APSEC 2012

The 19th Asia-Pacific *Software Engineering* Conference

Hong Kong
December 4–7, 2012

Sponsors

IEEE Hong Kong Section
Computer Society Chapter

ACM Hong Kong Chapter

Exhibitor

Special thanks to

Prof. T.H. Tse

for his excellent graphic designs of the conference logo as well as the cover and layouts of the front matter of the conference proceedings.

Message from the General Chairs
APSEC 2012

Welcome to the 19th Asia-Pacific Software Engineering Conference (APSEC 2012) in Hong Kong, a lively place with a beautiful harbor and landscape. Hong Kong is a city internationally known for its finance, shopping, and food with good mixes of eastern and western cultures. It is our pleasure to host APSEC for the second time in Hong Kong.

APSEC is a premier international conference in software engineering held annually in the Asia-Pacific region. Since 1994, it has been held in Tokyo, Brisbane, Seoul, Hong Kong, Taipei (twice), Takamatsu, Singapore, Macao, Gold Coast, Chiang Mai, Busan, Bangalore, Nagoya, Beijing, Penang, Sydney, and Ho Chi Minh City. The conference brings together researchers and practitioners to exchange their latest ideas on the advancement of software development, maintenance, and management.

APSEC has a rich program this year. It contains four keynote presentations, regular and emerging research tracks, an industry track, workshops, tutorials, and a postgraduate symposium. The presentations address the latest advancements in the engineering of software for the evolving world. The program each day is featured with a keynote on a major software engineering topic delivered by a world-renowned expert. The speakers are Dines Bjørner, ACM Fellow and Prof. Emeritus at DTU Informatics; Jeff Kramer, ACM Fellow and Senior Dean at Imperial College London; David S. Rosenblum, ACM Fellow and Director of the Felicitous Computing Institute of the National University of Singapore; and Barbara G. Ryder, ACM Fellow and Head of the Department of Computer Science at Virginia Tech.

We would not be able to organize this event without the contribution of time and hard work by many people. In particular, we would like to thank the keynote speakers, the chairs and members of the program committee, and the chairs and members of the organizing committee.

We wish you a fruitful conference and an enjoyable time in Hong Kong.

Shing-Chi Cheung and T.H. Tse
General Chairs, APSEC 2012

Message from the Program Chairs
APSEC 2012

Welcome to APSEC 2012! On behalf of the organizing committee and program committee, we enthusiastically welcome you to Hong Kong for the 19th Asia-Pacific Software Engineering Conference (APSEC 2012). APSEC is a series of leading international conferences on software engineering and technology held annually in the Asia-Pacific region. The objective is to promote and enhance software engineering research in the region. The Steering Committee of APSEC currently comprises prominent software engineering researchers from Australia, Hong Kong, Japan, Korea, Singapore, Taiwan, Thailand, India, China, Malaysia, and Vietnam.

The theme of APSEC 2012 is “Software Engineering for the Evolving World”. Software has been increasingly utilized to improve the quality of lives and, subsequently, to change the world in a myriad of unforeseeable ways. Software systems have evolved to meet the changing needs of society, while society has in turn evolved in response to the use of innovative and omnipresent software systems. The emergence of contemporary software for the evolving world has posed unprecedented challenges to software engineering professionals and the community. APSEC 2012 provides an excellent opportunity for software engineering researchers and practitioners around the world to exchange ideas on how to address these challenges.

APSEC 2012 offers a program that features research and industrial practice papers with a wide range of topics, focusing on different software engineering aspects. This year, APSEC 2012 received 200 submissions from 38 countries on all continents except Antarctica. Each paper was reviewed by at least 3 reviewers to evaluate its quality, especially on knowledge advancement in software engineering. After a rigorous peer review process, 51 regular papers and 48 short papers have been selected for presentation and inclusion in the conference proceedings. The regular papers include mature results. Short papers represent quality work that will spur discussions.

The technical program also contains four keynote addresses, three tutorials, an Emerging Research Track to discuss promising early research ideas, a Postgraduate Symposium, an Industry Track whose papers bridge the academia and industrial interests, and two workshops.

We take this opportunity to thank all the Track and Workshop Chairs and Program Committee members who served in these important positions to solicit papers, review papers, and organize the tracks. Without their hard work, we could not have such a quality conference. Special thanks are extended to the strong support from the APSEC Steering Committee and the Organizing Committee, as well as the contributions from student helpers. We also thank all the keynote speakers and authors for their contribution to the exciting technical program of APSEC 2012. Finally, we thank the generous support of the sponsors, IEEE Hong Kong Section Computer Society Chapter and ACM Hong Kong Chapter, who jointly organize the conference.

Karl R.P.H. Leung and Pornsiri Muenchaisri
Program Chairs, APSEC 2012

Message from the Emerging Research Track Chair APSEC 2012

This is the first time for the Asia-Pacific Software Engineering Conference (APSEC) series to introduce the Emerging Research Track. This introduction reflects the growing complexity of modern software, which demands new software engineering ideas to address challenges therein. In particular, papers of this track propose emerging research directions with tangible future work, define new research problems with outlines of solutions, and/or describe hot problems with novel ideas to address them.

We received 35 submissions, each of which consists of at most 4 pages and is formatted according the IEEE conference proceedings template. Each paper was peer-reviewed by two to four program committee members followed by a two-day online discussion so that all non-conflicting program committee members can express their comments and communicate among them. The decision on each paper was made after the corresponding discussions on the paper. All non-conflicting program committee members were able to view the review reports, discussion comments, and the paper decision via the online manuscript system. The full list of accepted papers was also distributed to all program committee members before the decision notifications were sent to the authors of all the submissions.

In this edition, eleven (11) papers were finally accepted to be included in the program of the APSEC 2012 Emerging Research Track. These papers feature new ideas from requirements and architecture to verification and maintenance, from data engineering and sampling to model-based approach and empirical studies, and from business processes to open-source software.

Finally, I would like to thank the authors, the program committee members, the organizing committee members for their contributions and efforts to make this track to be successful.

W.K. Chan
Emerging Research Track Chair, APSEC 2012

Message from the Chairs
SQAM 2012

The goal of the International Workshop on Software Quality and Management (SQAM 2012) is to bring together researchers and practitioners for discussion and presentation on the emerging aspects pertaining to software quality research, management techniques, best practices and lessons learned. The workshop provides a leading forum to present new ideas and to explore future directions for software quality and management.

We have accepted 9 papers from a pool of over 20 papers, covering various topics such as coupling metrics, maintenance effort, productivity analysis, data management, and quality construction.

Baowen Xu, Nanjing University, China
Hareton Leung, The Hong Kong Polytechnic University, Hong Kong
Chairs, SQAM 2012

Message from the Chairs
SATA 2012

The International Workshop on Software Analysis, Testing and Applications (SATA) 2012 provides opportunities for researchers to exchange views and discuss the latest developments on software analysis, testing and related areas. It also serves as an incubator for the community to work towards a particular research agenda for software analysis/testing techniques and applications.

The workshop has accepted 9 papers addressing the software engineering challenges in the application of these techniques in source code pre-processing, program structure simplification, resource leaks, bug fix prediction, and so on. These papers have covered a range of topics such as code analysis and program understanding, design patterns, model-driven software analysis, software testing, and software debugging.

Bo Jiang, Beihang University, China
Xinming Wang, The Hong Kong University of Science and Technology, Hong Kong
Chairs, SATA 2012

December 4 (Tuesday)

Time	Room			
	Sung I	Sung II	Ching	Ming I
08:30 – 09:30	Registration			
09:30 – 11:00	Postgraduate Symposium		SQAM Workshop I: Process and Quality (09:40 – 11:00)	SATA Workshop I: Testing (09:40 – 11:00)
11:00 – 11:30	Coffee Break			
11:30 – 12:30	Keynote Address: A Survey of Formal Methods in Software Development *Dines Bjørner*			
12:30 – 13:30	Lunch			
13:30 – 15:00	Tutorial *Dines Bjørner*	Tutorial *Mikio Aoyama and Shinobu Saito*	SQAM Workshop II: Maintenance and Tools (13:30 – 15:10)	Tutorial *Mike Mannion and Hermann Kaindl*
15:00 – 15:40	Coffee Break			
15:30 – 17:00	Tutorial *Dines Bjørner*	Tutorial *Mikio Aoyama and Shinobu Saito*	SATA Workshop II: Analysis (15:20 – 17:00)	Tutorial *Mike Mannion and Hermann Kaindl*
18:30 – 21:00	WELCOME RECEPTION			

December 5 (Wednesday)

Time	Room			
	Sung I	Sung II	Ching	Ming I
08:00 – 08:30	Registration			
08:30 – 09:00	OPENING CEREMONY			
09:00 – 10:00	Keynote Address: Whither Software Architecture? *Jeff Kramer*			
10:00 – 10:30	Coffee Break			
10:30 – 12:30	Session 1A: Formal Methods I	Session 2A: Requirements Engineering I	Session 3A: Software Engineering Environments and Tools	Session 4A: Code Analysis and Testing
12:30 – 13:30	Lunch			
13:30 – 15:00	Session 1B: Context-Aware and Intelligent Systems	Session 2B: Product-Line Software Engineering	Session 3B: Software Maintenance and Evolution I	Session 4B: Requirement Modelling and Analysis
15:00 – 15:30	Coffee Break			
15:30 – 17:30	Session 1C: Software Security, Reliability and Privacy	Session 2C: Software Verification and Validation	Session 3C: Software Project Management and Applications	Session 4C: Novel Techniques

December 6 (Thursday)

Time	Room			
	Sung I	Sung II	Ching	Ming I
08:00 – 08:30	Registration			
08:30 – 09:30	Keynote Address: Whither Software Engineering Research? *David S. Rosenblum*			
09:30 – 10:00	Coffee Break			
10:00 – 12:00	Session 5A: Empirical Software Engineering I	Session 6A: Requirements Engineering II	Session 7A: Formal Methods II	Session 8A: Software Analysis, Understanding and Quality Measurement
12:00 – 13:00	Lunch			
13:00 – 14:30	Session 5B: Empirical Software Engineering II	Session 6B: Software Systems and Management	Session 7B: Software maintenance and Evolution II	Session 8B: Software Architecture and Design
15:00 – 18:00	LOCAL TOUR			
18:30 – 22:00	CONFERENCE BANQUET			

December 7 (Friday)

Time	Room			
	Sung I	Sung II	Ching	Ming I
09:30 – 10:30	Keynote Address: Blended Program Analysis for Improving Reliability of Real-World Applications *Barbara G. Ryder*			
10:30 – 11:00	Coffee Break			
11:00 – 13:00	Session 9A: Formal Methods III	Session 10A: Emerging Ideas I	Session 11A: Software Process, Quality and Education	Session 12A: Empirical Software Engineering III
13:00 – 14:00	Lunch			
14:00 – 16:00	Session 9B: Formal Methods IV	Session 10B: Emerging Ideas II	Session 11B: Software Quality Assurance and Component-Based Systems	Session 12B: Web-Based Software Engineering
16:30 – 17:00	CLOSING CEREMONY			

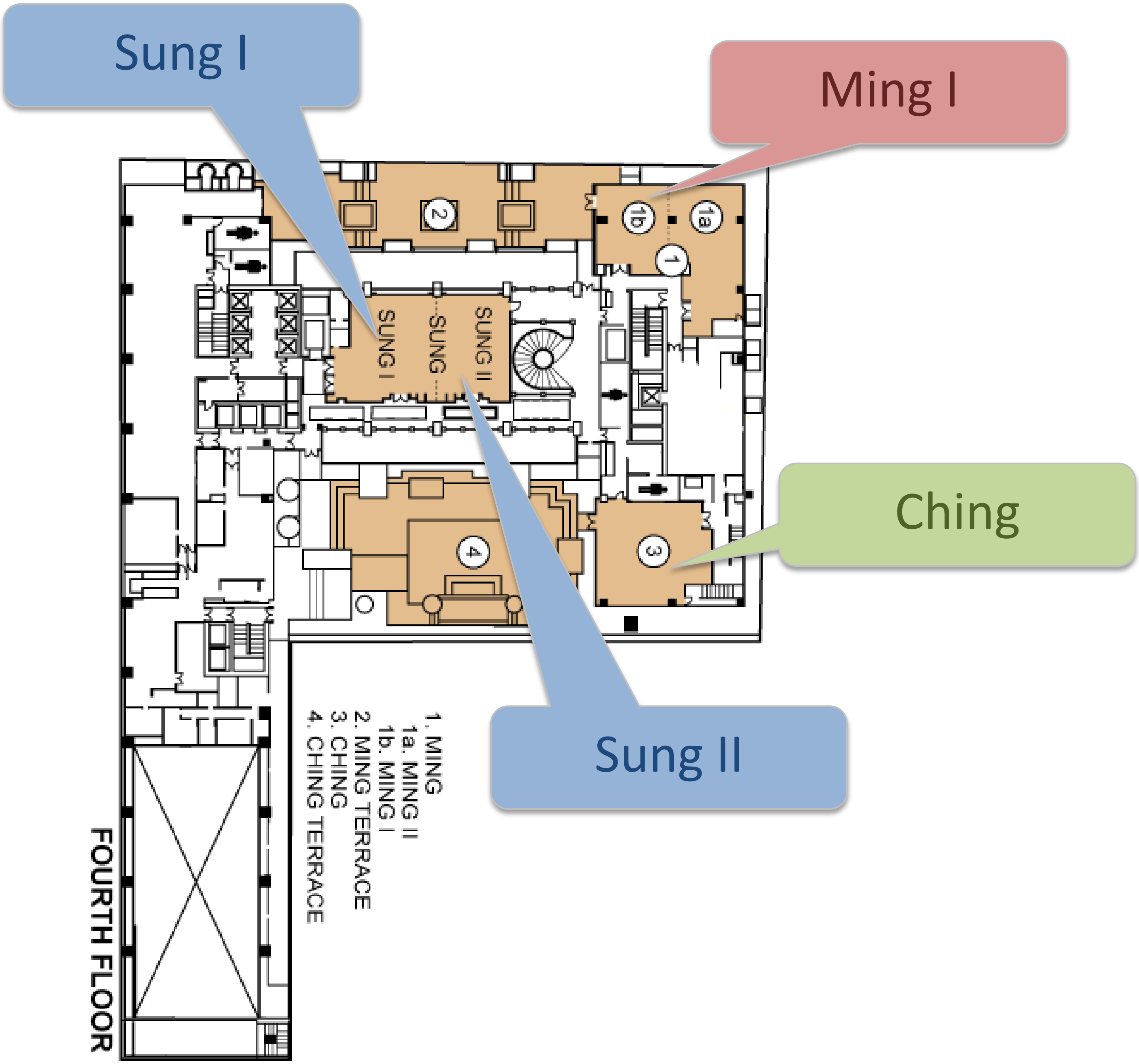
Conference Web Site

http://www.apsec2012.org

Conference Venue

4th Floor, Sheraton Hong Kong Hotel & Towers.

20 Nathan Road, Kowloon. Phone: (852) 2369 1111.



Keynote Address

Whither Software Architecture?

Jeff Kramer
Senior Dean
Department of Computing
Imperial College London

Abstract

Since its early beginnings in the 1980s, much has been achieved in the research field of software architecture. Among other aspects, this research has produced foundational work on the specification, analysis and component configuration of software architectures, including development of associated software tools. However, adoption of the research by industry has been largely methodological rather than based on precise specifications in architecture description languages (ADLs) or rigorously underpinned by formal models of behaviour and non-functional attributes.

Why is this? Why were the actual formalisms and tools not more widely adopted? Can we draw any lessons from this? In this talk, I hope to explore this further, drawing on my personal experience as a researcher in distributed software architectures.

I particularly hope to tickle the fancy of the younger members of our community, indicating the excitement of research, the benefits of belonging to a vibrant research community such as ours and of being an active contributor. For the more mature researchers, there will be some nostalgic memories combined with some inevitable stepping on toes. For both young and old, there will be some thoughts for research opportunities for both researchers and industry, particularly as the need for software evolution and for self-managing adaptive software systems becomes more urgent.

Dr. Jeff Kramer is the Senior Dean at Imperial College London. He was Head of the Department of Computing from 1999 to 2004 and Dean of the Faculty of Engineering from 2006 to 2009.

His research work is primarily concerned with software engineering, focusing on software architecture, behaviour analysis, the use of models in requirements elaboration and architectural approaches to adaptive software systems. He was a principal investigator of research projects that developed the CONIC and DARWIN architectural environments for distributed programming and of associated research into software architectures and their analysis.
Professor Jeff Kramer was Program Co-chair of ICSE ’99, Chair of the ICSE Steering Committee from 2000 to 2002, and General Co-chair of ICSE 2010 in Cape Town. He was Editor in Chief of *IEEE TSE* from 2006 to 2009, received the Most Influential Paper Award at ICSE 2003, and was awarded the 2005 ACM SIGSOFT Outstanding Research Award and the 2011 ACM SIGSOFT Distinguished Service Award. He is a co-author of books on Concurrency and on Distributed Systems and Computer Networks, and the author of over 200 journal and conference publications. He is a Fellow of the Royal Academy of Engineering, a Chartered Engineer, Fellow of the IET, Fellow of the ACM, Fellow of the BCS and Fellow of the City and Guilds of London Institute.

Keynote Address

Whither Software Engineering Research?

David S. Rosenblum
School of Computing
National University of Singapore

Abstract

In this talk I will present an assessment of the current state of research in our field and the ways in which the research has evolved over time. Unlike other talks that have been presented in recent years by prominent researchers in the field, this talk will place much less emphasis on my own personal reflections and opinions. Instead, I will attempt to apply a more objective, data-driven assessment of the field, with the data in this case being bibliographic data obtained from digital libraries. I will present a variety of quantitative characterizations of the research in our field and will apply them to compare our field with other fields. Among the characteristics I will analyze are the distribution and popularity of topics, and the impact of different topics in terms of citation counts.

David S. Rosenblum is a Professor in the Department of Computer Science of the School of Computing at the National University of Singapore. He directs the Felicitous Computing Institute and is a member of the NUS Graduate School for Integrative Sciences and Engineering.

He received his PhD in 1988 from Stanford University. From 1988 to 1996 he was a Member of the Technical Staff at AT&T Bell Laboratories (now AT&T Labs Research) in Murray Hill, New Jersey. After leaving Bell Labs, from 1996 to 2001 he was on the Faculty of the Department (now School) of Information and Computer Science at the University of California, Irvine. From 2001 to 2003 he was Chief Technology Officer and Principal Architect of PreCache, a startup company developing technology in the area of publish/subscribe networking. And from 2004 to 2011 he was Professor of Software Systems in the Department of Computer Science at University College London.

Professor Rosenblum’s research has addressed a wide range of problems spanning the breadth of the software development life cycle, including software specification, architecture, design, testing, analysis and maintenance, as well as problems underlying the processes that guide large software projects. His current research focuses on the scalability of architectures for large-scale software systems; probabilistic modeling and analysis of software systems; and the design and validation of mobile, context-aware adaptive systems for ubiquitous computing.

In 1997 he received a CAREER Award from the US National Science Foundation for his work on distributed component-based software, and from 2004–2009 he held a Wolfson Research Merit Award from the Royal Society. In the past decade he served as General Chair of the 2007 International Symposium on Software Testing and Analysis (ISSTA 2007) and Program Co-Chair of the 2004 International Conference on Software Engineering (ICSE 2004). He is an Associate Editor of the *ACM Transactions on Software Engineering and Methodology* (*ACM TOSEM*) and was previously an Associate Editor of the *IEEE Transactions on Software Engineering* (*IEEE TSE*). In 2002 he received the ICSE Most Influential Paper Award for his ICSE 1992 paper on assertion checking, and in 2008 he received the first ACM SIGSOFT Impact Paper Award with Alexander L. Wolf for their ESEC/FSE 1997 paper on Internet-scale event notification. He is a Fellow of the ACM, IEEE, BCS and IET, and he was the Chair of the ACM Special Interest Group on Software Engineering (ACM SIGSOFT), 2009–12.

Keynote Address

Blended Program Analysis for Improving Reliability of Real-world Applications

Dr. Barbara G. Ryder
J. Byron Maupin Professor of Engineering
Head – Department of Computer Science
Virginia Tech

Abstract

A new analysis paradigm, blended program analysis combines a dynamic representation of program calling structure with a static analysis applied to a region of that calling structure. Traditionally, compilers have used static analysis to enable semantics-preserving program transformations. Blended analysis supports tool-building to aid software developers improve the performance and reliability of their applications.

The initial instantiation of the blended paradigm addressed the issue of performance bottlenecks stemming from overuse of temporary objects in Java framework-intensive programs; this phenomenon is called *object churn* and is common in these applications. On the Trade benchmark, the approach helped locate a performance problem involving objects created at 9 distinct sites and as far away as 6 levels of calls, in a region which calls 223 distinct methods with a maximum call depth of 20. A newer instantiation is being used to find potential security vulnerabilities in JavaScript codes from popular websites.

This talk will present the blended analysis paradigm and our results of using it on Java and JavaScript codes.

**This research has been funded by the IBM Open Collaboration Research program and NSF-CCF 0811518.*

Dr. Barbara G. Ryder is Head of the Department of Computer Science at Virginia Tech, where she holds the J. Byron Maupin Professorship in Engineering. She received her PhD degree in Computer Science at Rutgers in 1982, and served on the faculty of Rutgers from 1982–2008. She also worked in the 1970s at AT&T Bell Laboratories in Murray Hill, NJ. Dr. Ryder's research interests lie in static and dynamic program analyses for object-oriented systems, focusing on usage in practical software tools for ensuring the quality and security of industrial-strength applications.

Dr. Ryder became a Fellow of the ACM in 1998, received the ACM President's Award in 2008, was selected as a CRA-W Distinguished Professor in 2004, and received the ACM SIGPLAN Distinguished Service Award in 2001. She has been an active leader in ACM (e.g., Vice President 2010–2012, Secretary-Treasurer 2008-2010; ACM Council 2000–2008; General Chair, FCRC 2003; Chair ACM SIGPLAN (1995–97)) and has served as a Member of the Board of Directors of the Computer Research Association (1998–2001). Dr. Ryder serves as a member of the ACM SIGSOFT ISSTA Steering Committee. She is a member of the NSF-funded ACM SIGSOFT IMPACT project that explores the impact of Software Engineering research on programming practice. As part of this project, she co-authored a paper with Mary Lou Soffa and Margaret Burnett, published in *ACM TOSEM* in October 2005, "The Impact of Software Engineering Research on Modern Programming Languages". Dr. Ryder has served as an editorial board member of *ACM Transactions on Programming Languages and Systems*, *IEEE Transactions on Software Engineering*, *IEEE Software*, *Software: Practice and Experience* and *Science of Computer Programming.*

Keynote Address

A Survey of Formal Methods in Software Development

Dines Bjørner, DTU Informatics
Fredsvej 11, DK-2840 Holte, Denmark

Abstract

The use of formal methods and formal techniques in industry is steadily growing. In this survey we shall characterise what we mean by software development and by a formal method; briefly overview a history of formal specification languages — some of which are: VDM (Vienna Development Method, 1974–..., [1]), Z (Z for Zermelo Fraenkel, 1980–..., [2]), RAISE (Rigorous Approach to Industrial Software Engineering, 1987–..., [3]) Event B (B for Bourbaki, 1990/2000–..., [4]) and Alloy [5]; and outline the basics of a formal development using, for example, RAISE: first developing a domain description D, then a requirements prescription R, and finally a software design S — showing (arguing or formally proving) that S, in the context of D satisfies (is correct with respect to) R. We shall then mention industries in Japan, Europe and USA which, in a number of projects, use formal methods; discuss what it takes for an industry to do so; discuss what education that candidates for these industries need, that is, which courses must be part of a BSc/MSc Software Engineering curriculum. Finally we shall comment on distinctions between formal methods and formal techniques; limitations of mono-language formalisations, hence need for multi-language formalisation (Petri Nets, MSC, StateChart, Temporal Logics); the sociology of university and industry acceptance of formal methods; the inevitability of the use of formal software development methods; while referring to seminal monographs and textbooks on formal methods.

[1] J. Fitzgerald and P. G. Larsen. *Modelling Systems: Practical Tools and Techniques in Software Development*. Cambridge University Press, Cambridge, UK, 1998.

[2] J. C. P. Woodcock and J. Davies. *Using Z: Specification, Proof and Refinement*. Prentice Hall International Series in Computer Science, Hemel Hempstead, Hertfordshire, UK, 1996.

[3] D. Bjørner. *Software Engineering, Vol. 1: Abstraction and Modelling, Vol. 2: Specification of Systems and Languages, Vol. 3: Domains, Requirements and Software Design*. Texts in Theoretical Computer Science, an EATCS Series. Springer, Berlin, Germany, 2006.

[4] J.-R. Abrial. *The B Book: Assigning Programs to Meanings and Modeling in Event-B: System and Software Engineering*. Cambridge University Press, Cambridge, UK, 1996 and 2009.

[5] D. Jackson. *Software Abstractions: Logic, Language, and Analysis*. The MIT Press, Cambridge, MA, USA, 2006.

Dines Bjørner was Professor of Computing Science at The Techn. Univ. of Denmark (DTU) 1976–2007 and is now a Professor Emeritus at DTU. He was with IBM R&D 1962–1976. At IBM Bjørner worked with Gene Amdahl (supercomputer design), John Backus (functional programming), E.F. Codd (relational databases) and at the IBM Lab., Vienna, Austria, where he worked in the small team that conceived VDM, the Vienna Development Method, the first ISO standardised formal method. Bjørner co-founded Dansk Datamatik Center 1979–1989, led many EU R&D projects including the formal specification of a semantics for Ada, the R&D of, and compilers for CHILL and Ada, and RAISE, the Rigorous Approach to Industrial Software Engineering. In 1991–1997 Bjørner was the founding and first UN Director of the Macau-based UNU-IIST, the UN University's International Institute for Software Technology. In 2006 Bjørner published a three volume book "Software Engineering" with Springer and in 2009 with Tsinghua University Press. He has published more than 120 papers, 6 books, edited 11 other books, is a member of Academia Europaea, an ACM Fellow, an IEEE Fellow, Member of the Russian Academy of Natural Science, received the John von Neuman Medal from Hungary (1993), the Masaryk Gold Medal (1996) and an honorary doctorate (2004) from the Masaryk Univ. of Brno, The Czech Republic, etc., and is a Royal Knight of the Danish Flag (1984).

Tutorial

A Survey of Domain Engineering

Dines Bjørner, DTU Informatics
Fredsvej 11, DK-2840 Holte, Denmark

Abstract

The practice of domain engineering as a precursor to requirements engineering in software development is gaining acceptance. In this tutorial we shall argue that before software, S, can be designed, one must understand its requirements, R, and that before requirements, can be prescribed, one must understand the application domain, D. Therefore, to us, software engineering consists of three major phases: domain engineering in which a domain description is developed; requirements engineering in which a requirements prescription is developed and related to the domain description; and software design in which a software specification is related to the former: the Software is correct with respect to the Requirements in the context of the Domain. In this tutorial we shall focus on domain engineering. Domain engineering, to recall, aims at producing a description of the domain: of its endurant ("data") and perdurant ("action, event and behaviour") entities. So a main activity of domain engineering is the analysis of endurant and perdurant entities. As for endurant entities we shall outline principles and techniques for describing how a domain is composed from parts, their values and types and how they relate to one another. We refer to this aspects of domain analysis as mereology. We shall, however briefly, also outline principles and techniques for describing perdurant entities: (actions, events and behaviours). The tutorial will start with the presentation of a non-trivial example: the domain description of a class of transport systems (road nets, rail nets, airline nets, shipping nets). And the tutorial will end with a discussion of how one can develop requirements from domain descriptions; how domain descriptions relate to ontology; issues of domain science: laws of domain descriptions; possible calculi of domain discoverers; who should be researching and developing domain descriptions; etcetera.

For extended tutorial notes see http://www2.imm.dtu.dk/˜dibj/dsae-a.pdf .

Dines Bjørner was Professor of Computing Science at The Techn. Univ. of Denmark (DTU) 1976–2007 and is now a Professor Emeritus at DTU. He was with IBM R&D 1962–1976. At IBM Bjørner worked with Gene Amdahl (supercomputer design), John Backus (functional programming), E.F. Codd (relational databases) and at the IBM Lab., Vienna, Austria, where he worked in the small team that conceived VDM, the Vienna Development Method, the first ISO standardised formal method. Bjørner co-founded Dansk Datamatik Center 1979–1989, led many EU R&D projects including the formal specification of a semantics for Ada, the R&D of, and compilers for CHILL and Ada, and RAISE, the Rigorous Approach to Industrial Software Engineering. In 1991–1997 Bjørner was the founding and first UN Director of the Macau-based UNU-IIST, the UN University's International Institute for Software Technology. In 2006 Bjørner published a three volume book "Software Engineering" with Springer and in 2009 with Tsinghua University Press. He has published more than 120 papers, 6 books, edited 11 other books, is a member of Academia Europaea, an ACM Fellow, an IEEE Fellow, Member of the Russian Academy of Natural Science, received the John von Neuman Medal from Hungary (1993), the Masaryk Gold Medal (1996) and an honorary doctorate (2004) from the Masaryk Univ. of Brno, The Czech Republic, etc., and is a Royal Knight of the Danish Flag (1984).

Tutorial

Requirements Engineering Based on REBOK (Requirements Engineering Body Of Knowledge) and its Practical Guide

Mikio Aoyama
Dep. of Software Engineering
Nanzan University
Seto, Japan
mikio.aoyama@nifty.com

Shinobu Saito
Research and Development Headquarters
NTT DATA Corporation
Tokyo, Japan
saitousnb@nttdata.co.jp

***Abstract*— We introduce a whole view of current status of requirements engineering based on REBOK (Requirements Engineering Body Of Knowledge) and its practical use with case studies.**

Keywords-requirements engineering, REBOK, BABOK, SWEBOK

I. Introduction

Requirements engineering is the most influential to success or failure of software development. Although the requirements engineering has been maturing with a large number of publications, there is no comprehensive BOK (Body Of Knowledge) to guide practitioners and researchers. REBOK (Requirements Engineering Body Of Knowledge) was developed to organize the body of knowledge on requirements engineering for practitioners [2, 3, 4].

II. Requirements Engineering Based on REBOK

A. *REBOK and Related BOKs*

There are a few BOKs related to requirements engineering. They include SWEBOK [1] and BABOK [5]. IREB(International Requirements Engineering Board) published a syllabus for a certification of CPRE(Certified Professional for Requirements Engineering)[6]. However, none of them can cover the whole knowledge area of requirements engineering. As illustrated in Figure 1, REBOK covers the whole requirements engineering from business requirements to software requirements, and from basic to advance level.

Table I summarizes eight knowledge areas of REBOK.

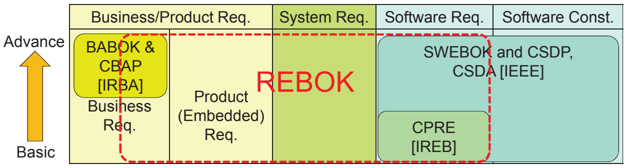


Figure 1. REBOK and Related BOKs

B. *Scope of Requirements*

As illustrated in Figure 2, REBOK defines the scope of requirements with three-layers of business, system, and software.

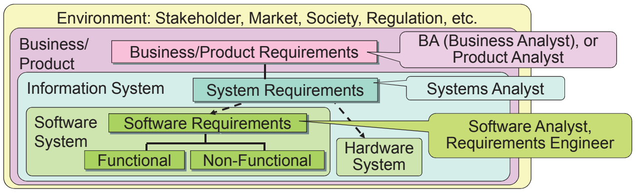

Environment: Stakeholder, Market, Society, Regulation, etc.
Business/ Product
Business/Product Requirements
BA (Business Analyst), or Product Analyst
Information System
System Requirements
Systems Analyst
Software System
Software Requirements
Software Analyst, Requirements Engineer
Hardware System
Functional
Non-Functional

Figure 2. Three Levels of Hierarchical Scopes of Requirements

C. *Requirements Engineering Process*

Figure 3 illustrates requirements engineering process. It configures four key knowledge areas of “elicitation”, “analysis”, “specification”, and “verification, validation and evaluation” in an incremental and iterative way. The process is consistent with ISO/IEC/IEEE 29148:2011, a standard of requirements engineering process [7].

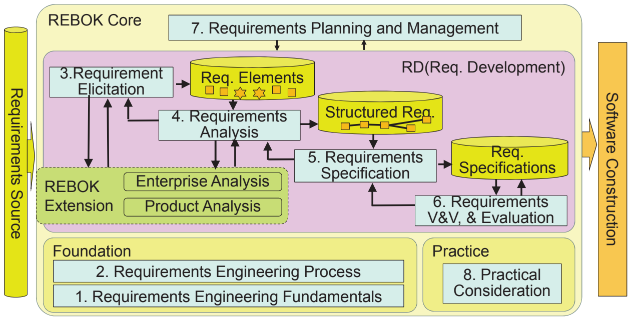

Requirements Source
REBOK Core
7. Requirements Planning and Management
RD(Req. Development)
3.Requirement Elicitation
Req. Elements
4. Requirements Analysis
Structured Req.
5. Requirements Specification
Req. Specifications
6. Requirements V&V, & Evaluation
REBOK Extension
Enterprise Analysis
Product Analysis
Foundation
2. Requirements Engineering Process
1. Requirements Engineering Fundamentals
Practice
8. Practical Consideration
Software Construction

Figure 3. REBOK Requirements Engineering Process

TABLE I. Eight Knowledge Areas of REBOK Core

KA	Definition
1. Requirements Engineering Fundamentals	Definition and essential properties on requirements.
2. Requirements Engineering Process	Concept and models of requirements engineering process.
3. Requirements Elicitation	Sources and techniques for requirements elicitation
4. Requirements Analysis	Techniques for analyzing requirements elicited
5. Requirements Specification	Specification techniques for requirements analyzed
6. Requirements Verification, Validation and Evaluation	Techniques validating requirements specification
7. Requirements Planning and Management	Properties, metrics and management techniques of requirements
8. Practical Consideration	Patterns and best practices for practicing requirements engineering

III. Empowering Requirements Elicitation

As the most influential requirements engineering process, REBOK empowers requirements elicitation process as illustrated in Figure 4. The process orchestrates the matured techniques in requirements elicitation.

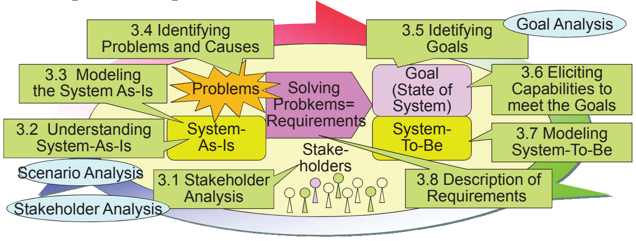

3.4 Identifying Problems and Causes
3.5 Idetifying Goals
Goal Analysis
3.3 Modeling the System As-Is
Problems
Solving Probkems= Requirements
Goal (State of System)
3.6 Eliciting Capabilities to meet the Goals
3.2 Understanding System-As-Is
System-As-Is
System-To-Be
3.7 Modeling System-To-Be
Stake-holders
Scenario Analysis
3.1 Stakeholder Analysis
3.8 Description of Requirements
Stakeholder Analysis

Figure 4. Requirements Elicitation Process

IV. Practical Guide

A. Many Ways to Use REBOK in Practice

To empower the organization and people with REBOK, there are many ways to apply REBOK in practice. Since REBOK covers the whole knowledge of requirements engineering, it is suggested to choose appropriate techniques from REBOK depending on the necessity at your practice.

(1) Introduction of requirements engineering concepts: you can read or teach the requirements engineering fundamentals.

(2) Introduction of requirements engineering to the organizational practice: you can teach REBOK core. A requirements-engineering workshop with real example is effective to learn.

(3) Improvement requirements engineering practice and personnel: you can use REBOK for improvement of practice and personnel as explained below.

(4) Education of students and practitioners: you can use REBOK as the foundation of curriculum of education and training.

B. Improving Requirements Engineering Practice

You can use REBOK in two ways of improvement of requirements engineering capability of both organizational practice and personnel as illustrated in Figure 5.

First, you can use REBOK as the reference model to evaluate current requirements engineering practice and reengineer it. You can find what is missing in current practice and what is needed to the RE-To-Be with REBOK.

Second, to make improved requirements engineering practice (RE-To-Be) effective, it is necessary to improve the capability of practitioners as well. You can use REBOK as the guideline to education and training practitioners.

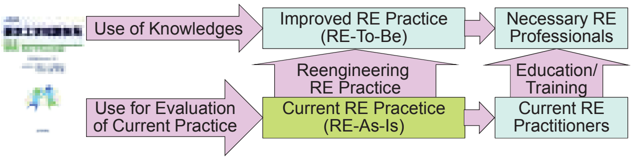

Use of Knowledges
Improved RE Practice (RE-To-Be)
Necessary RE Professionals
Reengineering RE Practice
Education/ Training
Use for Evaluation of Current Practice
Current RE Precetice (RE-As-Is)
Current RE Practitioners

Figure 5. Improvement of Practice and Professionals Based on REBOK

V. Practical Cases

Major Japanese IT companies are working hard to improve requirements engineering practice not only inside the companies but also at whole industry. Some companies publish their comprehensive requirements engineering methodologies as exemplified in Table II.

TABLE II. Major Requirements Engineering Methodologies

Company	Methodology
Fujitsu Limited	Tri-Shaping
Hitachi Solutions, Ltd.	HyThology
NTT DATA Corporation	TERASOLUNA [8]

Members of the companies are working for REBOK, too.

For example, TERASOLUNA is a comprehensive methodology, including requirements engineering, for developing open systems [8]. It is based on rich experiences in the systems development at NTT Data Corporation. TERASOLUNA has been applied to the systems development not only at the corporation but also at the customers. With consulting service from the corporation, the customer can develop own development standard based on TERASOLUNA.

REBOK is used for education and training, too. IPA (Information-technology Promotion Agency), an agency of Japanese government, is now developing an e-Leaning program based on REBOK. HyThology is developed based on REBOK. It is used for cultivating requirements engineering professionals at the whole company.

VI. Conclusion

REBOK is gaining a momentum in both practice and education. It is worth to apply REBOK in practice.

Acknowledgment

We thank REBOK WG team for their contributions to the development of REBOK, and JISA (Japan Information Technology Services Association) for continuous support.

References

[1] A. Abran, et al. (eds.), Guide to the Software Engineering Body of Knowledge, 2004 Ver., IEEE CS, http://www.swebok.org.

[2] M. Aoyama, et al., REBOK Manifest: Towards a Requirements Engineering Body Of Knowledge, Proc. IEEE RE 2010, IEEE CS, Sep.-Oct. 2010, pp. 383-384, doi: 10.1109/RE.2010.71.

[3] M. Aoyama, et al., A Model and Architecture of REBOK (Requirements Engineering Body Of Knowledge) and Its Evaluation, Proc. APSEC 2010, IEEE CS, Nov.-Dec. 2010, pp. 50-59, doi: 10.1109/APSEC.2010.16.

[4] M. Aoyama, et al.(JISA REBOK WG), Requirements Engineering Body Of Knowledge, V. 1.0, Kindaikagakusha, 2011[In Japanese].

[5] IIBA(Int'l Institute of Business Analysis), A Guide to the Business Analysis Body of Knowledge (BABOK Guide), V. 2.0, 2009.

[6] IREB, http://www.certified-re.de/en/.

[7] ISO/IEC/IEEE 29148:2011 Software and Systems Engineering – Life Cycle Processes – Requirements Engineering, 2011.

[8] NTT DATA Corporation, Support Services Launched for System Development Solution TERASOLUNA®, May 2009, http://www.nttdata.com/global/en/news-center/global/2009/052500.html.

Product Line Requirements Reuse based on Variability Management

Mike Mannion
Glasgow Caledonian University
Glasgow, UK
m.a.g.mannion@gcu.ac.uk

Hermann Kaindl
Vienna University of Technology
Vienna
kaindl@ict.tuwien.ac.at

***Abstract*—As organizations respond to changing environments new software products emerge as a compromise between customer requirements, extensions of existing products and commercial needs. Success emanates from the processes and technology used to capture, adapt and manage the deep knowledge of existing products or services. In this tutorial we describe how requirements for a product line can be captured, managed and reused to generate the requirements for innovative new products, and how requirements selections for new products are often constrained by the design of the existing product line architecture. A key technical issue is the efficient management of the commonality and variability of requirements between products. One approach is to establish a pool of reusable requirements and to construct the requirements for a new product by making a selection from the pool. A concern of this approach is the efficient and clean selection of a consistent combination of requirements. A consistent combination is one in which the requirements selected satisfy all constraints imposed by the pool of reusable requirements. Critical issues are the management of requirements variability across different products, the management of inter-dependencies between selection decisions from the pool, the constraints placed upon these selections by existing architectures and being able to manage each of these issues when the number of requirements is very large. We address these concerns, present results of using these techniques for real-world applications, and describe some software tools that can be used to support them.**

Keywords—component; requirements, reuse, variability

I. Introduction

In modern business, one significant challenge is to develop a product line which is flexible enough to be tailored to meet different customer demands quickly and at the right price. Examples include washing machines, financial products, and mobile phones. Many of these products are software-intensive systems and their customisation often necessitates a certain variability within the software. Exploiting the commonality and managing the variability often requires a successful software reuse initiative. The success of such an initiative depends on identifying a set of reuse processes and adapting them to suit an organisation's context.

In the following, we sketch our approach for addressing these issues. (For more details on some this, please see [1].) First, we deal with product line requirements engineering in general, followed by more specific accounts of product line requirements definition and derivation. Finally, we point to case studies.

II. PRODUCT LINE REQUIREMENTS ENGINEERING

Product Line Requirements Engineering begins after the task of scoping a domain of focus has been completed i.e., that the products that fall within the domain (and those that do not) have been identified. It covers the tasks of product line requirements definition and the derivation of a specific product from the product line requirements and has three processes: capturing product line requirements in a product line model, selecting requirements for a single product from the product line model, and verifying that the requirements selected satisfy the constraints within the product line model.

III. PRODUCT LINE REQUIREMENTS DEFINITION

Defining product line requirements means building a product line model in our approach. A product line model of requirements consists of a domain dictionary and a pool of numbered, atomic, natural language reusable requirements. It contains all the requirements in all the existing products in a product line. To assist with organising the information held about product line requirements in order to support the selection of application requirements, a requirements *metamodel* can be defined that includes common and variable requirements, their properties and the relationships between them. Building a product line model of requirements is a difficult process for several reasons. One technique that can help make this activity manageable is to identify separate stakeholder viewpoints and then to elicit, analyse, and compare and contrast the requirements in each product from each separate viewpoint which are documented in views.

IV. PRODUCT LINE REQUIREMENTS DERIVATION

A. Selecting Requirements

Given a product line model of requirements, it is possible to select the requirements for a new instance of the product line from this model. There are two primary methods of selection of requirements. *Free selection* allows a requirements engineer to browse a product line model and simply copy and paste a single requirement from anywhere in the model to the new product requirements. It does not use the constraints built into the model structure to guide model traversal and can lead to selection combinations being made that do not satisfy the constraints of the product line model, e.g. two mutually exclusive requirements can be selected or requirements that must be included can be omitted. In addition there can be an untenable number of choices making single product specifications time-consuming to select and check that they are free of errors.

Variation-point *selection* is grounded in using the constraints built into the product line model to drive selection and permits choices to be made only at discriminant points. This ensures that the choices produce a set of single product requirements that satisfy the constraints built into the product line model and reduces the time spent on specification.

B. Verifying Requirements

We can express the variability constraints of a product line model more formally using propositional logic. By writing a Boolean variable for each requirement and by defining a logical operator for each constraint relationship between requirements, a logical expression for the product line model can be developed, and used to verify selection decisions made from the product line model. This is achieved by assigning a Boolean value of TRUE or FALSE, depending on whether a requirement is selected or not, so that the logical expression can be evaluated to TRUE or FALSE. If and only if all constraints are fulfilled, the expression evaluates to TRUE.

V. CASE STUDIES

To evaluate this approach we present some examples and case studies from the fields of mobile telephony, spacecraft control and manufacturing.

VI. CONCLUSION

This work presents our approach to capturing and representing reusable requirements represented in natural language; the structure of a product line model of such reusable requirements; the selection of requirements from the product line model for single products; and the verification of the selections made from the product line model.

REFERENCES

[1] M. Mannion, H. Kaindl, Using Parameters and Variation Points for Product Line Requirements, *Systems Engineering*, Volume 11, Issue 1, Spring 2008, pp. 61–80.

Conference Day 1 (December 4, 2012)

Postgraduate Symposium

Time: 09:30 – 11:00
Room: Sung I
Chair: *Charles Zhang*

Deepak Garg, Amitava Datta, and Tim French	A Two-level Prioritization Approach for Regression Testing of Web Applications
Xuelin Zhang, Tian Zhang, and Xuandong Li	Query by Drawing Examples of UML Model
Binh T. Nguyen, Binh T. Ngo, and Tho T. Quan	A Memory-based Abstraction Approach to Handle Obfuscation in Polymorphic Virus
Yi Zhou, Mengxiang Lin, Yang Lu, and Xiaomei Hou	Towards Generating Local High-coverage Tests for Evolving Software

Keynote Address: A Survey of Formal Methods in Software Development

Speaker: *Dines Bjørner*, DTU Informatics, Denmark
Chair: *Jacky Keung*
Time: 11:30 – 12:30
Room: Sung

Tutorial: A Survey of Domain Engineering

Presenter: *Dines Bjørner*
Time: 13:30 – 15:00; 15:30 – 17:00
Room: Sung I

Tutorial: Requirements Engineering Based on REBOK (Requirements Engineering Body Of Knowledge) and Its Practical Guide

Presenter: *Mikio Aoyama and Shinobu Saito*
Time: 13:30 – 15:00; 15:30 – 17:00
Room: Sung II

Tutorial: Product Line Requirements Reuse Based on Variability Management

Presenter: *Mike Mannion and Hermann Kaindl*
Time: 13:30 – 15:00; 15:30 – 17:00
Room: Ming I

Welcome Reception

Time: 18:30 – 21:00

Conference Day 1 (December 4, 2012)

International Workshop on Software Quality and Management (SQAM Workshop) I — Process and Quality

Time: 09:40 – 11:00
Room: Ching
Chair: *Hareton Leung*

Shuib Basri and Rory V. O'Connor	Software Development Team Dynamics in SPI: A VSE Context
Miguel Nicolas Gubitosi, *Basava Raju M., and* *Allahbaksh Asadullah*	Metrics for Measuring Quality of Modularization of Scala Systems
Frank Philip Seth, *Erja Mustonen-Ollila, Ossi Taipale, and Kari Smolander*	Software Quality Construction: Empirical Study on the Role of Requirements, Stakeholders and Resources
Dongjin Yu, Peng Geng, and Wei Wu	Constructing Traceability between Features and Requirements for Software Product Line Engineering

International Workshop on Software Quality and Management (SQAM Workshop) II — Maintenance and Tools

Time: 13:30 – 15:10
Room: Ching
Chair: *Jacky Keung*

Jean-Vivien Millo and S. Ramesh	Relating Requirement and Design Variabilities
Siti Nor Asikin Kamalzaman, Sharifah Mashita Syed-Mohamad, Shahida Sulaiman, and Kamalzuhairi Z. Zamli	Supporting Maintenance of Web Applications Using User-Centered Technique
Raula Gaikovina Kula, *Kyohei Fushida, Norihiro Yoshida, and Hajimu Iida*	Experimental Study of Quantitative Analysis of Maintenance Effort Using Program Slicing-Based Metrics
Chunbai Yang, Changjiang Jia, *W.K. Chan, and Y.T. Yu*	On Accuracy-Performance Tradeoff Frameworks for Energy Saving: Models and Review
Hyun Cho, Jeff Gray, and Yu Sun	Quality-Aware Academic Research Tool Development

Conference Day 1 (December 4, 2012)

International Workshop on Software Analysis, Testing and Applications (SATA Workshop) I — Testing

Time: 09:40 – 11:00
Room: Ming I
Chair: *Bo Jiang*

Ziying Dai, Xiaoguang Mao, Yuhua Qi, and Kerong Ben	Light-Weight Test Oracles for Resource Leaks Based on Finalizers
Niusha Hakimipour, and Paul Strooper	Exploring an Approach to Model-Based Testing from Behavior Trees
Nihar Sharma, Anjaneyulu Pasala, and Rupa Kommineni	Generation of Character Test Input Data Using GA for Functional Testing
Feiyu Li and Yunzhan Gong	Memory Modeling-Based Automatic Test Data Generation for String-Manipulating Programs

International Workshop on Software Analysis, Testing and Applications (SATA Workshop) II — Analysis

Time: 15:20 – 17:00
Room: Ching
Chair: *Rubin Wang*

Liguo Yu, S. Ramaswamy, and A. Vaidyanathan	Understanding the Effects of Code Clones on Modularity in Software Systems
Akinori Ihara, Yasutaka Kamei, Akito Monden, Masao Ohira, Jacky Wai Keung, Naoyasu Ubayashi, and Ken-Ichi Matsumoto	An Investigation on Software Bug-Fix Prediction for Open Source Software Projects — A Case Study on the Eclipse Project
Yui Sasaki, Tomoya Ishihara, Keisuke Hotta, Hideaki Hata, Yoshiki Higo, Hiroshi Igaki, and Shinji Kusumoto	Preprocessing of Metrics Measurement Based on Simplifying Program Structures
Allahbaksh Asadullah, Basavaraju M., Ilan Stern, and Vasudev D. Bhat	Design Patterns Based Pre-processing of Source Code for Plagiarism Detection
Hongbo Zhou, Dahai Jin, and Yunzhan Gong	An Interval-Based Model for Detecting Software Defect Using Alias Analysis

Conference Day 2 (December 5, 2012)

Opening Ceremony

Chair: *S.C. Cheung*
Time: 08:30 – 09:00
Room: Sung

Keynote Address: Whither Software Architecture?

Speaker: *Jeff Kramer*, Senior Dean, Department of Computing, Imperial College London, UK
Chair: *S.C. Cheung*
Time: 09:00 – 10:00
Room: Sung

Session 1A: Formal Methods I

Time: 10:30 – 12:30
Room: Sung I
Chair: *Shaoying Liu*

Mark Anderson, Paul Montague, and Benjamin Long	A Context-Based Integrity Framework
Haiying Sun, Jing Liu, and Dehui Du	Integration of Safety Verification with Conformance Testing in Real-Time Reactive System
Longming Dong, Ji Wang, and Liqian Chen	Modular Heap Abstraction-Based Memory Leak Detection for Heap-Manipulating Programs
Suman Roy, Sidharth Bihary, and Jose Alfonso Corso Laos	A CSP-theoretic Framework of Checking Conformance of Business Processes

Session 1B: Context-Aware and Intelligent Systems

Time: 13:30 – 15:00
Room: Sung I
Chair: *Jun Han*

Jari Pääkkö, Mikko Raatikainen, Varvana Myllärniemi, and Tomi Männistö	Applying Recommendation Systems for Composing Dynamic Services for Mobile Devices
Ziyou Wang, Minghui Zhou, and Hong Mei	Towards an Adaptive Service Degradation Approach for Handling Server Overload

Session 1C: Software Security, Reliability and Privacy

Time: 15:30 – 17:30
Room: Sung I
Chair: *TL Wong*

Rui Song, Bixin Li, Xiaona Wu, Cuicui Liu, and Shanshan Qi	A Preference and Honesty Aware Trust Model for Web Services
Xiaona Wu, Bixin Li, Rui Song, Cuicui Liu, and Shanshan Qi	Trust-based Service Composition and Optimization
Shanshan Qi, Bixin Li, Cuicui Liu, Xiaona Wu, and Rui Song	A Trust Impact Analysis Model for Composite Service Evolution
Minh H. Tran, Alan Colman, Jun Han, and Hongyu Zhang	Modeling and Verification of Context-Aware Systems
Faqing Yang, Jean-Pierre Jacquot, and Jeanine Souquières	The Case for Using Simulation to Validate Event-B Specifications
Kaiping Liu, Hee Beng Kuan Tan, and Lwin Khin Shar	Semi-Automated Verification of Defense against SQL Injection in Web Applications

Conference Day 2 (December 5, 2012)

Session 2A: Requirements Engineering I

Time: 10:30 – 12:30
Room: Sung II
Chair: *Mikio Aoyama*

Jonathan Lee, Kuo-Hsun Hsu, Shin-Jie Lee, and Wen-Tin Lee	Discovering Early Aspects through Goals Interactions
Mo Li and Shaoying Liu	Automated Functional Scenarios-Based Formal Specification Animation
Rong Peng and Han Lai	DRE-specific Wikis for Distributed Requirements Engineering: A Review
Ziyan Xu, Haihong Zhao, Sheau Ling Tan, and Lin Liu	Service-Driven Migrating of Enterprise Information Systems: A Case Study

Session 2B: Product-Line Software Engineering

Time: 13:30 – 15:00
Room: Sung II
Chair: *Kyo-Chul Kang*

Gerald Holl, Paul Grünbacher, Christoph Elsner, and Thomas Klambauer	Supporting Awareness during Collaborative and Distributed Configuration of Multi Product Lines
Emmanuelle Rouillé, Benoît Combemale, Olivier Barais, David Touzet, and Jean-Marc Jézéquel	Leveraging CVL to Manage Variability in Software Process Lines
Stephen Creff, Joël Champeau, Arnaud Monégier, and Jean-Marc Jézéquel	Relationships Formalization for Model-Based Product Lines

Session 2C: Software Verification and Validation

Time: 15:30 – 17:30
Room: Sung II
Chair: *Bo Jiang*

J. Schönböck, G. Kappel, M. Wimmer, A. Kusel, W. Retschitzegger, and W. Schwinger	Debugging Model-to-Model Transformations
Masahiro Sakai, Kohei Maruchi, and Takeo Imai	Model-Checking C Programs against JML-like Specification Language
Linghao Zhang, Chang Xu, Xiaoxing Ma, Tianxiao Gu, Xuezhi Hong, Chun Cao, and Jian Lu	Resynchronizing Model-Based Self-Adaptive Systems with Environments
Adriaan Schmidt and Oliver Horst	Software-Based Online Monitoring of Cache Contents on Platforms without Coherence Fabric

Conference Day 2 (December 5, 2012)

Session 3A: Software Engineering Environments and Tools

Time: 10:30 – 12:30
Room: Ching
Chair: *Tetsuo Tamai*

Frank Padberg and Michael Mirold	An Experimentation Platform for the Automatic Parallelization of R Programs
Lili Mou, Ge Li, Zhi Jin, Yangyang Lu, and Yiyang Hao	Discovering Domain Concepts and Hyponymy Relations by Text Relevance Classifying Based Iterative Web Searching
Xiaowei Zhou, Wenbo Zhang, and Jianhua Zhang	Semi-static Detection of Runtime Type Errors in Component-Based Java Programs
Filippo Gioachin, Qianhui Liang, Yuxia Yao, and Bu-Sung Lee	Protego: In-Memory Version Control System in the Cloud

Session 3B: Software Maintenance and Evolution I

Time: 13:30 – 15:00
Room: Ching
Chair: *Sung Kim*

Cheng-Zen Yang, Chun-Chi Hou, Wei-Chen Kao, and Ing-Xiang Chen	An Empirical Study on Improving Severity Prediction of Defect Reports Using Feature Selection
Hongji Yang, Shang Zheng, William Cheng-Chung Chu, and Ching-Tsorng Tsai	Linking Functions and Quality Attributes for Software Evolution
Girish Maskeri, Deepthi Karnam, Sree Aurovindh Viswanathan, and Srinivas Padmanabhuni	Bug Prediction Metrics Based Decision Support For Preventive Software Maintenance

Session 3C: Software Project Management and Applications

Time: 15:30 – 17:30
Room: Ching
Chair: *Hongji Yang*

Saurabh Tiwari, Santosh Singh Rathore, Abhijeet Singh, Abhinav Singh, and Atul Gupta	An Approach to Generate Actor-Oriented Activity Charts from Use Case Requirements
Natalja Nikitina, and Mira Kajko-Mattsson	Process Model of Software Method Transition
Benoît Combemale, Xavier Crégut, and Marc Pantel	A Design Pattern to Build Executable DSMLs and Associated V&V Tools
Yiwei Zhang, Eric Lo, and Ben Kao	Evaluation Metric for Multiple-Bug Localization with Simple and Complex Predicates
Chaman Wijesiriwardana, Giacomo Ghezzi, Emanuel Giger, Atsushi Sawada, and Harald Gall	Dependency Based Approach for Software Analysis Web Services Replacement
Bilal Kanso, Marc Aiguier, Fréderic Boulanger, and Christophe Gaston	Testing of Component-Based Systems

Conference Day 2 (December 5, 2012)

Session 4A: Code Analysis and Testing

Time: 10:30 – 12:30
Room: Ming I
Chair: *Hareton Leung*

Priyanka Darke, Mayur Khanzode, Arun Nair, Ulka Shrotri, and R. Venkatesh	Precise Analysis of Large Industry Code
Norihiro Yoshida, Yoshiki Higo, Shinji Kusumoto, and Katsuro Inoue	An Experience Report on Analyzing Industrial Software Systems Using Code Clone Detection Techniques
Susumu Tokumoto, Tadahiro Uehara, Kazuki Munakata, Haruyuki Ishida, Toru Eguchi, and Masafumi Baba	Enhancing Symbolic Execution to Test the Compatibility of Re-engineered Industrial Software
Kazuki Munakata, Shoichiro Fujiwara, Susumu Tokumoto, and Tadahiro Uehara	Test Case Selection Based on Path Conditions of Symbolic Execution

Session 4B: Requirement Modelling and Analysis

Time: 13:30 – 15:00
Room: Ming I
Chair: *W.K. Chan*

Weam M. Farid	The NORMAP Methodology: Lightweight Engineering of Non-functional Requirements for Agile Processes
Naveen Kulkarni, Deepti Parachuri, Madhuri Dasa, and Abhishek Kumar	Automated Analysis of Textual Use-Cases: Does NLP Components and Pipelines Matter?
Shinobu Saito, Takashi Hoshino, Mutsuki Takeuchi, Masatoshi Hiraoka, Tsuyoshi Kitani, and Mikio Aoyama	Empirical Analysis of the Impact of Requirements Traceability Quality to the Productivity of Enterprise Applications Development

Session 4C: Novel Techniques

Time: 15:30 – 17:30
Room: Ming I
Chair: *Oliver Au*

Kiyoshi Nishikawa, Kenji Oki, and Akihiko Matsuo	SaaS Application Framework using Information Gateway Enabling Cloud Service with Data Confidentiality
Mehrdad Saadatmand and Mikael Sjödin.	Towards Accurate Monitoring of Extra-Functional Properties in Real-Time Embedded Systems
Shimpei Soda, Masahide Nakamura, Shinsuke Matsumoto, Shintaro Izumi, Hiroshi Kawaguchi, and Masahiko Yoshimoto	Implementing Virtual Agent as an Interface for Smart Home Voice Control
Tak Wah Kwan and Bryan Cheung	Determining and Selecting the Optimal Sets of Risk Treatments under Project Constraints

Conference Day 3 (December 6, 2012)

Keynote Address: Whither Software Engineering Research?
Speaker: *David S. Rosenblum*, School of Computing, National University of Singapore, Singapore
Chair: *Paul Strooper*
Time: 08:30 – 09:30
Room: Sung

Session 5A: Empirical Software Engineering I

Time: 10:00 – 12:00
Room: Sung I
Chair: *Shahida Binti Sulaiman*

Santosh Singh Rathore and Atul Gupta	Validating the Effectiveness of Object-Oriented Metrics over Multiple Releases for Predicting Fault Proneness
Najeeb Ullah and Maurizio Morisio	An Empirical Study of Reliability Growth of Open versus Closed Source Software through Software Reliability Growth Models
Hirohisa Aman	An Empirical Analysis of the Impact of Comment Statements on Fault-Proneness of Small-Size Module
Ferdian Thung, David Lo, and Lingxiao Jiang	Diffusion of Software Features: An Exploratory Study
Masateru Tsunoda, Akito Monden, Jacky Keung, and Kenichi Matsumoto	Incorporating Expert Judgment into Regression Models of Software Effort Estimation
Fan Yang-Turner, Lydia Lau, and Vania Dimitrova	A Model-Driven Prototype Evaluation to Elicit Requirements for a Sensemaking Support Tool

Session 5B: Empirical Software Engineering II

Time: 13:00 – 14:30
Room: Sung I
Chair: *Jacky Keung*

Diana Kirk and Ewan Tempero	Software Development Practices in New Zealand
Lijie Wang, Yanzhen Zou, Lu Fang, Bing Xie, and Fuqing Yang	An Exploratory Study of API Usage Examples on the Web
Syed Muhammad Ali Shah, Maurizio Morisio, and Marco Torchiano	An Overview of Software Defect Density: A Scoping Study

Conference Day 3 (December 6, 2012)

Session 6A: Requirements Engineering II

Time: 10:00 – 12:00
Room: Sung II
Chair: *Shamsul Sahibuddin*

Mahvish Khurum, Niroopa Uppalapati, and Ramya Chowdary Veeramachaneni	Software Requirements Triage and Selection: State-of-the-Art and State-of-Practice
Luxi Chen, Linpeng Huang, Chen Li, and Weichao Luo	Software Architecture Matching by Meta-model Extension and Refinement
Vladimir Estivill-Castro, René Hexel, and David A. Rosenblueth	Efficient Modelling of Embedded Software Systems and their Formal Verification
Haruhiko Kaiya, Shunsuke Morita, Shinpei Ogata, Kenji Kaijiri, Shinpei Hayashi, and Motoshi Saeki	Model Transformation Patterns for Introducing Suitable Information Systems
Chun Liu, Wei Zhang, Haiyan Zhao, and Zhi Jin	A Problem Oriented Approach to Modeling Feedback Loops for Self-Adaptive Software Systems
Abdul Babar and Bernard Wong	Capturing Strategic Business Requirements: An Exploratory Study

Session 6B: Software Systems and Management

Time: 13:00 – 14:30
Room: Sung II
Chair: *TL Wong*

Peng Zhou and Hareton K.N. Leung	An Integrated Risk Analysis Method Using Spatial Interpolation
Benjamin Morandi, Sebastian Nanz, and Bertrand Meyer	Who is Accountable for Asynchronous Exceptions?
Sunuk Park, Dae-Kyoo Kim, and Sooyong Park	Pattern-Based Model Transformation Using QVT

Session 7A: Formal Methods II

Time: 10:00 – 12:00
Room: Ching
Chair: *Pornsiri Muenchaisri*

Patrice Carle, Christine Choppy, Romain Kervarc, and Ariane Piel	Behavioural Analysis for Distributed Simulations
Bo Sun, Xiaohong Zhang, and Hao Wu	Extending Linden Scripting Language on OpenSim
Xiang Ling and Tom Maibaum	Architecture Based Development With DYNACOMM: Incorporating Dynamic Reconfiguration and Hierarchical Design into Community
Chen-Wei Wang	Calculating Preconditions for Parallel Workflows
Chen-Wei Wang	A Formal Approach for the Iterative Design of Behavioural Models
Min Zhang and Kazuhiro Ogata	Invariant-preserved Transformation of State Machines from Equations into Rewrite Rules

Conference Day 3 (December 6, 2012)

Session 7B: Software Maintenance and Evolution II

Time: 13:00 – 14:30
Room: Ching
Chair: *Rubin Wang*

Sangeeta Lal and Ashish Sureka	Comparison of Seven Bug Report Types: A Case-Study of Google Chrome Browser Project
Tianxiao Gu, Chun Cao, Chang Xu, Xiaoxing Ma, Linghao Zhang, and Jian Lu	Javelus: A Low Disruptive Approach to Dynamic Software Updates
Hadaytullah, Allan Gregersen, and Kai Koskimies	Pattern-Based Dynamic Maintenance of Software Systems

Session 8A: Software Analysis, Understanding and Quality Measurement

Time: 10:00 – 12:00
Room: Ming I
Chair: *Jin Song Dong*

Kunming Nie and Li Zhang	Software Feature Location Based on Topic Models
Dang Viet Dzung and Atsushi Ohnishi	A Verification Method of Elicited Software Requirements Using Requirements Ontology
Yang Zhao, Lingnan Song, *Gongxuan Zhang, and John Boyland*	Permission Assurance for Semantic Aspects of Design Patterns
Pei-Chi Chen, Ching-Chin Chern, and Chung-Yang Chen	Software Project Team Characteristics and Team Performance: Team Motivation as a Moderator
Huihong He, Zhiyi Ma, Xiang Li, Hongjie Chen, and Weizhong Shao	An Approach to Estimating Cost of Running Cloud Applications Based on AWS
K. Lochmann, D. Méndez Fernández, and S. Wagner	A Case Study on Specifying Quality Requirements Using a Quality Model

Session 8B: Software Architecture and Design

Time: 13:00 – 14:30
Room: Ming I
Chair: *Charles Zhang*

Nazri Kama and Faizul Azli	A Change Impact Analysis Approach for the Software Development Phase
Shaukat Ali and Tao Yue	Studying the Understandability of Aspect State Machines through the Weaving Activity
Weiqing Zhang, Vincent Leildé, *Birger Møller-Pedersen,* *Joël Champeau, and* *Christophe Guychard*	Towards Tool Integration through Artifacts and Roles

Local Tour

Time: 15:00 – 18:00

Conference Banquet

Time: 18:30 – 22:00
Venue: *Choi Fook Restaurant, Level 8, Fortune Metropolis, Hung Hom, Kowloon*

Conference Day 4 (December 7, 2012)

Keynote Address: Blended Program Analysis for Improving Reliability of Real-world Applications
Speaker: *Barbara G. Ryder*, Head – Department of Computer Science, Virginia Tech, USA
Chair: *T.H. Tse*
Time: 09:30 – 10:30
Room: Sung

Session 9A: Formal Methods III

Time: 11:00 – 13:00
Room: Sung I
Chair: *Shaoying Liu*

Weiqiang Kong, Leyuan Liu, Yoriyuki Yamagata, Kenji Taguchi, Hitoshi Ohsaki, and Akira Fukuda	On Accelerating SMT-based Bounded Model Checking of HSTM Designs
Yan Zhang and Tian Zhang	Hybrid Interface Automata
Juliana Bowles and Dulani Meedeniya	Parametric Transformations for Flexible Analysis
Jiexin Zhang, Yang Liu, Mikhail Auguston, Jun Sun, and Jin Song Dong	Using Monterey Phoenix to Formalize and Verify System Architectures

Session 9B: Formal Methods IV

Time: 14:00 – 16:00
Room: Sung I
Chair: *Hongji Yang*

Fauziah Binti Zainuddin and Shaoying Liu	An Approach to Low-fidelity Prototyping Based on SOFL Informal Specification
Min Zhang, Kazuhiro Ogata, and Kokichi Futatsugi	An Algebraic Approach to Formal Analysis of Dynamic Software Updating Mechanisms
Teng Long and Wenhui Zhang	Proving Liveness Property under Fairness Requirements
Xiaoquan Wu, Jun Wei, and Xu Wang	Debug Concurrent Programs with Visualization and Inference of Event Structure

Session 10A: Emerging Ideas I

Time: 11:00 – 13:00
Room: Sung II
Chair: *Zheng Zheng*

Shin Nakajima	Importance Sampling of Runtime Interference
Ana Dragomir and Horst Lichter	Model-Based Software Architecture Evolution and Evaluation
Xi Liu, Jianwen Su, and Xuandong Li	Proactive Enforcement of Data Consistency by Business Processes
Hyun Cho, Jeff Gray, and Eugene Syriani	Syntax Map: A Modeling Language for Capturing Requirements of Graphical DSML
Xin Yang, Raula Gaikovina Kula, Camargo Cruz Ana Erika, Norihiro Yoshida, Kazuki Hamasaki, Kenji Fujiwara, and Hajimu Iida	Understanding OSS Peer Review Roles in Peer Review Social Network (PeRSoN)
Izuru Kume, Masahide Nakamura, and Etsuya Shibayama	Toward Comprehension of Side Effects in Framework Applications as Feature Interactions

Conference Day 4 (December 7, 2012)

Session 10B: Emerging Ideas II

Time: 14:00 – 16:00
Room: Sung II
Chair: *Shin Nakajima*

Bharath Cheluvaraju, Kartikay Nagal, and Anjaneyulu Pasala	Mining Software Revision History Using Advanced Social Network Analysis
Mikhail Chalabine	Empowering Non-conservative Parallelizing Refactoring by Non-monotonic Reasoning
Chaman Wijesiriwardana, *Giacomo Ghezzi, and Harald Gall*	A Guided Mashup Framework for Rapid Software Analysis Services Composition
Cheng Gong, Zheng Zheng, *Yunqian Zhang, Zhenyu Zhang, and Yunzhi Xue*	Factorising the Multiple Fault Localization Problem: Adapting Single-Fault Localizer to Multi-fault Programs
Uaychai Chotjaratwanich and Chonlameth Arpnikanondt	A Visualization Technique for Metrics-Based Hierarchical Quality Models

Session 11A: Software Process, Quality and Education

Time: 11:00 – 13:00
Room: Ching
Chair: *Danny Poo*

Peng Zhou and Hareton K.N. Leung	A Stochastic Simulation Model for Risk Management Process
Simona Jeners, Horst Lichter, and Elena Pyatkova	Automated Comparison of Process Improvement Reference Models Based on Similarity Metrics
Jonathan Lee, Alan Liu, *Yu Chin Cheng, Shang-Pin Ma, and Shin-Jie Lee*	Execution Plan for Software Engineering Education in Taiwan
Abdehamid Abdelhadi Mansor, *Wan M.N. Wan Kadir, Toni Anwar, and Shamsul Sahibuddin*	Analysis of Adaptive Policy-Based Approach to Avoid Policy Conflicts
Balwinder Sodhi and T.V. Prabhakar	Assessing Platform Suitability for Achieving Quality in Guest Applications
Xiaorui Zhang, Øystein Haugen, and Birger Møller-Pedersen	Augmenting Product Lines

Session 11B: Software Quality Assurance and Component-Based Systems

Time: 14:00 – 16:00
Room: Ching
Chair: *Rubin Wang*

Shimul Kumar Nath, Robert Merkel, Man Fai Lau, and Tanay Kanti Paul	Towards a Better Understanding of Testing if Conditionals
Changhai Nie, Huayao Wu, *Yalan Liang, Hareton Leung,* *Fei-Ching Kuo, and Zheng Li*	Search Based Combinatorial Testing
Rogardt Heldal, Daniel Arvidsson, *and Fredrik Persson*	Modeling Executable Test Actors Exploratory Study Done in Executable and Translatable UML
Chen Li, Linpeng Huang, Luxi Chen, Weichao Luo, and Xu Li	PBDDR: Probe-Based Deadlock Detection and Recovery Strategy for Component-Based Systems
Shang-Pin Ma and Jonathan Lee	A Graph-Based Approach to Web Service Matchmaking
Wei Ye, Ruici Luo, Shikun Zhang, Xueyang Liu, and Wenhui Hu	BuOA: An Achitecture Style for Modular Web Applications

Conference Day 4 (December 7, 2012)

Session 12A: Empirical Software Engineering III

Time: 11:00 – 13:00
Room: Ming I
Chair: *Motoshi Saeki*

Hao Zhong, Ye Yang, and Jacky Keung	Assessing the Representativeness of Open Source Projects in Empirical Software Engineering Studies
Chris Lokan and Emilia Mendes	Investigating the Use of Duration-Based Moving Windows to Improve Software Effort Prediction
Claes Wohlin and Darja Šmite	Classification of Software Transfers
Akito Monden, Jacky Keung, Shuji Morisaki, Yasutaka Kamei, and Ken-Ichi Matsumoto	A Heuristic Rule Reduction Approach to Software Fault-proneness Prediction

Session 12B: Web-based Software Engineering

Time: 14:00 – 16:00
Room: Ming I
Chair: *Claes Wohlin*

Kaiping Liu and Hee Beng Kuan Tan	Automated Insertion of Exception Handling for Key and Referential Constraints
Cuicui Liu, Bixin Li, Shanshan Qi, Xiaona Wu, and Rui Song	Data Dependency Based Trust Evaluation for BPEL Processes
Navid Ahmadi, Mehdi Jazayeri, and Alexander Repenning	Engineering an Open-Web Educational Game Design Environment
Yonghong Chen, Xiwei Xu, and Liming Zhu	Web Platform API Design Principles and Service Contract

Closing Ceremony

Chair: *S.C. Cheung*
Time: 16:30 – 17:00
Room: Sung

Steering Committee
APSEC 2012

Chair

Paul Strooper, The University of Queensland, Australia

Members

Mikio Aoyama, Nanzan University, Japan
Doo-Hwan Bae, KAIST, Korea
William C. Chu, Tung Hai University, Taiwan
Jin Song Dong, National University of Singapore, Singapore
Jun Han, Swinburne University of Technology, Australia
Pankaj Jalote, IIT Delhi, India
Karl R.P.H. Leung, Hong Kong Institute of Vocational Education, Hong Kong
Huimin Lin, Chinese Academy of Sciences, China
Sooyong Park, Sogang University, Korea
Danny Poo, National University of Singapore, Singapore
Pornsiri Muenchaisri, Chulalongkorn University, Thailand
Shamsul Sahibuddin, Universiti Teknologi Malaysia, Malaysia
Tetsuo Tamai, The University of Tokyo, Japan
Tran Dan Thu, University of Science, VNU-HCM, Vietnam

Emeritus Members

Paul Bailes, The University of Queensland, Australia
Kyo-Chul Kang, POSTECH, Korea
Yong Rae Kwon, KAIST, Korea
Karl Reed, La Trobe University, Australia
Motoshi Saeki, Tokyo Institute of Technology, Japan

Organizing Committee

APSEC 2012

General Chairs

S.C. Cheung, The Hong Kong University of Science and Technology, Hong Kong
T.H. Tse, The University of Hong Kong, Hong Kong

Organizing Chairs

Victor Lee, City University of Hong Kong, Hong Kong
Y.T. Yu, City University of Hong Kong, Hong Kong

Emerging Research Track Chair

W.K. Chan, City University of Hong Kong, Hong Kong

Industry Track Chair

Hareton Leung, The Hong Kong Polytechnic University, Hong Kong

Workshop Chair

Jacky Keung, The Hong Kong Polytechnic University, Hong Kong

Tutorial Chair

Sung Kim, The Hong Kong University of Science and Technology, Hong Kong

Postgraduate Symposium Chair

Charles Zhang, The Hong Kong University of Science and Technology, Hong Kong

Treasurer

Philips Wang, Caritas Institute of Higher Education, Hong Kong

Secretary

Tak-lam Wong, The Hong Kong Institute of Education, Hong Kong

Publication Chair

Joe Yuen, City University of Hong Kong, Hong Kong

Local Arrangements Chair

Oliver Au, The Open University of Hong Kong, Hong Kong

Sponsorship Chair

Joseph Ng, Hong Kong Baptist University, Hong Kong

Publicity Chair

Eric Lo, The Hong Kong Polytechnic University, Hong Kong

Publicity Committee Members

Hyunsook Do, North Dakota State University, USA
Gordon Fraser, Saarland University, Germany
Edmonds Lau, Swinburne University of Technology, Australia
Jason Lee Hua Jie, The University of Melbourne, Australia
David Lo, Singapore Management University, Singapore
Soojin Park, Korea Sogang University, Korea

Program Committee
APSEC 2012

Chairs

Karl R.P.H. Leung, Hong Kong Institute of Vocational Education, Hong Kong
Pornsiri Muenchaisri, Chulalongkorn University, Thailand

Members

Muhammad Ali Babar, IT University of Copenhagen, Denmark
Toshiaki Aoki, JAIST, Japan
Mikio Aoyama, Nanzan University, Japan
Doo-Hwan Bae, KAIST, Korea
Arun Bahulkar, Tata Consultancy Services, India
Jongmoon Baik, KAIST, Korea
Paul Bailes, The University of Queensland, Australia
Umesh Bellur, Indian Institute of Technology Bombay, India
Reda Bendraou, INRIA Bretagne Atlantique Rennes, France
Heung-Seok Chae, Pusan National University, Korea
Joel Champeau, ENSTA Bretagne
Chih-Hung Chang, Hsiuping University of Science and Technology, Taiwan
Chunqing Chen, HP Labs Singapore, Singapore
Haiming Chen, Chinese Academy of Sciences, China
Ho-Jin Choi, KAIST, Korea
William Chu, Tung Hai University, Taiwan
Neville Churcher, University of Canterbury, New Zealand
Alan Colman, Swinburne University of Technology, Australia
Benoit Combemale, IRISA, Universite de Rennes 1, France
Son Dang, Ho Chi Minh City University of Technical Education, Vietnam
Philippe Dhaussy, ENSIETA, France
Tien Dinh, University of Science, VNU-HCM, Vietnam
Elsa Estevez, UNU-IIST, Macau
Ian Gorton, The Pacific Northwest National Laboratory, USA
Jun Han, Swinburne University of Technology, Australia
Shinpei Hayashi, Tokyo Institute of Technology, Japan
Ian J. Hayes, The University of Queensland, Australia
Liguo Huang, Southern Methodist University, USA
Hajimu Iida, NAIST, Japan
Tomasz Janowski, UNU-IIST Center for Electronic Governance, Macau
Mehdi Jazayeri, University of Lugano, Switzerland
Zhi Jin, Peking University, China
Rushikesh Joshi, Indian Institute of Technology Bombay, India
Wan Mohd Nasir Kadir, Universiti Teknologi Malaysia, Malaysia
Kyo-Chul Kang, POSTECH, Korea

Shanika Karunasekera, The University of Melbourne, Australia
Jacky Keung, The Hong Kong Polytechnic University, Hong Kong
Takashi Kobayashi, Nagoya University, Japan
Ralf-Detlef Kutsche, Technische Universität Berlin, Germany
Man Fai Lau, Swinburne University of Technology, Australia
Jonathan Lee, National Central University, Taiwan
Hareton Leung, The Hong Kong Polytechnic University, Hong Kong
Xiaoshan Li, University of Macau, Macau
Yuan-Fang Li, Monash University, Australia
Deron Liang, National Central University, Taiwan
Huai Liu, Swinburne University of Technology, Australia
Jenny Liu, The Pacific Northwest National Laboratory, USA
Shaoying Liu, Hosei University, Japan
Yang Liu, National University of Singapore, Singapore
Eric Lo, The Hong Kong Polytechnic University, Hong Kong
Ali Afzal Malik, UCP, Pakistan
Larissa Meinicke, The University of Queensland, Australia
Akito Monden, NAIST, Japan
Takako Nakatani, University of Tsukuba, Japan
Cuong Nguyen, International University, VNU-HCM, Vietnam
Viet Ha Nguyen, Coltech, Vietnam
Vu Nguyen, USC, Vietnam
Iulian Ober, University of Toulouse, IRIT, France
Leon Osterweil, UMass Amherst, USA
Sooyong Park, Sogang University, Korea
Christian Percebois, IRIT, France
Mauro Pezzé, University of Lugano, Switzerland
Danny Poo, National University of Singapore, Singapore
Pak Lok Poon, The Hong Kong Polytechnic University, Hong Kong
Tho Quan, University of Technology – Ho Chi Minh City, Vietnam
S. Ramesh, General Motors R&D, India
Karl Reed, La Trobe University, Australia
Motoshi Saeki, Tokyo Institute of Technology, Japan
Shamsul Sahibuddin, Universiti Teknologi Malaysia, Malaysia
A. S. M. Sajeev, University of New England, Australia
Jean-Guy Schneider, Swinburne University of Technology, Australia
Bikram Sengupta, IBM, India
Graeme Smith, The University of Queensland, Australia
Padmanabhuni Srinivas, Infosys, India
Jim Steel, The University of Queensland, Australia
Paul Strooper, The University of Queensland, Australia
Shahida Sulaiman, Universiti Sains Malaysia, Malaysia
Jing Sun, The University of Auckland, New Zealand
Jun Sun, Singapore University of Technology and Design, Singapore
Ashish Sureka, Indraprastha Institute of Information Technology (IIIT), India
Stanley Sutton, IBM T. J. Watson Research Center, USA

Prabhakar Tadinada, Indian Institute of Technology Kanpur, India
Tetsuo Tamai, The University of Tokyo, Japan
Hee Beng Kuan Tan, Nanyang Technological University, Singapore
Ewan Tempero, The University of Auckland, New Zealand
Dan Thu Tran, University of Science, VNU-HCM, Vietnam
Hanh Nhi Tran, ENSTA-Bretagne, France
Minh-Triet Tran, FIT-HCMUS, Vietnam
Hai H. Wang, University of Aston, UK
Hironori Washizaki, Waseda University, Japan
Jun Wei, Institute of Software, Chinese Academy of Sciences, China
Eric Wong, University of Texas at Dallas, USA
Peng Wu, Institute of Software, Chinese Academy of Sciences, China
Qiwen Xu, University of Macau, Macau
Hongji Yang, De Montfort University, UK
Yang Ye, Institute of Software, Chinese Academy of Sciences, China
Kamal Z. Zamli, Universiti Sains Malaysia, Malaysia
Hongyu Zhang, Tsinghua University, China
Jian Zhang, Institute of Software, Chinese Academy of Sciences, China
Zhenyu Zhang, Institute of Software, Chinese Academy of Sciences, China
Didar Zowghi, University of Technology, Sydney, Australia

Emerging Research Track
Program Committee
APSEC 2012

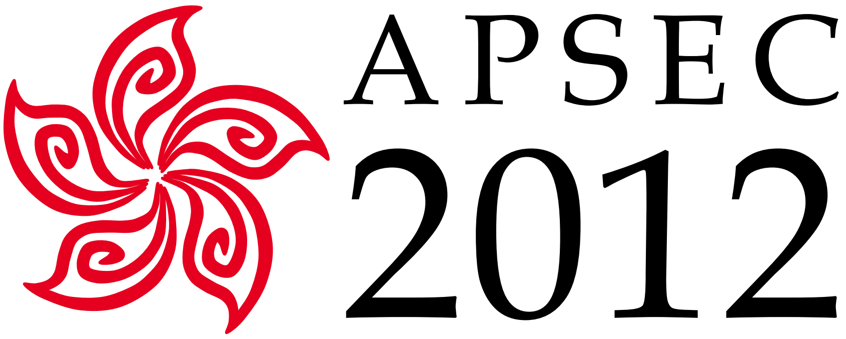
Chair

W.K. Chan, City University of Hong Kong, Hong Kong

Members

Rui Abreu, University of Porto, Portugal
Muhammad Ali Babar, IT University of Copenhagen, Denmark
Tadashi Dohi, Hiroshima University, Japan
Yann-Gaël Guéhéneuc, École Polytechnique de Montréal, Canada
Atul Gupta, Indian Institute of Information Technology, Design and Manufacturing, India
Shinpei Hayashi, Tokyo Institute of Technology, Japan
Pao-Ann Hsiung, National Chung Cheng University, Taiwan
Bo Jiang, Beihang University, China
Gregory Kapfhammer, Allegheny College, USA
David Lo, Singapore Management University, Singapore
Wes Masri, American University of Beirut, Lebanon
Mercedes Merayo, Universidad Complutense de Madrid, Spain
Raman Ramsin, Sharif University of Technology, Iran
Juha Savolainen, Danfoss, Denmark
Sara Sprenkle, Washington & Lee University, USA
Paul Strooper, University of Queensland, Australia
Linzhang Wang, Nanjing University, China
Eric Wong, University of Texas at Dallas, USA
Franz Wotawa, Graz University of Technology, Austria
Dianxiang Xu, Dakota State University, USA
Jianjun Zhao, Shanghai Jiao Tong University, China
Zhiquan Zhou, University of Wollongong, Australia

SEE YOU AT APSEC 2013



About the Logo

- The logo is rendered by the stylized characters “a p s e c”.
- It symbolizes the bauhinia blakeana, which is the emblem of Hong Kong and used on its flag.
- The logo is styled after ancient Chinese totems.

About the Designer

Prof. T.H. Tse is well known for his logo designs for non-government organizations, such as the Hong Kong Joint Council for People with Physical Disabilities, the SAHK, the Jockey Club Arts Accessibility Service Centre, the CereCare Foundation, Hong Kong Abilympics 2007, the Best Practice Award logo for the Hong Kong Council of Social Service, the 50th anniversary logo for SAHK, the 25th and 30th anniversary logos for the Hong Kong PHAB Association, and the 10th anniversary logo for Rehabilitation Alliance Hong Kong. He was also the designer of the logo of the International Conference on Quality Software (QSIC). He was awarded first prize for his logo design in the HKU Visions and Missions retreat, and a special prize for the open logo competition of the Hong Kong Computer Society. Please refer to http://www.cs.hku.hk/~tse/logos.html for more details.

Prof. Tse is a professor in computer science at The University of Hong Kong.