Evolving Critical Systems
Mike Hinchey
University of Limerick, Ireland

Abstract: Increasingly software can be considered to be critical, due to the business or other functionality which it supports. Upgrades or changes to such software are expensive and risky, primarily because the software has not been designed and built for ease of change. Expertise, tools and methodologies which support the design and implementation of software systems that evolve without risk (of failure or loss of quality) are essential. We address a research agenda for building software in computer-based systems that (a) is highly reliable and (b) retains this reliability as it evolves, either over time or at run-time and illustrate this with a complex example from the domain of space exploration.

Biographical Information: Mike Hinchey is Director of Lero—the Irish Software Engineering Research Centre and Professor of Software Engineering at University of Limerick, Ireland. Prior to joining Lero, Professor Hinchey was Director of the NASA Software Engineering Laboratory; he continues to serve as a NASA Expert. In 2009 he was awarded NASA’s Kerley Award as Innovator of the Year. Hinchey holds a B.Sc. in Computer Systems from University of Limerick, an M.Sc. in Computation from University of Oxford and a PhD in Computer Science from University of Cambridge. The author/editor of more than 15 books and over 200 articles on various aspects of Software Engineering, at various times Hinchey previously held positions as Full Professor in Australia, UK, Sweden and USA. He is a Chartered Engineer, Chartered Engineering Professional, Chartered Mathematician and Charted Information Technology Professional, as well as a Fellow of the IET, British Computer Society and Irish Computer Society. He is Vice President of IFIP (International Federation for Information Processing) and Chair of the IFIP Technical Assembly.
Modeling and Verification of Concurrent Systems with MSVL
Zhenhua Duan
Xidian University, China

Abstract: The reliability and security of software and hardware systems as well as communication protocols are significant at present. However, how to ensure the reliability and security of these systems still remains challenging. This talk presents a formal specification and verification approach for concurrent systems based on a Modeling Simulation and Verification Language (MSVL). MSVL is a parallel programming language and an executable subset of Projection Temporal Logic (PTL). The talk is organized in 6 parts: part 1 briefly introduces the syntax and semantics of MSVL including statements, data types, function callings, and a framing technique in MSVL. Part 2 presents a supporting toolkit MSV. It consists of a modeling tool, simulation tool, and several verification tools as well as several translators. In particular, a unified model checker, abstract model checker and theorem prover are available. Further, to verify C programs and Verilog descriptions of hardware and embedded systems, translators from C and Verilog to MSVL programs have been developed. Part 3 proposes a modeling approach for a system based on Normal Form Graph (NFG). Part 4 devotes to simulation of a system modeled by an MSVL program. Part 5 concentrates on verification approaches including model checking and theorem proving. In part 6, a Cylinder Computation Model (CCM) is presented so that many core parallel computing systems can be modeled and verified.

Biographical Information: Zhenhua Duan(段振华) is a professor in computer science at Xidian University. He obtained his PhD degree at the University of Newcastle upon Tyne and the University of Sheffield in 1996 and 1997 respectively. He is a senior member of IEEE, invited reviewer of "Mathematical Review" and editor of international journal, Discrete Mathematics, Algorithm and Application. He was also co-chairs of several international conferences and PC members of over 30 international conferences. His research interests are in the fields of reliability and security of software systems, embedded systems and communication protocols. In particular, he is currently devoting to verifying software and detecting malware using model checking approach.
Testing-Based Formal Verification: A New and Practical Approach for Software Quality Assurance
Shaoying Liu
Hosei University, Japan

Abstract: Specifications and programs for complex computer systems inevitably contain bugs and their identification and elimination are crucial for the assurance of the quality of the software artifacts. Formal verification can be used to prove that a software artifact possesses desired properties such as consistency and correctness, but their application to the faulty specifications and programs usually fail. Testing is a practical technique that uses selected data to detect bugs but is usually unable to establish the desired properties.

In this talk, a new and practical approach known as testing-based formal verification (TBFV) is first introduced and its application to verifying properties of specifications and programs are then discussed. This approach is expected to strike an effective balance between formal verification and testing by utilizing their advantages and avoiding their weaknesses. The underlying principle is to use well selected data to test whether desired properties expressed as logical formula do not hold and the result can help the developer to determine the quality of the artifact. The biggest advantage of TBFV over the testing and formal verification approaches is the great potential of full automation with the effect of approximation to formal proof in most cases.

Biographical Information: Shaoying Liu (刘少英) is a Professor of Software Engineering at Hosei University, Japan. His research interests include formal engineering methods, specification-based program inspection and testing, and intelligent software engineering environments. He has published a book titled Formal Engineering for Industrial Software Development Using the SOFL Method (Springer-Verlag), 6 edited conference proceedings, and more than 150 academic papers in refereed journals and international conferences. He proposed to use the terminology of “Formal Engineering Methods” in 1997, and has established Formal Engineering Methods as a research area based on his extensive research on the Structured Object-Oriented Formal Language (SOFL) and the development of the ICFEM conference series since 1997. In recent years, he has been serving as the Steering Committee Chair for ICFEM conferences, and on the editorial board for the Journal of Software Testing, Verification and Reliability (STVR). He is a fellow of the British Computer Society, a senior member of IEEE, and member of the Japan Society for Software Science and Technology.