

Guest Editorial

Special Issue for the Second Asia-Pacific Conference on Quality Software

1. Background

The quality of software has an important bearing on the financial and safety aspects in our daily lives. Unfortunately, software systems often fail to deliver to promises. It is well known that there are still unresolved errors in many of the software systems that we are using every day. The prime objective of the Asia-Pacific Conference on Quality Software (APAQS) is to bring together researchers and practitioners to address this issue seriously.



The Second Asia-Pacific Conference on Quality Software (APAQS 2001) was held in Hong Kong on December 10–11. The conference attracted a total of 100 submissions from 25 countries/regions: 46 from Asia, 31 from Europe, 15 from North America, and 8 from Australia. The quality of submissions was impressive. After a process of rigorous peer reviews, 34 full papers, nine short papers and seven experience reports were accepted for presentation at the conference. Prof. C.V. Ramamoorthy of University of California at Berkeley and Dr Ray Paul of the Office of the Assistant Secretary of Defense, USA, presented keynote addresses to the delegates. The conference proceedings were published by IEEE Computer Society Press [6].

Authors of eight selected papers at the conference were invited to submit expanded versions for this special issue. Substantial improvements on the conference versions were required. The expanded papers underwent a further review process to ensure that they conform to the standard of the journal. Finally, four papers were accepted for publication in this special issue.

2. The papers

Testing is undoubtedly one of the most popular methods used in practice for ensuring the quality of software. In the first paper “Optimal Software Testing and Adaptive Software Testing in the Context of Software Cybernetics”, the author K.Y. Cai notes the similarity of software testing with the control problem widely studied in control engineering. He develops a theory that employs the controlled Markov chain (CMC) approach. In this approach, the testing strategy serves as a controller and the software under test serves as a controlled object. The corresponding optimal test profile is found to be Markovian under certain conditions, but not the case otherwise. When these conditions are met, the use of the Markovian model for statistical software testing [4, 5] is theoretically justified. When these conditions are not met, Markovian statistical software testing is not optimal and can be further improved. Cai also introduces the adaptive software testing strategy, which can be adjusted on-line as the tester’s understanding of the software is improved. Simulation results show that the adaptive software testing strategy is superior to random testing, which is one of the common benchmarks of testing strategies.

Compared to testing, formal methods have undeniably been less frequently applied in practice. Despite decades of development, there has still been much controversy on the benefits of formal methods [1–3]. The second paper, “Detection of Dynamic Execution Errors in IBM System Automation’s Rule-Based Expert System” by C. Sinz, T. Lumpp, J. Schneider, and W. Küchlin, reports on a successful industrial application of formal verification techniques to real production software. The authors have formally verified aspects of the rule-based expert system in IBM’s System Automation software for the zSeries mainframes. The paper demonstrates the usefulness of their formalization and verification techniques by detecting residual non-termination bugs in a tested program version close to shipment, and then verifying the absence

absence of this class of bugs in the corrected production code. Furthermore, the authors have done an excellent job in reflecting their experiences and relating them to the well-known “myths” of formal methods [1, 3]. The paper is highly recommended for anyone interested in applying the use of formal methods in practice.

The commercialization of the Internet has made e-commerce a popular topic, as reflected by the keen interest of the conference delegates. The remaining two papers in this special issue both deal with e-commerce, albeit using two entirely different approaches.

The third paper, “A CSP and Z Combined Modeling of Document Exchange Processes in E-commerce Protocols” by X. Wang, S.C. Cheung, and J. Wei, studies the quality of e-commerce protocols, which are a vital component of the trading infrastructure over the Internet. The authors argue that it is difficult to build a quality e-commerce protocol because it is deployed in an open environment and is data context dependent. To address this issue, the authors have developed a formal approach of modeling e-commerce protocols in a combined language of CSP and Z. They have applied their approach to the Internet Open Trading Protocol (IOTP), a trading protocol recommended by the Internet Engineering Task Force (IETF). The paper reports their formal model of IOTP and discusses the challenges they have encountered in the formalization process.

The final paper, “Software Processes for the Development of Electronic Commerce Systems” by V. Gruhn and L. Schöpe, is an experience report on the development process of a real e-commerce system. The system is an e-commerce portal called Internet Portal System for Insurances (IPSI), intended to provide support for insurance agents in their daily work. The system demonstrates some of the idiosyncrasies of e-commerce systems whose development requires adaptation of conventional software processes. By generalizing the main features of the software process adapted for the development of IPSI, the authors have proposed a process model that may be useful for other e-commerce system development projects with a high degree of integration.

The concern for software quality has called for the use of many different approaches. The collection of papers in this special issue has well illustrated the more commonly used or advocated approaches: testing, formalization and verification, and process improvements. Taken together, the papers have also presented a balanced snapshot of the contemporary advances in the theories and practices of the field.

3. Looking ahead

In view of the success and global interests of the APAQS in two consecutive years since its inception, the Steering Committee has decided to expand the conference into a truly international one. The successor of APAQS 2001 will be known as the Third International Conference on Quality Software (QSIC 2003), to be held on September 25–26, 2003 in Beijing, China. Prof. Huimin Lin of the Institute of Software, Chinese Academy of Sciences, China, and Prof. Hans-Dieter Ehrich of the Technical University of Braunschweig, Germany, will serve as Program Co-Chairs. The acronym “QSIC” reminds us of the high quality and strength of Cu-SiC, a ceramic reinforced metal. For more information of QSIC 2003, please visit <http://www.csis.hku.hk/~apaqs/qsic2003.html>.

References

- [1] J.P. Bowen, M.G. Hinchey, Seven more myths of formal methods, *IEEE Software* 12 (4) (1995) 34–41.
- [2] N. Fenton, S.L. Pfleeger, Can formal methods always deliver? *IEEE Computer* 30 (2) (1997) 34.
- [3] A. Hall, Seven myths of formal methods, *IEEE Software* 7 (5) (1990) 11–19.
- [4] J.H. Poore, Introduction to the special issue on: model-based statistical testing of software intensive systems, *Information and Software Technology* 42 (12) (2000) 797–799.
- [5] J.A. Whittaker, M.G. Thomason, A Markov chain model for statistical software testing, *IEEE Transactions on Software Engineering* 20 (10) (1994) 812–824.
- [6] Y.T. Yu, T.Y. Chen (Eds.), *Proceedings of the Second Asia-Pacific Conference on Quality Software (APAQS '01)*, IEEE Computer Society, Los Alamitos, CA (2001).

T.Y. Chen ^{a*}

T.H. Tse ^b

Y.T. Yu ^c

^a*School of Information Technology
Swinburne University of Technology
Hawthorn, Vic. 3122, Australia
E-mail address: tychen@it.swin.edu.au*

^b*Department of Computer Science
The University of Hong Kong
Pokfulam, Hong Kong
E-mail address: thtse@cs.hku.hk*

^c*Department of Computer Science
City University of Hong Kong
Kowloon Tong, Hong Kong
E-mail address: csytyu@cityu.edu.hk*

* Corresponding author.