Introduction

William Cheng-Chung Chu

Tunghai University, Taiwan
E-mail: cchu@thu.edu.tw

Huaglory Tianfield

Glasgow Caledonian University, Scotland
E-mail: h.tianfield@gcal.ac.uk

Hongji Yang

De Montfort University, England
E-mail: hyang@dmu.ac.uk

Jianjun Zhao

Shanghai Jiao Tong University, China
E-mail: zhao-jj@cs.sjtu.edu.cn

Hong Zhu

Oxford Brookes University, England
E-mail: hzhu@brookes.ac.uk

Biographical notes: William Cheng-Chung Chu is a Professor of the Department of Computer Science and Information Engineering and the Director of Software Engineering and Technologies Center. He had served as the Dean of Research and Development office at Tunghai University from 2004 to 2007, Taiwan. He is serving as the Associate Editor for *Journal of Software Maintenance and Evolution (JSME)* and *Journal of Systems and Software (JSS)*. He has edited several books and published over 100 referred papers and book chapters, as well as participating in many international activities, including organising international conferences.

Huaglory Tianfield is the Professor of Distributed Systems at Glasgow Caledonian University, UK since March 2001. Previously he held research or academic positions in China, Germany, France and England, UK. He is a worldwide propeller for the new inter-disciplinary direction combining Systems Science with Computer Science. He is the Chair of the IEEE Systems, Man, and Cybernetics Society's Technical Committee on Self-Organization and Cybernetics for Informatics, Editor-in-Chief of *International Transactions on Systems Science and Applications*, Editor-in-Chief of *System and Information Sciences Notes*, Executive Editor of *Multiagent and Grid Systems – An International Journal*. He has served as (Co-)Chair or IPC member for over 80 international conferences and symposia.

Hongji Yang is Head of Computer Science Division at Schooling of Computing, De Montfort University, England and leads the Software Evolution and Re-engineering Group. His general research interests include software engineeering and distributed computing. He served as a Programme Co-Chair at IEEE International Conference on Software Maintenance 1999 (ICSM'99) and is serving as the Programme Chair at IEEE Computer Software and Application Conference 2002 (COMPSAC'02).

Jianjun Zhao is a Professor of Computer Science at Shanghai Jiao Tong University, China. He received the BS and PhD in Computer Science from Tsinghua University, China and Kyushu University, Japan. He was a coorganiser of several workshops including the International Workshop on Aspect-Oriented Software Development (WAOSD 2004) and Asian Workshop on Aspect-Oriented Software Development (AOAsia 2005, 2006, and 2007). He was also served on program committees of International Conference on Aspect-Oriented Software Development (AOSD 2003, 2004, and 2005). His research interests include aspect-oriented software engineering, especially aspect-oriented software testing, refactoring, and maintenance.

Hong Zhu is a Professor of Computer Science at Oxford Brookes University, where he chairs the Applied Formal Methods Research Group. He obtained his BSc, MSc and PhD Degrees in Computer Science from Nanjing University, China, in 1982, 1984 and 1987, respectively.

He was with Nanjing University from 1987 to 1998. He was a research fellow at Brunel University and the Open University, UK, from 1990 to 1994. He joined Oxford Brookes University in 1998. His research interests are in the area of software engineering, including software development methodology, software testing, agent technology, automated software development tools.

This special issue presents a collection of papers on two of the commonly discussed software engineering techniques currently: agent-oriented and aspect-oriented software methods.

The theme of the papers in the agent half of the issue is the development of agent-oriented software methodology for internet-based computing. The internet and the web have developed into the main platform of software applications in a wide range of domains. Consequently, software engineering is confronted with a number of challenges, such as dealing with service-oriented computing, dynamic integration of autonomous components, distributed and mobile computing, etc. Agent technology has long been perceived as a viable solution for large complex applications in dynamic environments. Recent years have seen a rapid growth of research in agent-oriented methodology. However, current agent-based systems are mostly developed in ad hoc methods without sound methodologies and effective language and tool supports. This has hampered the wide adoption of agent technology in large complex applications and more broadly the IT industry.

The theme of the papers in the aspect half of the issue is the ways in which aspect-oriented techniques are impacting software evolution, which is an essential part of the software project life and often costs up to 70% of a software project budget and starts at discussion concerns in software development. Concern became a useful concept for software system development initially in terms of decomposing a software system into smaller, more manageable and comprehensible modules. A concern includes a property or an area of interest ranging from high-level notions to low-level notions. It can be functional or non-functional. Concerns can be further grouped into aspect orientation, subject orientation, feature orientation and have facilitated software development and therefore naturally how they are impacting software evolution is worth studying academically, because separating concerns can be thought as a technique with potential, e.g., in identifying, encapsulating and manipulating only those parts of software that are relevant to a particular concept, goal, or purpose in understanding an existing system. Aspect oriented techniques attempt to abstract out features common to many parts of the code beyond simple functional modules and thereby to improve the quality of software through enabling clean modularisation of crosscutting concerns. Its feature is to provide some explicit mechanisms to modularly represent and separate the crosscutting concerns in software systems.

This special issue addresses research topics that can normally be grouped into conceptual model, techniques and methods/methodology, languages and tools, and experiences and empirical studies. The conceptual model is the foundation of development and evolution methodology for the analysis, design and implementation of software systems. Techniques, methods/methodologies topics include various development, validation and verification, and redevelopment techniques in the development and evolution of software systems. Languages and tools can play a significant role and experiences and empirical studies cover experiences and research results in developing and evolving various kinds of software systems in all types of applications. Table 1 shows how these issues are addressed by the papers in this special issue.

Table 1 Topics covered by the papers in the special issue

	Paper no.										
Topic	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*
Conceptual model			√	√	√	√					√
Methods and techniques	√	√					√	√	√	√	
Languages and tools		√	√	√						√	
Empirical study and experiments	√				√	√	√	√	√		√

The special issue originated from a collection of papers on aspect-oriented software evolutions and they were joined by papers selected and expanded from the *Second International Workshop on Agent-Oriented Software Development Methodology (AOSDM), collocated with the International Conference on Software Engineering and Knowledge Engineering (SEKE)*, San Francisco Bay, USA, July 2006. All the papers are reviewed by at least by one of the guest editors. Acknowledgements also go to additionally invited reviewers.