# TABLE OF CONTENTS

## LETTERS

1. From the Chair – William Griswold
2. From the Editor – Will Tracz
3. The Second SEN Logo Design Contest
4. Crossword Puzzle – Jetty Heyman
5. New SEN Web Pages – Greg Cooper
6. The Second Annual SE Crossword Puzzle Contest
7. You are invited to ESEC/FSE 2007! – Antonia Bertolino and Ivica Crnkovic
8. A Call for Professionalism, Now and Forever – Larry Bernstein
10. Upcoming SE Conference Map – Tao Xie and Sung Kim
11. Nominations Solicited from SIGSoft Members for the Software System Award
12. Comment on Risks to the Public May 2007, F-22 – Dr. Gregory Chapelle
13. SIGSOFT Impact Paper Award – William Griswold

## COLUMNS

8. ACM TOSEM: FAQs and Figures – David Notkin
9. Superhack – Michael Wing
10. Surfing the Net for Software Engineering Notes – Mark Doernhoefer
20. Risks to the Public – Peter G. Neumann and contributors

## PAPERS

30. The list of Papers is found on page 2

## WORKSHOP REPORTS

33. The list of Workshop Reports is found on page 2

## REVIEWS

53. TSP: Coaching Development Teams – reviewed by Mordechai Ben-Menachem

## WORKSHOP & CONFERENCES

54. Conference Acronyms / Calendar of Events – Yu Feng
56. ICSE
57. ESEC/FSE
Letter from the Chair

William G. Griswold
wgg@cs.ucsd.edu

I have just returned from a rewarding week at ICSE 2007. It was great seeing you so many of you. Let me provide a quick report from my (biased) SIGSOFT perspective: It was a great conference. Frankly, I was afraid of a letdown after last year’s ICSE’06 in Shanghai, but I needn’t have been worried. Thanks to the efforts of General Chair John Knight, co-PC Chairs Gregg Rothermel and Wolfgang Emmerich, and John’s energetic assistant Peggy Reed, it was a wonderful and rewarding time. We had three great keynotes, a great excursion to Niccolet Island (we rode Segways!), a leading-edge technical program, and great conversation in the hallways. In truth, it’s the people that make ICSE. Thanks to everyone who did their part, and I hope to see you all in Leipzig the same time next year (http://icse08.upb.de/).

ICSE is of course one of SIGSOFT’s two signature events (the other being the annual SIGSOFT conference that hosts FSE symposium). As such, we did some special things at ICSE.

First, SIGSOFT held a “Town Hall” meeting on Wednesday evening, where we quaffed stunningly good local brews and ruminated on the future of software engineering and SIGSOFT’s role therein. A big question was what we can do for practicing software engineers in a constantly changing field. In part we reaffirmed our tri-part mission of supporting the community’s efforts in education, research, and policy, noting that ACM itself does much for professional software engineers, such as providing instant access to 1000 books through portal.acm.org, as well as providing access to personal benefits such as affordable medical insurance. We also acknowledged the need to do more in guiding the development of SE curricula – perhaps at the Masters level – and helping to create excitement in undergraduates for software engineering to reverse possible declining interest in SE. In fact, coming ICSE’s will feature an undergraduate project competition – the timing couldn’t be better. You can learn more about it at http://score.elet.polimi.it/.

Second, I had the great pleasure of announcing and presenting several awards. This year’s Research Award went to Elaine Weyuker of AT&T Labs for her numerous foundational and practical contributions to the field of testing. Elaine is already an ACM Fellow, among numerous other recognitions. A distinct pleasure in making her award was noting all she has done from her “bully pulpit” for women in computing, for example currently serving as the Chair of the ACM Committee on Women in Computing. This year’s Service Award went to David Notkin of the University of Washington, whose long history of service includes SIGSOFT Chair, ICSE and FSE PC Chair, CRA Board member, and currently Editor in Chief of ACM TOSEM. David was my Ph.D. advisor, so this was particularly rewarding for me. Both awards were decided by nominations from the community and discussion within the SIGSOFT leadership.

Third, I recognized SIGBED’s first two choices for the SIGBED/SIGSOFT Frank Anger Award, Cesar Sanchez of Stanford and Bernhard Eggers of Seoul National University. This is an interdisciplinary student travel award, and Bernhard was in attendance at the conference; Cesar will attend next year.

I’m looking forward to SIGSOFT’s upcoming selections for this award, who will attend core SIGBED conferences like EMSOFT.

Finally, I was happy to invite the ICSE audience to recognize the ACM Fellow selection of SIGSOFT members Alex Wolf (SIGSOFT’s most recent Past-Chair – way to go, Alex!) and Matthias Felleisen.

Upon my return from ICSE, I was happy to learn that SIGSOFT’s proposal for an “Impact Paper Award” was approved. This award will annually recognize the most influential paper appearing at any SIGSOFT conference, 10 years ago or earlier. This award will start in 2008, and the authors of the selected paper will be invited to make a retrospective keynote at SIGSOFT FSE in Atlanta in November 2008. Additionally we will be retroactively recognizing several papers, since SIGSOFT has been sponsoring conferences for some 20 years now. For details, see the write-up on the SIGSOFT Impact Paper Award in this issue of SEN.

Sincerely,

William G. Griswold

Letter from the Editor

Will Tracz
<Will.Tracz@ACM.org>

Another ICSE for the history books! From Bill’s description and from the photos I had planned on including in this issue, it sounded like a very good ICSE. And yes, I had hoped to see you there, but, my priorities were rearranged and I was unable to attend, or do much of anything else, including getting this issue of SEN out on time. Speaking of being there, this issue contains 4 workshop reports (3 ICSE), a CFP for next year’s ICSE and a letter and AP describing the upcoming ESEC/FSE in Croatia. I am really impressed with the program and location of ESEC/FSE – check it out!

That’s not all you should check out. In this issue, let me recommend Mike Wing’s usual outstanding and entertaining column – and there is one paper you will have to read – online “The Grand Theory of Everything: What Man-Made Systems Are, and Why They Fail” by Robert Schaefer. While you are online, I recommend checking out the Upcoming SE Conference Map that Tao Xie and Sung Kim have put together, as well as other things on that web site. Of course, Mark Doerrhofer (how does he do it?) put together a truly interesting read in his Surfing the Net column.

In closing, let me float an idea, I was thinking about proposing a “War Story” Workshop at a future SIGSOFT event. It would have NO student registration fee and would feature “real world” recaps of “Lessons that are not (formally) taught in the classroom.” Let me know what you think! Until next time!
Table 1: SEN Volume 32 Number 4 Papers

<table>
<thead>
<tr>
<th>#</th>
<th>Title/Author</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ordering of Events in Two-Process Concurrent System</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Jayasri Banerjee, Anup Kumar Bandyopadhyay, and Ajit Kumar Mandal</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Application of Dijkstra’s Weakest Precondition Calculus to Dining Philosophers Problem</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Jayasri Banerjee, Anup Kumar Bandyopadhyay, and Ajit Kumar Mandal</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Research Directions in Verification and Validation Process Improvement</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Ayaz Farooq and Reiner R. Dumke</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SEC+: An Enhanced Search Engine for Component-Based Software Development</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Sofien Khemakhem, Khalil Drira, and Mohamed Jmaiel</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Code Quality Tools: Learning from our Experience</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>R Krishnan, S Murali Krishna and Nishil Bharill</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Adaptive Web Service Composition</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Fernando Antônio Aires Lins, José Carlos dos Santos Júnior, and Nelson Souto Rosa</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Designing Software Components to Tolerances</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Richard Riehle</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>The Grand Theory of Everything: What Man-Made Systems Are, and Why They Fail</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Robert Schaefer</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>An Improved Component Model for Component-based Software Engineering</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>R. Senthil, D. S. Kushwaha, and A. K. Misra</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>A Step Towards Software Preventive Maintenance</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Yogesh Singh and Bindu Goel</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>On Checking the Consistency of Object-Z Classes</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Fathi Taibi, Jacob K. Daniel, and Fouad Mohammed Abbou</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: SEN Volume 32 Number 4 Workshop Summaries

Letters to the Editor

The Second SEN Logo Design Contest

Submissions are now being accepted for the second SEN Logo Design Contest. Single page PDF file entries should be sent to the SEN EOC (Will.Tracz@LMCO.COM).

Editor’s Note: A concerned reader has expressed an opinion about the Logo-loss and other ideas to make SEN more SENsational. Readers comments and suggestions are always welcomed.

Hi,
I got my copy of SIGACT News a few days ago and, just like our formerly artsy SEN, it also has lost its artsy logo. So I guess the logo-deprivation was nothing personal against SEN. However, I hasten to point out to you that although SIGACT News doesn’t have a high-class intellectual feature such as a crossword puzzle it DOES now have photos included for most of its columnists and famous named people (e.g winner of the Knuth prize). It’s kind of fun to see how everyone got older, except for me.

That reminds me – I was going to put some ICSE photos in this issue, but I ran out of time/space. Oh well, now you have something to look forward to in the next issue. BTW, they look better in color on the portal.
Across

1. cvs and subversion are examples of this

5. description of project - of which design is the next step

6. Admiral of COBOL fame

7. should be done for each module before claiming complete

9. various measurements used to track progress

12. approach for recreating software from a published specification

14. Peter G Neumann's column

15. spiral/iterative model produces one at each iteration

18. opposite of programming in the small

19. newest focus in SE - tw person team programming

20. Dr Barry, famous SE author

21. mathematical way of validating software

Down

2. Dr Barry's estimation model

3. Classic SE design model

4. User's definitions of what project should be

8. pictoral view of a program

10. Interactive Development Environment


13. Computer Aided Software Engineering

14. using a component over and over

16. mid 1980s Grady Booch approach, "blank" oriented

17. emacs or vi (also known as "blank" wars)

20. short term for software construction
New SEN Web Pages
Greg Cooper
g.r.co@att.net

When you get a chance, take a look at http://www.sigsoft.org/SEN/newindex.html and let us know what you think of the look.

The Second Annual
SE Crossword Puzzle Contest

Yes, we have submissions! In fact 4 submissions! Of course there is always room for more! I would like to thank Jerry Heyman and Conrad Weisert (who submitted 3!) for their entry. Jerry’s is found on the previous page, with the answer on page 8.

Submissions will continue to be accepted for the second annual Software Engineering Crossword Puzzle Contest. Single page PDF file entries should be sent to the SEN EOC (Will.Trazz@LMCO.COM). Entries will be published in SEN as they are submitted. Prizes will be awarded (though I can’t say what they will be now).

You are invited to ESEC/FSE 2007!
Antonia Bertolino
Program Chair
ISTI-CNR, Italy
<antonia.bertolino@isti.cnr.it>

Ivica Crnkovic
General Chair
Mälardalen University, Sweden
<ivica.crnkovic@mdh.se>

It is our great pleasure to invite you to be part of ESEC/FSE’07, the 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering. The biennial series of joint ESEC and FSE provides a lively and outstanding forum where researchers and practitioners can report and discuss leading edge research results and trends, as well as their impact on industrial applications in the broadest area of software engineering (SE). This year’s event offers what we believe will be a very strong and exciting program on both technical and social sides.

To start with, we have a rich and attractive list of invited presentations. In his opening keynote talk, Roberto Siagri, President and CEO of Eurotech, will envision how the computers of the next future will be hidden everywhere into the surrounding environment to provide us with an “augmented” reality. The 2007 ACM SIGSOFT Outstanding Research Award recipient is Elaine Weyuker from AT&T Labs-Research: based on her five-years experience with investigating faults patterns in large systems, she will focus on the complete path for doing good SE research, all along from problem inception to eventually packaging the results and making them accessible to practitioners.

To the benefit of all ESEC/FSE attendees, we decided this year not to have as by tradition a separate tutorials track, but rather to package a series of State-of-the-Art (SOTA) presentations provided by leading researchers within the main program in plenary session. The SOTA Chairs (Mehdj Jazayeri, from University of Lugano and Technical University of Vienna, and David Rosenblum, from University College London) propose you an interesting blend of three talks, one for each Conference day, on up-to-date SE themes: Steffen Staab from Koblenz-Landau University will first overview ontology technologies, such as those employed for the emerging Semantic Web, and then contemplate possible avenues to a “happy marriage” between ontology and software technology; Marta Kwiatkowska from Oxford University will analyze challenges for probabilistic model checking, a promising technique for the quantitative verification of modern systems increasingly facing the need to manage critical resources and stringent requirements; finally, Walt Scacchi from University of California at Irvine will review free and open source software development (FOSSD), including current work practices, project and community dynamics, and socio-technical relationships, to finish with a hint on future opportunities. These three SOTA presentations will certainly make an important part of your ESEC/FSE bring-home baggage.

Other invited talks will include an outline of opportunities in the seventh Framework Programme delivered by Paolo Brasciani, Project Officer of the European Commission, and a tribute to Dean Rosenzweig, Professor at Zagreb University, who passed away in January 2007 (presented by Andre Scedrov from University of Pennsylvania).

The traditional track of full Research Papers includes thirteen Technical Sessions, with 42 highest quality papers selected by the ESEC/FSE PC among a record number of 251 submissions (that was a hard task!). In view of this very positive response to our call for contributions, we have also selected some nice-idea works for a very interesting Poster exhibition. Another novelty launched this year is the Widened Software Engineering (WISE) track, co-chaired by Antonia Bertolino (yes, me again) and Henry Muccini from Università dell’Aquila. WISE explicitly aims at widening international participation. Of the 23 submissions received from 15 different countries, the WISE PC selected two papers, from Bulgaria and Pakistan, reporting about education in those countries, and a third one, from Jordan, surveying Web Engineering practice.

The main program is preceded by nine focused Workshops on advanced research themes (please see on the ESEC/FSE web site for their list), selected by the workshop chairs Matthew Dwyer from University of Nebraska, Lincoln, and Amy Murphy from FBK-IRST; and by the Doctoral Symposium, chaired by Carlo Ghezzi of Politecnico di Milano.

ESEC/FSE 2007 will be held in Cavtat/Dubrovnik, in the beautiful Croatia region. It is really a charming location not to be missed, and if you can we warmly advice to bring your family along! The conference hotel, Croatia Hotel, placed on a beautiful peninsula, offers many facilities for meetings, sport and other activities.

ESEC/FSE 2007 social program includes a reception at the historical place Sponza Palace in the old city of Dubrovnik, which we will reach by a boat trip, and the conference dinner on the Lido terrace of the Croatian Hotel, with an astonishing view on the sea, Cavtat and Dubrovnik on the horizon. Finally, for the many ESEC/FSE passionate runners, we are proud to announce a primer: the SE.run( ) 5/10k Race along the charming peninsula...
of Cavtat, with prizes for winners, and nice T-shirts and caps for all participants: the race will take place on Thursday 6th early morning.

Haven't we convinced you yet? Then please visit http://www.idt.mdh.se/esec-fse-2007 and have a look to the impressive views. Early registration deadline expires on July 31, and if you need a Visa be sure to apply for it early in advance.

With all ESEC/FSE organizers and of the members of ESEC/FSE and WISE Committees (whom are too many to list in this one page summary, but to whom we are strongly indebted for their invaluable contribution\(^1\)) we look forward to receiving you in Cavtat/Dubrovnik for an unforgettable event!

---

**A Call for Professionalism, Now and Forever**

Larry Bernstein  
Fellow of ACM  
Stevens Institute of Technology  
<bernstein@ieee.org>  
http://guinness.cs.stevens-tech.edu/~lbernste/

---

The last several issues of SEN are informative, easy to read, and attractive. Well done!

The 2007 International Conference for Software Engineering (ICSE) put me in mind of your publication last year of the scathing Defense Software Summit Report of October 2006. Attendees were treated to a floor debate that included these claims: “...the ACM/IEEE ethics are ‘inert’,” “software engineers do not agree on what minimum knowledge you need to do your job,” “…you are interchangeable,” and even worse, “anyone can be a software engineer.” A year ago the DOD said the root causes of our unprofessional behavior “…include lack of requirements discipline, limited staff experience, external adverse influences, poor planning, resource and budget constraints, weak contract management, funding turbulence, inadequate program documentation, and incomplete risk management.” Have we learned nothing? Can we not accept thoughtful analysis?

The DOD recommended the following in 2006:

- Establish strategic initiatives for acquisition process improvement, program measurement, training and education, architecture, system of systems integration, analysis, and planning.
- Perform systems and software engineering over the life cycle.
- Integrate software acquisition in make or buy frameworks.
- Understand that software is a primary performance, schedule, and cost driver; recognize that software engineering is inseparable from disciplined systems engineering; and define a capability engineering framework.
- Remove cultural barriers between information technology and weapon software developers; and need to track and manage the health of system and software engineers.
- Increase leadership awareness; improve engineering practice and discipline; and develop and retain a skilled work force.
- Exploit reuse to reduce cost and schedule; promote architecture; and use open architecture and product line approaches.
- Establish centralized policy; integrate systems and software engineering; and provide guidebooks for Product/Project/Program Managers (PMs).
- Address acquisition management, engineering, development techniques, business implications, and human resources.
- Establish near-term tasks including the following: Conduct “proof of concept” Six Sigma software-intensive projects; incorporate process improvements where feasible; provide request-for-proposal preparation for software-intensive systems guidance; collect metrics; improve quality; develop and conduct a software leadership course; and establish and use software product lines.
- Improve software estimating.

This is 2007 and we are still arguing about our differences. As we argue, enrollments in software engineering programs decline, systems engineering programs teach technology originally developed for software engineers, and clients look elsewhere for the skills they need. The state of the software engineering profession alarms me.

Many actions are needed to build a stronger software engineering profession. Among the most important is to strengthen what we teach. Today, there is no model curriculum for graduate software engineering education. Every university is on its own to decide what to include in its program. The lack of a model graduate curriculum undermines the public’s reliance on the quality of senior software engineering professionals. Let’s address that lack and create a model graduate software engineering curriculum.

To ensure the value of that graduate-level curriculum and to encourage its broad adoption, it should be drafted in a collaborative manner with leading academic authorities, the primary professional societies with a strong software engineering faction, government sponsors, and industrial organizations whose employees are the direct consumers of software engineering education. The resulting curriculum should be suitable for an education leading to a Masters Degree in Software Engineering. If done right, the curriculum will help universities turn out professionals who understand how to produce trustworthy software, on-time and within budget. The curriculum will be responsive to the needs of those who employ software engineers because those employers will help define it. And finally, it will include many aspects of systems engineering because almost all of today’s interesting systems are software-intensive. (In fact, over time, the distinction between systems engineering and software engineering will continue to blur.)

A project to create a model graduate-level software engineering curriculum has been started at Stevens Institute of Technology by Dr. Art Pyster. To participate, reach him at art.pyster@stevens.edu.

---

\(^1\)Editor’s Note: A partial list can be found on page 63.
Grace Hopper Celebration of Women in Computing 2007

The Grace Hopper Celebration of Women in Computing 2007 (www.gracehopper.org) is the seventh in a series of conferences designed to bring the research and career interests of women in computing to the forefront. Presenters are leaders in their respective fields, representing industrial, academic and government communities. Leading researchers present their current work, while special sessions focus on the role of women in today’s technology fields, including computer science, information technology, research and engineering.

Past Grace Hopper Celebrations have resulted in collaborative proposals, networking, mentoring, and increased visibility for the contributions of women in computing. This year’s theme is “I Invent the Future”.

The Grace Hopper Celebration is a program of the Anita Borg Institute for Women and Technology – www.anitaborg.org - and is co-hosted with the Association for Computing Machinery (ACM).

Upcoming SE Conference Map

Tao Xie
Sung Kim

We have created a web for upcoming SE conference map: http://ase.csc.ncsu.edu/semap/. We will actively maintain and add new conferences to the map along the way. We will keep you updated on the SELife portal.

Editors Note: This web site also contains links to:
• Software Engineering Conferences,
• Software Engineering Academic Genealogy,
• Software Engineering Awards
• Others cool things…. Check it out!!!

Nominations Solicited from SIGSoft Members for the Software System Award

This is an ACM award to an institution or individual(s) recognized for developing a software system that has had a lasting influence, reflected in contributions to concepts, in commercial acceptance, or both. The Software System Award carries a prize of $10,000.

Nominations should be submitted by October 31 to the committee chair Carlo Ghezzi <carlo.ghezzi@polimi.it> (see http://www.acm.org/awards/award_nominations.html)

A chronological listing of past recipients of the award can be found at: http://awards.acm.org/software%5Fsyste...

Comment on Risks to the Public

May 2007, F-22
Dr. Gregory Chapelle
chapelle@ieee.org

Dear Dr. Neumann,

This letter is to comment on a Risks to the Public article published in May 2007 ACM SIGSOFT Software Engineering Notes. In particular my comments will address the article “USAF F-22 jets grounded by software glitch (R 24 58)”.

I believe your comment at the end of the article “However, the F-22 Raptor was presumably unwrapped without the benefit of raptor simulation, testing, and other preflight analyses. Perhaps the quality control is going downhill” was out of line. I personally worked on the Raptor Integrated CNI (Communications Navigation and Identification) system and can attest that extensive 4+ years of testing and analysis went into that system.

I think the fundamental “take away” from this is not “what a bunch of stupid idiots”, but rather what was the basic development/testing process problem that allowed this issue to slip through. I think I can shed some light on this.

Basically with a complex system like this, the government (the Air Force in this particular instance) has detailed specific performance requirements that the system must meet. A great deal of design and testing go into verifying that the system meets these functional requirements. Even failure modes are addressed when resource sharing and detailed studies/testing are performed to keep classified operational computer data from being accidentally released into the unclassified processors. Extensive operational scenarios were developed, and detailed Rate Monotonic Analysis were performed for each of these.

The hardest part in trying to meet these large number of requirements is to step back and say “what have I forgotten to test for”. It’s easy to test and identify the written requirements in front of you, but much more difficult to identify less obvious failure modes.

The real kicker for most people when learning of this reported error is how obvious it is in hindsight. Why didn’t we test for this obvious operational mode of crossing the Interna-
tional Date Line? This is probably the second "take away" from this error. For Navigation, an accurate time reference is the key that unlocks everything. We tested for timing errors in GPS, requisition of time if it was lost, and numerous other "time" type of errors. We were confident that we had addressed any time reference errors, but we never specifically addressed the International Date Line. In the rush to verify functional requirements, we did not look carefully at our testing coverage, and because we danced around similar failure modes, we were confident that we had "covered all our bases". Again, I would say if we had stepped back and took a careful look at the "completeness" of our testing, we might have identified this hole.

So where does that put us today? I think today the pressures on software development to produce and test faster prevents a "stand down" moment. The fast pace does not allow reflective contemplation for an overall view of a project's objectives and to confirm adequate design and testing. As systems become more complex there comes a need for a group/person separate from the design group to cast an impassionate eye over things and independently verify that there are no "unresolved operational issues".

While I no longer work for the Integrated CNI group, I continue to work on government programs and even with 20+ years of experience strive to improve and do better. I always find your "Risks to the Public" engaging and unnerving at the same time. I share pertinent articles my design teams and will be discussing the F-22 one with them too!

Thank you for helping to make software systems better.

SIGSOFT Impact Paper Award
William Griswold

Description
Presented annually to the author(s) of a paper presented at a SIGSOFT sponsored or co-sponsored conference held at least 10 years prior to the award year. In including all of SIGSOFT's conferences in the competition, this award recognizes the breadth and vitality of the software engineering community. The papers are judged by their influence since their publication. The award includes a $1000 honorarium to be split amongst the authors as they choose, a award certificate of recognition for each author, an invitation for the authors to present a retrospective keynote talk at the current year's annual SIGSOFT Foundations of Software conference, as well as inclusion of a full-length retrospective paper in the SIGSOFT conference proceedings. Travel support in the amount of $2000 will be provided, split amongst the attending authors as they choose. A public citation for the award paper will be placed on the SIGSOFT web site.

Selection Committee
The award given in year N is for a highly influential paper presented at a conference held in calendar year N-10 or prior. A selection committee and selection committee chair will be selected by the current SIGSOFT Executive Committee. The committee chair shall adjudicate conflicts of interest, appointing substitutes to the committee as necessary. For purposes of continuity, committee members may remain on the committee for up to three years. The award committee shall be no less than three people in size.

Funding
The SIGSOFT Impact Paper Award is a SIGSOFT award, and will be funded by SIGSOFT.

Proposed Procedure
Nominations will be solicited annually during the December prior to the award year, via major mailing lists and web forums. Additionally, the General Chair and Program Chair of each eligible N-10 and current year conference will be contacted (as available) to form a satellite committee to nominate 1 paper from their N-10 conference. Additionally, citation counts in major citation indices will be used to identify the top 10 cited SIGSOFT papers from the N-10 publication year, as well as the top 25 cited SIGSOFT papers prior to the N-10 year. A multi-round ranking procedure will be used to identify the top few papers, and then a final decision will be made by consensus of the committee. The selection committee has the prerogative to make no award, but not to make multiple awards (with one exception outlined below).

The first award year will be made in 2008, and thus the initial 10 year eligibility year will be 1998. In the first 5 years of the award, an additional selection committee will be appointed in the same method as above, to make up to 23 additional retrospective awards, no more than 5 per year, for papers published prior to 1998, the first N-10 year (SIGSOFT was founded in 1975, 23 years prior to 1998). This committee will communicate with the other committee to avoid duplicate awards. Retrospective awards will comprise an award certificate for each author and a public citation of the award.

Upcoming SE Conference Map
http://ase.csc.ncsu.edu/semapi/ - In case you missed it – you have to check out this web site – lots of cool stuff!!!!!
Paper Abstracts

Paper 1:

Ordering of Events in Two-Process Concurrent System
Jayasri Banerjee¹, Anup Kumar Bandyopadhyay² and Ajit Kumar Mandal³
Electronics and Telecommunication Engineering Dept. Jadavpur University, Kolkata – 700032, India
{anupbandyopadhyay@hotmail, comajit.k.mandal@vsnl.com}

Abstract
Dijkstra’s weakest precondition calculus is extended to capture temporal ordering in concurrent systems. This is done by defining temporal ordering predicates that is used to describe necessary conditions. A new logical connective, viz., “implies in the past” is also defined to describe the cause and effect relationships. Ordering mechanism used in Peterson’s two process mutual exclusion algorithm is explained by proving a theorem.

Keywords: program correctness, program specification, parallel processing, weakest precondition

Paper 2:

Application of Dijkstra’s Weakest Precondition Calculus to Dining Philosophers Problem
Jayasri Banerjee, Anup Kumar Bandyopadhyay and Ajit Kumar Mandal
Electronics and Telecommunication Engineering Dept. Jadavpur University, Kolkata – 700032, India
{anupbandyopadhyay@hotmail, ajit.k.mandal@ieee.org}

Abstract
Dijkstra’s weakest precondition calculus is extended to model the well known Dining Philosophers problem. Process and state definitions are done in such a manner that only the deadlock property of the system is highlighted. Care has been taken to choose the proper details such that it is not too elaborate to obscure the requirements also not be too abstract to mask the actual analytical needs. State transition rules specify the system behavior. Intuitive reasoning as well as formal technique has been applied to get the deadlock condition. Two well known solutions are specified and proved. The proof technique being analytical, its complexity does not depend on the size of the problem. The second solution requires an event ordering and therefore a temporal ordering predicate has been used to prove its correctness.

Keywords: software process, SP, software process improvement, SPI, verification & validation process, V&V, software test process, STP, test process improvement, Testing Maturity Model, TMM

Paper 3:

Research Directions in Verification and Validation Process Improvement
Ayaz Farooq and Reiner R. Dumke
Institute for Distributed Systems
University of Magdeburg
P.O. Box 4120, 39106 Magdeburg, Germany
{farooq@ivs.cs.uni-magdeburg.de, Khamil@laas.fr

Abstract
Software process establishment, evaluation and improvement are key research areas in the software engineering field today. Extensive research has been carried out and many different kinds of approaches exist to improve the software process and even more efforts are underway. Verification & validation process, which is part of the broader software process activities, plays a vital role in quality and profitability of the developed product but is believed to consume major portion of the development expenses and resources. Probably, research towards improving the verification & validation process has not been as actively directed as compared to software process improvement research. This paper identifies several potential future research directions towards improving verification and validation process.

Keywords: software process establishment, evaluation, improvement, V&V process, software test process, STP, test process improvement, Testing Maturity Model, TMM
**Paper 5**

**Code Quality Tools: Learning from our Experience**

R Krishnan, S Murali Krishna and Nishil Bharill
Motorola Software Group, Bangalore, India

Abstract

In this paper we share some of our experiences relating to tools used in coding phase. We primarily focus our discussion on two topics, namely UT (Unit Testing) and Memory related errors.

Unit Testing (UT) [1] is a critical early-phase verification activity that ensures the quality of the product soon after the code is ready. Newer paradigms of software development like Extreme Programming [7] have a strong focus on UT. However, UT is very effort-intensive as it involves test harness development and repetitive execution. In Motorola software Group, we evaluated a set of commercial tools and selected a tool for use in our projects. In this paper, we will share the evaluation criteria, which was identified for the tool selection and the various technical challenges encountered in adopting a UT tool solution in a heterogeneous environment along with the associated learning. One of the major challenges encountered, was arriving to a consensus on definition of term ‘Unit’. The other issues included handling of complex data structures, event driven code, integration with development environment etc.

One of the major categories of errors introduced and found in coding phase relates to memory. As these errors may have security implications, leaving them undetected might be a big risk. For trapping memory related errors, a framework to categorize these errors was created along with the associated code samples. In this paper, two types of memory related errors will be discussed, namely memory leak and buffer overflow. This memory related errors can be exploited to compromise the security of the system. This paper also discusses the causes, types and impact of buffer overflows, as well as the precautions to be taken to prevent buffer overflows.

**Keywords:** Code quality, Memory leak, Buffer overflow and memory corruption

**Paper 7**

**Designing Software Components to Tolerances**

Richard Riehle
Computer Science Department
Naval Postgraduate School, Monterey, California

Abstract

Many traditional engineering designs, other than software, depend on the physical properties of components. Those properties enable the engineer to specify precise tolerances between those components. Software components are abstractions with no inherent physical properties. The absence of physical properties makes it more difficult, but not impossible, to design to tolerances. This paper describes some design metrics for designing software components to tolerances. It uses some already established design metrics, and expands on the role of other software practices already available. This paper also restricts the discussion to software components, rather than to the algorithms contained within those components.

**Keywords:** Tolerances, Metrics, Snugness of Fit, Assertions, Design, Components, Constraints, Pre-conditions, post-conditions, invariants

**Paper 8**

**The Grand Theory of Everything: What Man-Made Systems Are, and Why They Fail**

Robert Schaefer
Daniel Webster College

Abstract

The Grand Theory of Everything (tGToE) is a powerful, elegant and unique Model which may be used towards the Understanding and Development of Man-Made Systems. This Model may be used to Identify, Explore, and Predict Faults of Systems-Making, and Faults in Systems and Faults in Systems-Use. Although the tGToE Model provides an Understanding of Systems and Choices, it may not necessarily identify Best Choices or Practical Solutions as that requires Judgment. As all Models are Abstractions, the Possibility of oversimplification that applies to Models applies equally to tGToE. Refinement of this Model has led the Author to several Curious Observations on Paradoxes in Systems and Observations on the Use of Power in Organizations. Depending on the Ethics of the Reader, the application of tGToE may be used to solve Great Problems or to cause Great Mischief and Harm.
Paper 9

An Improved Component Model for Component-based Software Engineering
R. Senthil, D. S. Kushwaha and A. K. Misra
Department of Computer Science & Engineering
Motilal Nehru National Institute of Technology,
Allahabad, India.
<sentil77@gmail.com>, <dharkush@yahoo.com>, <arun_kmisra@hotmail.com>

Abstract
This paper focuses on Component Based Software Engineering (CBSE) and generic connectors for the software components. An attempt has been made to describe n-tier architecture; in particular, data access architecture in a component based application. It attempts to express how data access objects interact with business-tier and data-tier in achieving reusable, robust and scalable component based architecture by implementing Data Adapter interface. It has been able to establish that code efficiency increases when data access object implements the data adapter interface by performing unit test on the code. It has also been able to show that there exists a loose coupling in the component based architecture.

Keywords: software components, composition, pattern, encapsulation, component-based software engineering, interfaces

Paper 10

A Step Towards Software Preventive Maintenance
Yogesh Singh and Bindu Goel
University School of Information Technology,
Guru Gobind Singh Indraprastha University,
Kashmere gate, Delhi (INDIA)
<ys66@rediffmail.com>, <bindu_delus@yahoo.com>

Abstract
In a world interwoven economically, increasing dependence on critical software applications either in transaction processing (banking, Government services etc) or manufacturing automation (automobiles, pharmaceuticals, chemicals) has accentuated the economic impact aging software can have. Software with increased usage operating in unforeseen conditions at throughput much higher then initial expectations can degrade fast leading to higher altered form and dramatic drop in performance indicators. In the current paper we have tried to analyze the issues governing software maintenance and how preventive maintenance which is still considered a very evolving field in the context of software engineering can help the software product age usefully. We have also attempted to address the above in the essence of how it is done for hardware preventive maintenance which is a better understood and commercially accepted concept. Finally we suggest model for the preventive maintenance integrated within software life cycle.

Keywords: Software maintenance, maintainability, hardware preventive maintenance, documentation

Paper 11

On Checking the Consistency of Object-Z Classes
Fathi Taibi1, Jacob K. Daniel2, Fouad Mohammed Abbou3
1Faculty of IT, University of Tun Abdul Razak
Selangor, Malaysia
2Faculty of IT, Multimedia University
Selangor, Malaysia
3Faculty of Engineering, Multimedia University
Selangor, Malaysia
<taibi@unitar.edu.my>, <jacob@mmu.edu.my>, <abbou@mmu.edu.my>

Abstract
Requirements elicitation involves gathering requirements from different stakeholders. Different stakeholders often hold different views of how a system should behave, resulting in inconsistencies between their descriptions. Rigorous consistency checking methods can be effectively applied if the different views are formally specified. This is possible because of the unambiguous and precise nature of formal specification languages. However, ensuring that each formal view is self-consistent is critical before checking that the different views are inter-consistent. In this paper, an algorithm is proposed to check the self-consistency of the classes of an Object-Z specification. The proposed approach combines specification testing, model abstraction, and model checking to perform the verification.

Keywords: Formal specification, Object-Z, Consistency, Specification testing, Model abstraction, Model checking

Visit SIGSOFT
http://www1.acm.org/sigs/sigsoft/
Visit SEN
http://www.sigsoft.org/SEN/

-Don’t Forget-
1. You can design a SEN Logo
2. Read Mike Wing’s article
3. Send in a happy word puzzle
4. Register for ICSE