

Frederick P. Brooks, Jr., ACM Turing Award Recipient, University of North Carolina, Chapel Hill, USA

What kind of person do we honor today? Fearlessly honest; honestly fearless. Dave is brilliant; many people are brilliant. Dave is impressively productive; many are productive. Dave is articulate and lucid; many are articulate, some are lucid. As much as we admire that profound, tough, clean mind, it is the attributes of character that we admire more. I submit that it is Dave’s character attributes, as much as his mind, that have produced this incredible body of work. First his honesty. Dave is ready to question his own assumptions, ready to accept such critique by others, and ready to let the chips fall where they may as the logical consequences of assumptions emerge. “Our first duty in research is to the truth.” Second, his principled approach to every task. He has technical principles, which he articulates, which he follows in his research, and which he tests and demonstrates in practice. His principles of professional ethics he has clearly enunciated and consistently followed. His social conscience, his professional conscience, and his personal conscience are all keen. Third, his boldness. He likes to question assumptions, to challenge widely held beliefs and attitudes. He is not shy about criticizing, even satirizing, work he considers shoddy. He freely states opinions, as such, that cannot yet be supported by data, but which he bases on experience and judgment. “I am not a modest man.” Fourth, his commitment. Mill has put it well: Dave has not looked for quick or superficial contributions, but has made a "long-term, focused, painstaking effort." Each of those four words is crucial for his contributions. An honest sketch of an honest man must remark that any one virtue—even honesty—fully embraced, makes some other virtues difficult.
To successfully compete in the drive towards e-business, businesses are faced with challenges that strain their resources across all fronts. In their bid to win new market share, businesses must balance the necessity of new innovative products, released on ever shortening cycles, with the maintenance of their core business—a core that provides the capital leverage needed to fuel this new growth. Businesses must succeed in these changes with a skills base that is, relative to the market requirements, diminishing. Achieving equilibrium between demands that historically have been treated as dichotomous will require nothing less than a change in the very culture of the software engineering community. This change is evident, but how will we, as a the software community, be successful in effecting this change? Successful modification of this culture begins with an understanding, at all levels, of the change in the skills pool and the exponential rise in the need for reliable, scalable systems that can accommodate millions of customers on ever more complex internet based e-business applications. While an increasing number of businesses deploy mission critical applications and begin to build e-market places on the internet, we need to be able to adapt our software engineering philosophy to create software in a more flexible environment that focuses on delivering capability in a more time-critical fashion than we have been challenged to do in the past. The paradigm of designing to perfection must be scaled back to a model that facilitates progressive discovery for the growing population of programmers who are relatively new to the business. Only when we can deliver flexible software to support the deployment of these new e-business applications will we succeed in supporting the drive to e-business.
WEDNESDAY, MAY 16

10:30 am–12:00 pm

S4 TP @Pier 4 & 5
Design and Specification of Distributed Systems
Session Chair: Albert Zündorf, University of Paderborn, Germany
—Composition Patterns: An Approach to Designing Reusable Aspects
S. Clarke, Trinity College, Ireland; R. J. Walker, University of British Columbia, Canada
—MAS- An Interactive Synthesizer to Support Behavioral Modeling in UML
E. Mäkinen, University of Tampere, Finland; and T. Systä, Tampere University of Technology, Finland
—Analysis and Testing of Web Applications
F. Ricca and P. Tonella, ITC-irst (Centro per la Ricerca Scientifica e Tecnologica), Italy

S5 TP @Queen’s Quay
Static Analysis
Session Chair: Joanne M. Atlee, University of Waterloo, Canada
J. M. Cobleigh, L. A. Clarke, and L. J. Osterweil, University of Massachusetts, Amherst, USA
—Static Checking of Interrupt-Driven Software
D. Brylow, N. Damgaard, and J. Palsberg, Purdue University, USA
—Lightweight Analysis of Operational Specifications Using Inference Graphs
L. K. Dillon and R. E. K. Stirewalt, Michigan State University, USA

S6 EDU @Pier 7 & 8
Practical Software Engineering Education
Session Chair: Hossein Saiedian, University of Kansas, USA
—Educating Software Engineering Students to Manage Risk
B. Boehm and D. Port, University of Southern California, USA
—A Formal Approach to Component-Based Software Engineering: Education and Evaluation
M. Sitaraman, Clemson University, USA; T. J. Long, B. W. Weide, Ohio State University, USA; and E. J. Harner and L. Wing, West Virginia University, USA
—Corrective Maintenance Maturity Model (CM³): Maintainer’s Education and Training
M. Kajko-Mattsson, S. Forssander and U. Olsson, Stockholm University and Royal Institute of Technology, Sweden

S7 IIP @Frontenac
Challenges in Software Development Practice
Session Chair: Dewayne E. Perry, University of Texas at Austin, USA
—Global Software Development: The Bell Labs Collaboratory
D. Atkins, Bell Laboratories Lucent Technologies, USA; M. Handel, University of Michigan, USA; J. D. Herbsleb, Bell Laboratories Lucent Technologies; A. Mockus, Bell Laboratories Lucent Technologies; D. E. Perry, University of Texas at Austin, USA; and G. Wills, Bell Laboratories Lucent Technologies, USA
—Does More Necessarily Mean Better? The Software Performance and Reliability Bottleneck
M. Cheng, ACD Systems Canada
—Software Engineering in a Startup
A. Ricciardi, Valaran Corporation, USA

1:10 pm–1:50 pm

S8 FoSP @Pier 4 & 5
Enabling Technologies for the Future of Voice-Based Web Access
Steve Woods, Quack.com, USA
Session Chair: Scott R. Tilley, University of California at Riverside, USA
Voice offers the ultimate in wireless access, providing information on existing data pathways and using existing technology in a new way. Several companies have recently begun offering platforms, tools, and processes for developing and delivering voice-based information applications as part of a SpeechWeb. These companies have been riding on the heels of a revolution in speaker-independent speech recognition that is fueled largely by consistent gains in processor power and memory availability as well as fundamental algorithmic improvements. This talk describes the enabling technologies behind the Quack/AOL Voice Services Platform (VSP), which is comprised of novel approaches to delivering information, to creating applications, and to publishing processes. The VSP not only duplicates the well-known models offered by web servers and feature-rich browsers in the web space, but extends this model to enable new kinds of application publishing paradigms that rely upon semantic models of information navigation rather than visual presentation models.

S9 FoSP @Queen’s Quay
Jini™ Network Technology: Devices, Desires, and Designs
Aleta Ricciardi, Valaran Corp., USA
Session Chair: Gail Kaiser, Columbia University, USA
This talk will examine the way in which Jini™ Network Technology facilitates both software development for distributed applications, and software integration. The connection between device integration and large, heterogeneous software systems will be examined. The technology itself will be presented and compared with other middleware, both distributed object and message-oriented, in the context of designing coherent, complex, scalable systems. I’ll close with a use case showing that the technology enables rapid, clean software practices, and how these in turn lead to a simplified, yet more powerful, notion of “work flow”.

ICSE 2001 • FINAL PROGRAM • http://www.csr.uvic.ca/icse2001/
Simple Object Access Protocol (SOAP) and Web Services

Arthur Ryman, IBM Canada Ltd. Canada
Session Chair: Ric Holt, University of Waterloo, Canada

Simple Object Access Protocol (SOAP) is an exciting new technology for developing Web Services. A Web Service is a set of related application functions that can be programmatically invoked over the Internet. In contrast to traditional Web applications, Web Services are application building blocks that use Extensible Markup Language (XML) for information interchange. In this session you will learn about SOAP, Web Services, and related standards including Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). These technologies will be demonstrated using the IBM XML and Web Services Development Environment which is available for free download from the alphaWorks Web site at http://www.ibm.com/alphaWorks/tech/WSDE.

Auditing Legacy Systems for Security and Survivability

Tom Longstaff, Carnegie Mellon Software Engineering Institute, USA
Session Chair: Prem Devanbu, University of California, Davis, USA

In our rapidly changing IT environment, it is essential that the security posture of all systems and networks are understood. This is extremely difficult when most of the systems that make up our infrastructures are legacy systems. Current state of the art techniques for auditing systems assume that expertise exists in all of the systems and platforms that make up the environment, when in fact many systems that are essential in the infrastructure predate any member of the auditing team. In this case, we show how use cases and intruder scenarios can be used to determine which systems are important from a security perspective and which services and properties of these systems must be verified. These techniques are part of the Survivable Network Analysis method, developed at the Software Engineering Institute to provide a measure of survivability for complex system architectures.
Wednesday, May 16

4:00 pm–5:30 pm

S16 TP @Pier 4 & 5
Effective Uses of Inspections
Session Chair: Lionel Briand, Carleton University, Canada

— Systematic Object-Oriented Inspection: An Empirical Study
  A. Dunsmore, M. Roper, and M. Wood, University of Strathclyde, UK

— Evaluating the Accuracy of Defect Estimation Models Based on Inspection Data From Two Inspection Cycles
  S. Biffl and W. Grossmann, Technical University of Vienna, Austria

— Investigating the Cost-Effectiveness of Reinspections in Software Development
  S. Biffl, Technical University of Vienna, Austria; and B. Freimut and O. Laitenberger, Fraunhofer IESE, Germany

S17 TP @Queen’s Quay
Building Formal Analysis Tools
Session Chair: Thomas Ball, Microsoft Research, USA

— A Component-Based Approach to Building Formal Analysis Tools
  R. E. K. Stirewalt and L. K. Dillon, Michigan State University, USA

— Tool-Supported Program Abstraction for Finite-State Verification
  M. B. Dwyer, J. Hatcliff, R. Joehanes, S. Laubach, Robby, Kansas State University, USA; C. Pasareanu, W. Visser, Research Institute for Advanced Computer Science, USA; and H. Zheng, Kansas State University, USA

— A Workbench for Synthesising Behaviour Models from Scenarios
  S. Uchitel and J. Kramer, Imperial College, UK

S18 EDU @Pier 7 & 8
Software Engineering Body of Knowledge (SWEBOK) Panel
Panel Chair: Peter Freeman, Georgia Institute of Technology, USA

Panelists
- Don J. Bagert, Texas Tech University, USA
- Robert Dupuis, Université du Québec, Montréal, Canada
- Dennis J. Frailey, Raytheon Company, USA
- Hossein Saiedian, University of Kansas, USA
- Mary Shaw, Carnegie Mellon University, USA
- J. Barrie Thompson, University of Sunderland, UK

The goals of the SWEBOK project have been to develop a topical guide to the body of knowledge (BoK) supporting the discipline of software engineering. The project, sponsored by IEEE Computer Society, is over three years old and is nearing completion of its third and final stage. However, there has been some disagreement as to whether there is currently a common core software engineering body of knowledge at its current stage of evolution, and if so, what is size and contents of that BoK. This panel will present the current status of the SWEBOK and discuss its strengths and weakness, as well as address the more general question of the possible existence and nature of a software engineering body of knowledge. The panel discussion will have two parts. The first part will be an informative session. A short history will be presented and issues related to the curriculum, accreditation, and the maturity of the field to warrant a defined BoK will be discussed. In the second part, the panel members will discuss and debate the planned experimentation of the guide, its shortcomings, and how various computing societies may and should cooperate to improve the guide.

S19 IIP @Frontenac
Frontiers of Component Technologies
Session Chair: Gene F. Hoffnagle, IBM Corporation, USA

— Enterprise Methodology /=/= Software Development Methodology
  J. Q. Ning, Accenture, USA

— Mining Selected Components: The Options Analysis for Reengineering (OAR)
  D. Smith, L. O’Brien, and J. Bergey, Carnegie Mellon Software Engineering Institute, USA

— Achieving Usability through Software Architecture
  L. Bass, Carnegie Mellon Software Engineering Institute, USA; and B. E. John, Carnegie Mellon University, USA
THURSDAY, MAY 17

S21 TP @Pier 4
Dynamic Analysis and Testing
Session Chair: David S. Rosenblum, University of California, Irvine, USA
- The Specification and Testing of Quantified Progress Properties in Distributed Systems
  P. Krishnamurthy and P. A. G. Sivilotti, Ohio State University, USA
- An Explorative Journey from Architectural Tests Definition down to Code Tests Execution
  A. Bertolino, IIE-CNR, Italy; and P. Inverardi and H. Muccini, University of L’Aquila, Italy
- Encoding Program Executions
  S. P. Reiss and M. Renieris, Brown University, USA

S22 TP @Pier 5
Construction of Component-Based Systems
Session Chair: Don Batory, University of Texas at Austin, USA
- Dynamic and Selective Combination of Extensions in Component-Based Applications
  E. Truyen, B. Vanhaute, W. Joosen, P. Verbaeten, Katholieke Universiteit Leuven, Belgium; and
  B. Norregaard Jørgensen, Southern University of Denmark, Denmark
  E. Wohlstadter, S. Jackson and P. Devanbu, University of California, Davis, USA
- Designing Components versus Objects: A Transformational Approach
  D. H. Lorenz, Northeastern University, USA; and J. Vlissides, IBM T. J. Watson Research Center, USA

S23 CSR @Frontenac
Infrastructure Support
Session Chair: Jeff Kramer, Imperial College, UK
- Using the Web for Document Versioning: An Implementation Report for Delta-V
  J. J. Hunt and J. Reuter, University of Karlsruhe, Germany
- Evaluating the Reverse Engineering Capabilities of Web Tools for Understanding Site Content and Structure: A Case Study
  S. R. Tilley and S. Huang, University of California, Riverside, USA
- A Case Study of the Evolution of Jun: An Object-Oriented Open-Source 3D Multimedia Library
  A. Aoki, K. Hayashi, K. Kishida, K. Nakakoji, Y. Nishinaka, Software Research Associates, Inc., Japan; B.
  Reeves, TwinBear Research, USA; and A. Takashima and Y. Yamamoto, Nara Institute of Science and Technology, Japan

S24 CHASE @Queen’s Quay
Process-Centered Software Engineering Environments: Academic and Industrial Perspectives
Session Chairs: Wilhelm Schäfer and Gregor Engels, University of Paderborn, Germany
- Process-Centered Software Engineering Environments: Academic and Industrial Perspectives
  R. Balzer, Teknowledge Corporation, USA; and V. Gruhn, University of Dortmund, Germany

S25 FD @Pier 7 & 8
Software Architecture
Session Chair: Rudolf K. Keller, University of Montréal, Canada
- Model Processing Tools in UML
  J. Koskinen, J. Peltonen, P. Selonen, T. Systä, and K. Koskimies, Tampere University of Technology, Finland
- Hyper/JTM: Multi-Dimensional Separation of Concerns for Java™
  H. Ossher and P. Tarr, IBM T. J. Watson Research Center, USA
- Architecture-Oriented Programming Using FRED
  M. Hakala, J. Hautamäki, K. Koskimies, Tampere University of Technology, Finland; and J. Paakkí, A. Viljamaa,
  J. Viljamaa, University of Helsinki, Finland

At a Glance

7:30  Breakfast  @Pier and @Frontenac Foyers
      Speaker’s Breakfast and Room  @Pier 6

8:30-10:00
S21 TP  @Pier 4
S22 TP  @Pier 5
S23 CSR  @Frontenac
S24 CHASE  @Queen’s Quay
S25 FD  @Pier 7 & 8

8:30 am–10:00 am
TP  Technical Papers
CSR  Case Study Reports
CHASE  Challenges and Achievements in SE
FD  Formal Demos

8:30 am–11:30 am
S26 Keynote  M. Shaw  @Frontenac
S27 Award Pres.  @Frontenac
Lunch  @Metro East

1:10–1:50
S28 Awards Pres.  @Frontenac
Break  @Pier and @Frontenac Foyers

2:00–3:30
S29 TP  @Pier 4
S30 TP  @Pier 5
S31 CSR  @Frontenac
S32 CHASE  @Queen’s Quay
S33 FD  @Pier 7 & 8
Nutrition Break  @Pier and @Frontenac Foyers

4:00–5:30
S34 TP  @Pier 4
S35 TP  @Pier 5
S36 CSR  @Frontenac
S37 CHASE  @Queen’s Quay
S38 FD  @Pier 7 & 8
Nutrition Break  @Pier and @Frontenac Foyers

7:00–10:00
S39 Reception  @Frontenac

Internet Café
@Pier 2 & 3, 7:30 am–7:00 pm
Exhibits, Informal Demos, and Posters
@Pier 2 & 3, 8:30 am–7:00 pm

Internet Café
@Pier 2 & 3, 7:30 am–7:00 pm
Exhibits, Informal Demos, and Posters
@Pier 2 & 3, 8:30 am–7:00 pm
The Coming-of-Age of Software Architecture Research

Mary Shaw, Alan J. Perlis Professor of Computer Science, Carnegie Mellon University, USA
Session Chair: Mary Jean Harrold, Georgia Institute of Technology, USA

Over the past decade, software architecture research has emerged as the principled study of the overall structure of software systems, especially the relations among subsystems and components. From its roots in qualitative descriptions of useful system organizations, software architecture has matured to encompass broad explorations of notations, tools, and analysis techniques. Whereas initially the research area interpreted software practice, it now offers concrete guidance for complex software design and development. We can understand the evolution and prospects of software architecture research by examining the research paradigms used to establish its results. These are, for the most part, the paradigms of software engineering. We advance our fundamental understanding by posing research questions of several kinds and applying appropriate research techniques, which differ from one type of problem to another. We identify a variety of research paradigms that yield correspondingly different kinds of results, and require different methods of validation. Unfortunately, these paradigms are not recognized explicitly and are often not carried out correctly; indeed not all are consistently accepted as valid. This retrospective on a decade-plus of software architecture research examines the maturation of the software architecture research area by tracing the types of research questions and techniques used at various stages. We will see how early qualitative results set the stage for later precision, formality, and automation and how results build up over time. This generates advice to the field and projections about future impact.

"Tolerating Inconsistency" Revisited

Robert Balzer, Chief Technical Officer and Director of Distributed Systems Unit, Teknowledge Corporation, USA

We're surrounded by inconsistency in our requirements, in the data that our software processes, and in those software systems themselves. Yet our formal systems can't handle such inconsistency. Most of them lose the ability to form any valid conclusions or analyses in the presence of even a single inconsistency. This forces our programs to operate in terms of an idealized model rather than the real world with the attendant requirement to either maintain a mapping between the two or force human operators to resolve the inconsistencies before the data is processed by the idealized system. My Tolerating Inconsistency paper introduced a simple way to scope formal constraint systems so that they applied only to the consistent data. Data inconsistent with these rules could then be represented and processed by giving them special marks to place them outside the rules’ scope. My talk will review the influence this idea had on the field and my subsequent work.

Applications of New Paradigms in Software Development

Session Chair: Dennis Smith, Carnegie Mellon Software Engineering Institute, USA

— Case Study: Extreme Programming in a University Environment
M. M. Müller and W. F. Tichy, University of Karlsruhe, Germany

— Improving Validation Activities in a Global Software Development Environment
C. Ebert, C. H. Parro, R. Suttels, and H. Kolarczyk, Alcatel, France

— Applying WinWin to Quality Requirements: A Case Study
H. In, B. Boehm, T. Rodgers, and M. Deutsch, Texas A&M University, University of Southern California, USA
### At a Glance

#### 8:30 am–10:00 am

**S40 TP @Pier 4 & 5**

**Analysis of Requirements**

Session Chair: Peri Tarr, IBM T.J. Watson Research Center, USA

- Fast Formal Analysis of Requirements via “Topoi Diagrams”
  T. Menzies, University of British Columbia, Canada; J. Powell, Averstar Inc., USA; and M. E. Houle, University of Sydney, Australia
- Conceptual Modeling through Linguistic Analysis Using LIDA
  S. P. Overmyer, Drexel University, USA, B. Lavoie, CoGenTex, Inc., USA; and O. Rambow, AT&T Research, USA
- A Framework for Multi-Valued Reasoning Over Inconsistent Viewpoints
  S. Easterbrook and M. Chechik, University of Toronto, Canada
- Functional Palaeontology: System Evolution as the User Sees It
  A. I. Antón, North Carolina State University, USA; and C. Potts, Georgia Institute of Technology, USA

#### 3:15–4:45 pm

**S50 FoSP @Frontenac**

#### 4:45–5:00 pm

**S51 SERA @Frontenac**

### S42 Panel @Frontenac

**Perspectives on Software Engineering Panel (PoSE)**

Panel Chair: David Notkin, University of Washington, USA

**Panelists:**

- Marc Donner, Morgan Stanley, USA
- Michael D. Ernst, Massachusetts Institute of Technology, USA
- Michael Corliss, Endeavors Technology, Inc.
- Margaret-Anne Storey, University of Victoria, Canada
- E. James Whitehead, University of California, Santa Cruz, USA

"The Future of Software Engineering"

This panel gives a non-standard view of the future of software engineering. Two of the speakers are recent Ph.D. graduates in computer science, with expertise in software engineering, who have taken academic positions; as people who will educate the next generation of software engineering practitioners and researchers, they provide a key vision of the future. The other two speakers are senior, having moved from the research community into a world in which they face the problems of engineering software on a daily basis. Collectively, along with interactions from the audience, these two often underrepresented perspectives provide a sense of the key directions in which software engineering—practice, research, and education—should and must go.
To be competitive in today’s market, businesses face many challenges in the development and maintenance of information systems. These systems are usually widely distributed. They incorporate highly critical corporate knowledge, which has to be easily accessible and maintainable. Engineering these large systems efficiently requires making decisions about a number of issues. Decisions about whether to build or buy software affect qualities of the software such as customization and reusability, and decisions about mobility of services affect qualities of the software such as maintainability. Moreover, for improved reliability, techniques that can provide seamless but secure and reliable information flow and transaction processing, although expensive, must be integrated into the development process. Finally, systems must be adaptable to new technologies such as wireless computing. Old existing legacy software has to be integrated with new web-based applications such as portals or the whole world of “e”. In this talk, I will consider software engineering from the point of view of a concerned layman, and discuss software development and systems at Lufthansa. I will discuss the underlying architecture of the Lufthansa systems, which consist of a variety of systems and interfaces. I will also discuss the role of the Y2K problem as a germ for renewed life. Finally, I will discuss Lufthansa’s view of the client-server architecture as an intermediate stage towards building advanced information systems, which meet the above mentioned challenges.

Software Engineering Challenges: A CIO’s Perspective
Bernd Voigt, Senior Vice President and Chief Information Officer, Lufthansa, Germany

To be competitive in today’s market, businesses face many challenges in the development and maintenance of information systems. These systems are usually widely distributed. They incorporate highly critical corporate knowledge, which has to be easily accessible and maintainable. Engineering these large systems efficiently requires making decisions about a number of issues. Decisions about whether to build or buy software affect qualities of the software such as customization and reusability, and decisions about mobility of services affect qualities of the software such as maintainability. Moreover, for improved reliability, techniques that can provide seamless but secure and reliable information flow and transaction processing, although expensive, must be integrated into the development process. Finally, systems must be adaptable to new technologies such as wireless computing. Old existing legacy software has to be integrated with new web-based applications such as portals or the whole world of “e”. In this talk, I will consider software engineering from the point of view of a concerned layman, and discuss software development and systems at Lufthansa. I will discuss the underlying architecture of the Lufthansa systems, which consist of a variety of systems and interfaces. I will also discuss the role of the Y2K problem as a germ for renewed life. Finally, I will discuss Lufthansa’s view of the client-server architecture as an intermediate stage towards building advanced information systems, which meet the above mentioned challenges.

Antipatterns in Software Architecture
Tom Mokrav, World Wide Institute of Software Architects (WWISA), USA

This presentation will provide an introduction to common Antipatterns in Software Architecture, and their resolution through reference models. Antipatterns are an intriguing variation of design patterns. Antipatterns clarify ineffective software practices, as well as raise awareness of alternative solutions. In the context of software architecture, Antipatterns can clarify the necessity of problem-solving skills revealed in software engineering reference models, such as the Software Design Level Model (SDLM) and the Reference Model for Open Distributed Processing (RM-ODP). In particular, this presentation will discuss how traditional software engineering assumptions become serious Antipatterns, when applied to distributed software architectures.

Dependability of Embedded Systems
John Knight, University of Virginia, USA

Embedded systems present especially difficult challenges in software engineering because they usually operate in real time on machines with modest resources and can cause extensive damage if they fail. In practice the interplay between systems and software plays a significant role in dependability and must be considered in embedded software development. In this presentation the basic issues of dependability as they apply to embedded systems will be discussed, and several systems-engineering and software-engineering techniques for improving dependability will be reviewed. Topics to be covered include hazard analysis, fault-tree analysis, formal specification, design for safety, and verification.

Inter-Language Object Sharing with the Common Language Runtime: Infrastructure for MS .NET
Jennifer Hamilton, Microsoft Corp., USA

The Common Language Runtime (CLR) is language and platform-neutral, and provides the underlying infrastructure for the Microsoft .NET Framework. It consists of several components, including a garbage collector, class loader, metadata engine, just-in-time compiler, and debugging and security services. This talk will introduce the CLR, describe its design, and demonstrate how it enables inter-language object sharing.

Collaborative Software Engineering
Alan Brown, Catapulse Inc., USA

Software development is a collaborative activity involving the sharing of ideas, artifacts, and knowledge among teams of engineers. Many elements are emerging to enhance the importance of collaboration in software development geographically dispersed teams, more partnering within and across organizations, and leveraging of the Internet as a coordination infrastructure. In particular, the future of software development will be through leveraging the Internet as the base on which collaborative development services will be offered. To be successful requires that the next generation of software development environments be based on the “Three C’s”–community, collaboration, and coordination. This presentation examines the requirements for future software development environments, illustrates the kinds of services that must be offered, and examines the challenges that must be overcome in collaborative environments for building the next generation of enterprise scale software solutions. As a result, the presentation explores the current state of collaborative software engineering, analyses the currently available collaborative technologies, and establishes the current state-of-the-practice in collaborative software engineering.
**S48 Keynote @Frontenac**

**Reuse That Pays**

Linda M. Northrop, Director, Product Line Systems Program, Carnegie Mellon Software Engineering Institute, USA

Session Chair: Hausi A. Müller, University of Victoria, Canada

A company builds a software system capable of running a diesel engine in a week, and in one case over a weekend, as opposed to the full year that it used to take. Another company builds one of its typical systems with 13 software engineers instead of the more than 100 it once required, and at the same time decreases the system’s defect rate tenfold. Still another increases its software-intensive product offerings from four per year to 50 per year. Imagine being able to use one person to integrate and test 1.5 million source lines of Ada for a real-time command-and-control system onboard a ship, with safety-critical requirements? Or increasing software productivity fourfold over three years, as another company has done? These organizations all achieved their results through strategic software reuse. We software people have been promising the benefits of reuse for decades. Are we finally achieving a reuse strategy that lives up to its hype?

**S49 TP @Queen’s Quay**

**Architecture for Emerging Applications**

Session Chair: Elisabetta Di Nitto, Politecnico di Milano, Italy

— An Architecture for Heterogeneous Groupware Applications
  I. Marsic, Rutgers University, USA

— A Web-Oriented Architectural Aspect for the Emerging Computational Tapestry
  K. J. Sullivan and A. Saxena, University of Virginia, USA

— XAS: A System for Accessing Componentized, Virtual XML Documents
  M.-L. Lo, S.-K. Chen, S. Padmanabhan and J.-Y. Chung, IBM T. J. Watson Research Center, USA

**S50 Panel @Frontenac**

**Software Engineering Research Agendas Panel (SERA)**

Panel Chair: Dieter Rombach, Fraunhofer IESE, Germany

Panelists:
- Ross Jeffery, University of New South Wales, Australia
- Ali Mili, West Virginia University, USA
- Leon J. Osterweil, University of Massachusetts, Amherst, USA
- David L. Parnas, McMaster University, Canada
- Tetsuo Tamai, University of Tokyo, Japan

“*What can’t we do, but need to learn how to do?”*

The software challenges of the new millennium include more mature users expecting functioning software, more critical technical and business applications requiring dependable software, globalization requiring distributed development teams, and paradigm clashes between new and old economy firms. Software engineering has to be evaluated anew in terms of “what can’t we do today, and what do we have to learn how to do in order to meet those challenges.” This panel discusses and proposes urgent research topics as well as research programs to address those topics. Although, such discussions are going on in all different countries (e.g., PITAC in the US, similar activities in Germany), this international panel will add new perspectives through inter-cultural cross-fertilization. The panel will consist of three parts: position statements and brief discussions among panelists, questions/answers from the audience, and summary.
Magic and Illusion

Alan Nackan mystifies with a deck of cards, causes solid to melt through solid, increases money tenfold, makes objects disappear, and even predicts the future.

Food and soft drinks will be served along with your choice of a complimentary wine or beer. A cash bar will be available. Enjoy the evening!

Tuesday, May 15, 2001
5:30 pm–9:00 pm
@Harbour Ballroom, @Harbour Foyer, and @Pier 2 & 3

Sponsored by ACD Systems, Canada

Harvey Seigel’s “Speak-Easy Jazz Band”

Relax and enjoy the company of your colleagues during an evening filled with the sounds of jazz and ragtime.

Food and soft drinks will be served along with your choice of a complimentary wine or beer. A cash bar will be available.

Wednesday, May 16, 2001
7:00 pm–10:00 pm
@Harbour Ballroom, @Harbour Foyer, and @Pier 2 & 3

Sponsored by IBM Canada Ltd.

Canadiana Coast to Coast

Enjoy an evening filled with music, laughter, singing, dance and fun! The show highlights the most popular music and dance of Canada. Both traditional favorites and current hits of today’s most celebrated Canadian artists encompasses Canada past to present, featuring the music of such famous artists as: The Rankin Family, Gordon Lightfoot, Anne Murray, Stomping Tom, Great Big Sea, Céline Dion, Bryan Adams, Shania Twain, and many more.

Food stations and soft drinks will be available along with your choice of a complimentary wine or beer. A cash bar will be available.

Thursday, May 17, 2001
7:00 pm–10:00 pm
@Frontenac

Sponsored by National Research Council of Canada
Exhibits, Posters and Informal Demos

ICSE 2001 exhibits feature publishers and leading companies offering software engineering capabilities and tools, education and training, books and journals, and consulting services. Formal research demos, posters and informal research demonstrations are displayed in the exhibits area and provide the opportunity to exhibit late-breaking results and to discuss these results with conference participants.

<table>
<thead>
<tr>
<th>Schedule for Exhibits</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tuesday, May 15</strong></td>
</tr>
<tr>
<td><strong>Wednesday, May 16</strong></td>
</tr>
<tr>
<td><strong>Thursday, May 17</strong></td>
</tr>
<tr>
<td><strong>Friday, May 18</strong></td>
</tr>
</tbody>
</table>

**Set Up**
- **Tuesday, May 15**: 2:00 pm–5:45 pm | @Pier 2 & 3
- **Friday, May 18**: 2:00 pm–5:00 pm | @Pier 2 & 3

**Formal Demos Presentations (see Thursday schedule for details)**
- **Thursday, May 17**: 10:30 am–5:30 pm | @Pier 2 & 3

Internet Cafés

You are welcome to visit the two ICSE 2001 Internet Cafés, which provide central zones for meetings, discussions, and collaboration of researchers and developers from around the world. Read your email, and visit all the interesting URLs you will notice during the conference and the exhibits at once. And don’t forget the ICSE 2001 web page with the latest announcements, up-to-date information, and the WOW daily newsletter. Come and meet your colleagues!

<table>
<thead>
<tr>
<th>Schedule for Internet Cafés</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monday, May 14</strong></td>
</tr>
<tr>
<td><strong>Tuesday, May 15</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Wednesday, May 16</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Thursday, May 17</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Friday, May 18</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Set Up**
- **Monday, May 14**: 8:30 am–12:00 pm | @Bay
- **Tuesday, May 15**: 2:00 pm–5:45 pm | @Pier 2 & 3

**Tear Down**
- **Friday, May 18**: 2:00 pm–5:00 pm | @Bay | @Pier 2 & 3

The ICSE 2001 Web Site
A Great Software Engineering Resource

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

Exhibits

Chairs: Homy Dayani-Fard, IBM Toronto Laboratory, Canada; homy@ca.ibm.com; and Kostas Kontogiannis, University of Waterloo, Canada; kostas@swen.uwaterloo.ca

Commercial Demos
- ACD Systems
- ASERC
- IBM (VJava)
- IBM (MQSeries)
- IBM (OLT)
- KLocWork
- Neuma
- Semantic Designs
- Rational

Publishers
- Artech House Publishers
- Cambridge University Press
- Elsevier Science
- John Wiley
- Kluwer
- Pearson PTR Canada

A  IBM (OLT/DD)
B  IBM (MQSeries)
C  IBM (VAJava)
D  Hyper/J: Multi-Dimensional Separation of Concerns for Java, IBM T. J. Watson, USA
E  Maintenance Support Tools for Java Programs: CCFinder and JAAT, Osaka University, Japan
F  Using OCL-Queries for Debugging C++, Clemson University, USA
G  Neuma Technology Corporation, Canada
H  KLOCwork Solutions, Canada
I  jMOCHA: A Model Checking Tool that Exploits Design Structure, University of Pennsylvania and University of California, Berkeley, USA
J  Architecture-Oriented Programming Using FRED, Tampere University of Technology and University of Helsinki, Finland
K  Semantic Designs, Inc.
L  Babel: Representing Business Rules in XML for Application Integration, University of Alberta, Canada
M  Model Processing Tools in UML, Tampere University of Technology, Finland
N  Rational Software
O  ACD Systems, Canada
P  CodeWeb: Data Mining Library Reuse Patterns - University of New South Wales, Australia
Q  Holmes: An Intelligent System to Support Software Product Line Development, University of Alberta, Canada
R  Artech House Publishers
S  Pearson PTR Canada
T  Springer Verlag
U  Kluwer
V  John Wiley & Sons
W  Elsevier Science
X  Cambridge University Press
Z  Alberta Software Engineering Research Consortium

Resources from the ICSE 2001 sponsors (see Page 5) can be found on the resource tables throughout the exhibits.
## Collocated Events and Meetings

### At a Glance

<table>
<thead>
<tr>
<th>Events</th>
<th>Fri May, 11</th>
<th>Sat May, 12</th>
<th>Sun May, 13</th>
<th>Mon May, 14</th>
<th>Tue May, 15</th>
<th>Wed May, 16</th>
<th>Thu May, 17</th>
<th>Fri May, 18</th>
<th>Sat May, 19</th>
<th>Sun May, 20</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conferences</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICSE 2001 Int'l Conference on Software Engineering 2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EHCI 2001 Engineering for Human-Computer Interaction 2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IWPC 2001 Intl. Workshop on Program Comprehension 2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSR 2001 Symposium on Software Reusability 2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPIN 2001 Spin Workshop on Model Checking of Software 2001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ICSE Meetings</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICSE 2001 WOW Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICSE 2002 Org. Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICSE 2002 PC Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICSE 2003 Org. Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICSE St. Committee Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICSE 2001 Post Mortem Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Birds of a Feather Session</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CeBASE Collaborators Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Other Meetings</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACM SIGSOFT General Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEEE TCSE General Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEEE TSE Ed. Board Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICSM 2001 PC Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESEC/FSE St. Committee Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESEC/FSE PC Meeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Venue Locations

- @Pier 5
- @Dockside I–IV
- @Harbour C and Pier 4
- @Harbour C
- @Pier 9
ICSE Meetings
- ICSE 2001 Window-on-the-World Meeting
  Saturday, May 12, 11:00 am–12:00 pm, @Vice Regal Suite
- ICSE 2002 Organizational Meeting
  Tuesday, May 15, 3:30 pm–5:30 pm
- ICSE 2002 Program Committee Meeting
  Thursday, May 17, 5:30 pm–7:00 pm, @Pier 7 & 8
- ICSE 2003 Organizational Meeting
  Thursday, May 17, 5:30 pm–7:00 pm, @Pier 5
- ICSE Steering Committee Meeting
  Tuesday, May 15, 8:30 pm–11:00 pm, @Pier 4
- ICSE 2001 Post Mortem Meeting
  Saturday, May 19, 8:00 am–10:00 am, @Wellington

Birds of a Feather Session
- CeBASE Collaborators Meeting
  Chairs: Victor R. Basili and Barry Boehm
  CeBASE, the NSF sponsored Center for Empirically-Based Software Engineering aimed at strengthening and propagating the results of empirical research is having a meeting of current and perspective CeBASE collaborators. The goal is to provide an update on activities for collaborators, discuss potential directions and future opportunities, and provide a forum for interaction.
  Thursday, May 17, 5:30 pm–7:00 pm, @Pier 4

Other Conference and Organizational Meetings
- ACM SIGSOFT General Meeting
  Wednesday, May 16, 8:00 pm–9:00 pm, @Pier 4 & 5
- IEEE TCSE General Meeting
  Wednesday, May 16, 9:00 pm–10:00 pm, @Pier 7 & 8
- IEEE TSE Editorial Board Meeting
  Thursday, May 17, 12:00 pm–1:30 pm, @Dockside IV
- ICSM 2001 Program Committee Meeting
  Saturday, May 19–Sunday, May 20, @Dockside I
- ESEC/FSE Steering Committee Meeting
  Wednesday, May 16, 12:00 pm–1:00 pm, @Dockside IV
- ESEC/FSE Program Committee Meeting
  Saturday, May 19–Sunday, May 20, @Dockside II

First name ___________________ Last name ___________________ Title ___________________
Name as preferred on badge ________________________________
Company/Institution ________________________________
Address ____________________________________________
City __________________ State/Province ______ Zip/Postal Code ______ Country __________________
Phone __________________ Fax __________________
E-mail ____________________________________________

I am a member of:  [ ] IEEE  [ ] IEEE CS  [ ] ACM  [ ] ACM SIGSoft  [ ] ACM SIGPlan
[ ] I am a full-time student and am including proof of current student status with my registration.
[ ] I have special needs and access: ________________________________
[ ] I do NOT want my address distributed.

TUTORIALS  (Please circle tutorial numbers)

| Sunday Full Day | T1 | T2 |
| Monday Full Day | T5 | T6 | T7 |
| T8 | T9 | T10 | T11 |
| T12 | T13 | T14 | T15 |
| T16 | T17 | T18 | T19 | T20 |
| Tuesday Morning | T21 |
| Tuesday Afternoon | T22 |

WORKSHOPS & COLLOCATED EVENTS  (Please circle number or name)

| Friday - Saturday - Sunday | EHCI |
| Saturday - Sunday (5/12 - 5/13) | IWPC |
| Sunday | W1 | W2 | W3 | W4 |
| Sunday-Monday | W5 | W6 |
| Monday | W7 | W8 | W9 | W10 | W11 |
| Monday-Tuesday | W12 | W13 | W14 |
| Tuesday | W15 | W16 | W17 | W18 |
| Tuesday Morning | New SE Faculty Symposium (no registration fee) |
| Tuesday | David Lorge Parsnas Symposium |
| Saturday - Sunday (5/19 - 5/20) | SPIN |
| Saturday | SSR |

(To be filled out by attendees)

On or Before April 12, 2001

| ICSE 2001 Full Conference (Wed, Thu, Fri) | $430 | $590 | $175 | $530 | $680 | $250 |
| One Day Conference:  [ ] Wed  [ ] Thu  [ ] Fri | $250 | $250 | $250 | $250 | $250 | $250 |
| Tutorial: Full Day or 2 Half Days | $350 | $450 | $200 | $450 | $550 | $225 |
| Tutorial: Half Day | $200 | $275 | $125 | $250 | $300 | $150 |
| Workshop: 2-Day | $250 | $350 | $125 | $385 | $475 | $235 |
| Workshop: 1-Day | $150 | $200 | $75 | $275 | $345 | $150 |
| David Lorge Parsnas Symposium | $250 | $320 | $175 | $320 | $400 | $250 |
| IWPC | $310 | $360 | $120 | $375 | $470 | $145 |
| EHCI | $400 | $425 | $125 | $500 | $525 | $150 |
| SPIN | $175 | $225 | $125 | $205 | $255 | $155 |
| SSR | $300 | $325 | $125 | $330 | $365 | $155 |
| SSR Tutorial* | $130 | $150 | $100 | $150 | $180 | $130 |
| Exhibit Hall Only (One Day) | $20 | $20 | $20 | $20 | $20 | $20 |
| Additional Reception Ticket | $50 | $50 | $50 | $50 | $50 | $50 |
| Additional Proceedings | $30 | $30 | $30 | $30 | $30 | $30 |

TOTAL ENCLOSED: $________________

ICSE 2001 Conference fee includes entry to all technical sessions, exhibits, receptions, nutrition breaks, and one copy of conference proceedings.
One Day Conference fee includes entry to one day of technical sessions, exhibits, reception if applicable, nutrition breaks for one day, and one copy of conference proceedings.
Tutorial fee includes entry to chosen tutorial(s), luncheon(s), nutrition break(s), and applicable tutorial notes.
Workshop fee includes entry to chosen workshop(s), luncheon(s), nutrition break(s), and preprints of workshop position papers.
Collocated Conference(s) (IWPC, EHCI, SPIN, SSR) fee(s) include entry to the chosen conference(s), luncheon(s) (if provided), nutrition break(s) and conference proceedings.
Cancellations made in writing and faxed or postmarked by April 12, 2001 will be accepted subject to a $50 cancellation fee. Refunds will be made within four weeks of the end of the conference. Cancellations will not be accepted after April 12, 2001. “No shows” are not refundable and are liable for the full registration fee. Instead of canceling, your registration may be transferred by giving a colleague a written authorization.

*You will choose your tutorial onsite. Please see http://www.abitmore.be/ssr2001/

ICSE 2001 Conference fee includes the keynote by Professor Frederick Brooks on Tuesday evening.

Online Registration:
http://www.csr.uvic.ca/icse2001/
Fax Registration: +1-407-628-3186
Mail form to:
ICSE 2001
P. O. Box 941156
Mailand, FL 32794-1156 USA

For express mail only:
ICSE 2001
c/o Registration Systems Lab
2060 Goldwater Court
Mailand, FL 32751 USA

REGISTRATION FORM
General Chair
Hausi A. Müller, University of Victoria, Canada; hausi@cs.uvic.ca

Program Chairs
Mary Jean Harrold, Georgia Institute of Technology, USA; harrold@cgatech.edu
Wilhelm Schäfer, University of Paderborn, Germany; willi@uni-paderborn.de

Technical Program Committee
Joanne M. Atlee, University of Waterloo, Canada
Thomas Hass, Microsoft Research, USA
Don Bavota, University of Texas at Austin, USA
Israel Ben-Shaul, Technion—Israel Institute of Technology, Israel
Daniel M. Berry, University of Waterloo, Canada; Technion—Israel Institute of Technology, Israel
Antonia Bertolino, Istituto di Elaborazione della Informazione CNR, Italy
Lionel Briand, Carleton University, Canada
Lori A. Clarke, University of Massachusetts, Amherst, USA
Laura K. Dillon, Michigan State University, USA
Elisabetta Di Nitto, Politecnico di Milano, Italy
Matthew B. Deyer, Kansas State University, USA
Phyllis C. Frankl, Polytechnic University, USA
Kokiichi Futatsugi, JAIST, Japan
Peter H. H. Green, Hewlett-Packard Laboratories, Palo Alto, USA
Martin Cline, University of Zürich, Switzerland
Koichi Hata, Osaka University, Japan
Jens H. Jahnke, University of Victoria, Canada
Meihs H. Jazayeri, Technical University of Vienna, Austria
Richard A. Kemmerer, University of California, Santa Barbara, USA
Nicole Ley, University of Versailles et Ile Villeneuve, France
Jeff Magee, Imperial College, UK
Nenad Medvidovic, University of Southern California, USA
Leon O. Osterweil, University of Massachusetts, Amherst, USA
Dwayne E. Perry, University of Texas at Austin, USA
Alexander Ran, Nokia Research Center, USA
Crusoe T.Roman, Washington University, St.Louis, USA
David S. Rosenblum, University of California, Irvine, USA
Greg Rothermel, Oregon State University, USA
Gregor Snelting, University of Passau, Germany
Kevin J. Sullivan, University of Virginia, USA
Peter Tarr, IBM T.J. Watson Research Center, USA
Walter F. Tichy, University of Karlsruhe, Germany
Jim Welsh, University of Queensland, Australia
Linda W. Wilson, Carnegie Institute of Technology, USA
Michal Young, University of Oregon, USA
Albert Zündorf, University of Paderborn, Germany

Case Study Reports
Jeff Kramer (co-chair), Imperial College, UK; jk@doc.ic.ac.uk
Dennis Smith (co-chair), Carnegie Mellon Software Engineering Institute, USA; dbs@sei.cmu.edu
Cerardo Canfora, University of Sannio, Italy
Wolfgang Emmerich, University College London, UK
Alfonso Fuggetta, Politecnico di Milano, Italy
Kai Koskimies, Tampere University of Technology, Finland
Timothy C. Lethbridge, University of Ottawa, Canada
Adam Porter, University of Maryland, College Park, USA
June Verner, Drexel University, USA

Education Papers
Anthony Finkelstein (co-chair), University College London, UK; a.finkelstein@cs.ucl.ac.uk
Hossein Saiedian (co-chair), University of Kansas, USA; saiedian@ece.ku.edu
Donald J. Bagert—Texas Tech University, USA
Robert Dupuis—University of Quebec, Montreal, Canada
Mike Lutz—Rochester Institute of Technology, USA

Challenges and Achievements in SE
Gregor Engels (co-chair), University of Paderborn, Germany; engels@uni-paderborn.de
Wilhelm Schäfer (co-chair), University of Paderborn, Germany; willi@uni-paderborn.de

Invited Industry Presentations
François Coailler (co-chair), Bell Canada, Canada; francois.coailler@bell.ca
Linda M. Northrop (co-chair), Carnegie Mellon Software Engineering Institute, USA; linn@sei.cmu.edu
Dwayne E. Perry (co-chair), University of Texas at Austin, USA; perry@ece.utexas.edu
Mihito Aoyama, Nipata Institute of Technology, Japan
Volker Grun, University of Dortmund, Germany
Gene F. Hoffnagle, IBM Corporation, USA
Jim Q. Ning, Accenture, USA
Jacob Slonim, Dalhousie University, Canada

Frontiers of Software Practice
Scott R. Tilley (co-chair), University of California, Riverside, USA; stolley@cs.ucr.edu
Hausi A. Müller (co-chair), University of Victoria, Canada; hausi@cs.uvic.ca

Perspectives on SE Panel
David Notkin (chair), University of Washington, USA; notkin@cs.washington.edu

SE Research Agendas Panel
Dieter Rombach (chair), Fraunhofer IESE, Germany; rombach@iese.fhg.de

New SE Faculty Symposium
Leon O. Osterweil (co-chair), University of Massachusetts, Amherst, USA; ljo@cs.umass.edu
Dieter Rombach (co-chair), IBM Toronto Laboratory, Canada; gabbbys@ca.ibm.com

Perspectives on SE Panel
David Notkin (chair), University of Washington, USA; notkin@cs.washington.edu

SE Research Agendas Panel
Dieter Rombach (chair), Fraunhofer IESE, Germany; rombach@iese.fhg.de

New SE Faculty Symposium
Leon O. Osterweil (co-chair), University of Massachusetts, Amherst, USA; ljo@cs.umass.edu
Dieter Rombach (co-chair), IBM Toronto Laboratory, Canada; gabbbys@ca.ibm.com
Kenny Wong (co-chair), University of Alberta, Canada; kenny@cs.ualberta.ca

David Lorge Parnas Symposium
Dan Hoffman (co-chair), University of Victoria, Canada; dhoffman@cs.uvic.ca

David M. Weiss (co-chair), Avaya Communication, USA; weiss@avaya.com

Tutorials
Gail Kaiser (co-chair), Columbia University, USA; kaiser@cs.columbia.edu
Paul Sorenson (co-chair), University of Alberta, Canada; sorenson@cs.ualberta.ca
Alan Brown, Carnegie Mellon University, USA
Paul C. Clements, Software Engineering Institute, USA
Gregor Engels, University of Paderborn, Germany
Oscar Nieterstrasz, University of Bern, Switzerland
Scott R. Tilley, University of California, Riverside, USA
Ursula Wozl, The College of New Jersey, USA

Workshops
Prem Devanbu (co-chair), University of California, Davis, USA; devanbu@cs.ucdavis.edu
John Mylopoulos (co-chair), University of Toronto, Canada; jm@cs.toronto.edu
Anneliese Amschler Andrews, Colorado State University, USA
Harald Gall, Technical University of Vienna, Austria
David Carlan, Carnegie Mellon University, USA
Roland Mittermeier, University of Klagenfurt, Austria
Frank Tipp, IBM T.J. Watson Research Center, USA

Doctoral Symposium
Mary Lou Soffa (chair), University of Pittsburgh, USA; sof2@cs.pitt.edu
Marsha Chechik, University of Toronto, Canada
Chris Martin, Flinders University, Australia
Aditya Mathur, Purdue University, USA
Jeff Offutt, George Mason University, USA
Mary Shaw, Carnegie Mellon University, USA
Margaret-Anne Storey, University of Victoria, Canada

Research Demos and Posters
Rudolf K. Keller (chair), University of Montréal, Canada; keller@iro.umontreal.ca
Gerald Cannond, Arizona State University, USA
George T. Heineman, Worcester Polytechnic Institute, USA
Spencer Rugaber, Georgia Institute of Technology, USA
Eleni Stroulia, University of Alberta, Canada
Steve Woods, Quack.com, USA

Exhibits and Tools Fair
Homy Dayani-Fard (co-chair), IBM Toronto Laboratory, Canada; homy@ciba.ibm.com
Kostas Kontogiannis (co-chair), University of Waterloo, Canada; kostas@pavan.uwaterloo.ca
Nicholas Graham, Queen’s University, Canada
Bruno Lagué, IBM Toronto Laboratory, Canada
Marin Litoiu, IBM Toronto Laboratory, Canada
Spiros Mancoridis, Drexel University, USA
Melody Moore, Georgia State University, USA
Ciancarlo Succo, University of Alberta, Canada
Chris Verhoef, University of Amsterdam, The Netherlands

Publicity
Bashar Nuseibeh (co-chair), Bashar Nuseibeh, The Open University, UK; B.A.Nuseibeh@open.ac.uk
Gabriel Silberman (co-chair), IBM Toronto Laboratory, Canada; gabbbys@ca.ibm.com
Victor R. Basili, University of Maryland, USA
Judith Bishop, University of Pretoria, South Africa
Elliot Chokofsky, META Group, USA
Aniello Cimatti, University of Naples, Italy
Peter Feiler, Carnegie Mellon Software Engineering Institute, Santa Clara, USA
Miguel Feller, University of Buenos Aires, Argentina
W. Morven Gentleman, Dalhousie University, Canada
Michel Lemoine, ONERA CERT, France
Nikolai Mansurov, Institute for System Programming, Russia
Walcélio Melo, Catholic University of Brasilia, Brazil
Ettoire Merlo, Ecole Polytechnique de Montréal, Canada
All Mili, West Virginia University, USA
Václav Rajlich, Wayne State University, USA
Karl Reid, The Telebrese University, Australia
Dieter Rombach, Fraunhofer IESE, Germany
Richard N. Taylor, University of California, Irvine, USA
Koji Torii, Nara Institute of Science and Technology, Japan
Will Tracez, Lockheed Martin Federal Systems, USA
Yelena Vesha, University of Maryland, Baltimore County, USA

Treasurer and Local Arrangements
Stephen Perelgut (chair), IBM Toronto Laboratory, Canada; perelgut@ca.ibm.com

Student Volunteers
Mike Godfrey, University of Waterloo, Canada; migod@plg.uwaterloo.ca
Jens H. Jahnke, University of Victoria, Canada; jens@cs.uvic.ca

Electronic Submission
Howard Johnson, National Research Council of Canada, Canada; Howard.johnson@nrc.ca
Anatol W. Kark, National Research Council of Canada, Canada; anatol.kark@nrc.ca
Stephan MacKay, National Research Council of Canada, Canada; stephen-mackay@nrc.ca

WOW Newsletter, Advance Program, Final Program, and Documents
Anke Weber, University of Victoria, Canada; anke@cse.uvic.ca

Web Site, Design, and Advertising
Kenny Wong, University of Alberta, Canada; kwong@cs.ualberta.ca
2002 International Conference
on Software Engineering

May 19-25, 2002
Buenos Aires, Argentina

General Chair:
Will Tracz <will.tracz@lmco.com>

Program Chairs:
Michal Young <michal@cs.uoregon.edu>
Jeff Magee <jnm@doc.ic.ac.uk>


ICSE 2003
International Conference on Software Engineering
http://www.icse-conferences.org/2003/

May 3-10, 2003
Portland, Oregon

General Chair:
Lori Clarke
University of Massachusetts
clarke@cs.umass.edu

Program Chairs:
Laurie Dillon
Michigan State University
dillon@cse.msu.edu

Walter Tichy
University Karlsruhe,
Germany
tichy@ira.uka.de
ICSE 2001 • Westin Harbour Castle Hotel • Toronto, Ontario, Canada • May 12-19, 2001

ICSE 2001
Conference Information

Department of Computer Science
University of Victoria
EOW Room 337
P.O. Box 3055 STN CSC
Victoria, BC, Canada
V8W 3P6
Fax: +1-250-721-7292
E-mail: icse2001@csr.uvic.ca