Preface

The 2nd School on Engineering Trustworthy Software Systems (SETSS 2016) was held during March 28 – April 2, 2016, at Southwest University, Chongqing, China. It was aimed at PhD and Master students in particular, from around China, as well as being suitable for university researchers and industry software engineers. The first 50 participants accepted for the School received free places. This volume contains a record of some of the lectures and seminars delivered at the School.

The School was held at the time when Southwest University was celebrating its 110th anniversary. It was organized by the School of Computer and Information Science at Southwest University, providing lectures on leading-edge research in methods and tools for use in computer system engineering. The School aimed to enable participants to learn about state-of-the-art software engineering methods and technology advances from experts in the field.

An opening address was delivered by the Vice President of Southwest University, Prof. Yanqiang Cui, followed by an introduction to SETSS 2016 by Prof. Zhiming Liu. Sessions at the School were chaired by Professors Zili Zhang, Jonathan Bowen, Zhiming Liu, and Jim Woodcock.

The following lectures courses (four 90-minute lecture sessions each) were delivered during the School:

- Tao Xie: *Parameterized Unit Testing: Theory and Practice*
- Michael Butler: *Modelling and Verification in Event-B*
- Martin Leucker: *Runtime Verification*
- Yifeng Chen: *Parallel Programming Today*
- Jim Woodcock: *Semantics of Reactive Systems*
- Alvaro Miyazawa: *Java in the Safety-Critical Domain – A Refinement Approach*

In addition, there were two 120-minute evening seminars on related subject areas:

- Jonathan Bowen: *Alan Turing: Founder of Computer Science*
- Zhilin Wu: *Formal Reasoning about Infinite Data Values: An Ongoing Quest*

These additional presentations complemented the longer lecture courses.

Courses

Modelling and Verification in Event-B

Lecturer: Prof. Michael Butler, University of Southampton, UK

Biography: Michael Butler is a Professor of Computer Science at Southampton University. He is internationally recognised as a leading expert in refinement-based formal methods. He holds a PhD (Computation) from the University of Oxford. His research work encompasses applications, tools and methodology for formal methods, especially refinement based method such as B and Event-B. He has made key methodological contributions to the Event-B formal method, especially around model composition and decomposition. He plays a leading role in the development of several tools for B and Event-B, especially the Rodin toolset. Butler has a strong track record of collaboration with industry on the deployment of formal methods.

Overview: Formal modelling and verification lead to deeper understanding and higher consistency of specification and design than informal or semi-formal methods. A refinement approach means that models represent different abstraction levels of system design; consistency between abstraction levels is ensured by formal verification. These lectures provided an introduction to modelling and verification using Event-B providing guidance on appropriate use of set theory for domain modelling, use of refinement to represent systems at different abstraction levels and use of mathematical proof to verify the consistency between refinement levels.

Parallel Programming Today

Lecturer: Prof. Yifeng Chen, Peking University, China

Biography: Yifeng Chen is a research professor of the School of Electronics Engineering and Computer Science at Peking University and Vice Head, Department of Computer Science and Technology. He is a member of the Software Institute and the theory group. His main research interests include parallel programming model for multi-core and many-core architectures and parallel computing, (imperative, parallel, object-oriented and probabilistic) programming languages, and programming theory. His research activities include serving as a PC member for conferences such as lCTAC, UTP, lFM, IPDPS, SC, PPoPP, and CCGrid.

Overview: Today's parallel computer systems are diverse. This lecture presented several lower-level tools of parallel programming and how to lift the level of programming in algebraic structures. The parallel programming paradigms of this lecture included the CUDA parallel computing platform for programming the General-Purpose Graphics Processor Unit that powers the Tianhe-1A supercomputer, OpenMP Offload for programming Intel's MIC (Many Integrated Core)

Architecture that powers Tianhe 2 supercomputer, and MPI (Message Passing Interface) for programming a cluster of servers connected with a high-speed network.

Runtime Verification

Lecturer: Prof. Martin Leucker, University of Lübeck, Germany

Biography: Martin Leucker is Director of the Institute for Software Engineering and Programming Languages at the University of Lübeck, Germany. He obtained his Habilitation at TU München (awarded in 2007) while being a member of Manfred Broy's group on Software and Systems Engineering. At TU Munich, he also worked as a professor of Theoretical Computer Science and Software Reliability. Martin Leucker is the author of more than a hundred reviewed conference and journal papers in software engineering, formal methods, and theoretical computer science. He is frequently a PC member of top-ranked conferences and has been the principal investigator in several research projects with industry participation, especially in the medical devices, automotive, and energy domains.

Overview: This tutorial course gave an introduction to the field of runtime verification. More specifically, it presented a comprehensive and coherent assessment to Linear Temporal Logic-based monitor synthesis approaches. Both rewriting and automata-based techniques, each from a propositional as well as from a data perspective, were covered. Beyond a formal account, applications, especially in the area of testing, were presented. To this end, a practical introduction to the tool JUnitRV, which combines traditional unit testing for Java with Runtime Verification techniques, was included.

Java in the Safety-Critical Domain – A Refinement Approach

Lecturer: Dr Alvaro Miyazawa, University of York, UK

Biography: Alvaro Miyazawa is a research associate at High Integrity Systems Engineering Group in the University of York. His doctoral work formalised the semantics of Stateflow charts and defined a refinement strategy for the verification of sequential and parallel implementations. Since then, he has worked on the COMPASS project developing a comprehensive and integrated formal semantics for SysML with particular emphasis on state machine, block definition and internal block diagrams, and on the hiJaC project, extending the formal semantics of Safety Critical Java and refinement strategies for verification and generation of SCJ programs. He has been working on the RoboCalc project, developing a formal state machine notation tailored for the design and analysis of robotic applications.

Overview: Safety-Critical Java (SCJ) is a version of Java designed for programming real-time and safety-critical systems that require certification. A group at the University of York is working with members of Open Group committee that is defining a standard for SCJ to define techniques for verification of programs. This course presented SCJ, the challenges involved in verifying SCJ programs, and the approach used for this. New modelling languages and techniques for automatic generation and verification of models was also covered.

Semantics of Reactive Systems

Lecturer: Prof. Jim Woodcock, University of York, UK

Biography: Jim Woodcock is Professor of Software Engineering and Head of the Department of Computer Science at the University of York in England. His main research interests are in the industrial applications of software engineering, formal verification, programming language semantics, and cyber-physical systems. The research team he previous led at Oxford University won the Queen's Award for Technological Achievement for its work on the formal development of smart cards. He is a Fellow of the UK Royal Academy of Engineering.

Overview: Unifying Theories of Programming (UTP) provides a foundation for compositional semantics for a variety of different language paradigms. This course showed how to give semantics to imperative programs, pointer-rich programs, reactive programs with concurrency and communication, and reactive programs with mobile channels. It also demonstrated how these different paradigms can be composed to create a powerful programming language with stateful, reactive, reconfigurable processes.

Parameterized Unit Testing: Theory and Practice

Lecturer: Prof. Tao Xie, University of Illinois at Urbana-Champaign, USA

Biography: Tao Xie is an Associate Professor and Willett Faculty Scholar in the Department of Computer Science at the University of Illinois at Urbana-Champaign, USA. His research interests are in software engineering and software security, with focus on software testing, software analytics, and educational software engineering. He was an ACM Distinguished Speaker and is an IEEE Computer Society Distinguished Visitor. He received an NSF CAREER Award in 2009. He received a 2014 Google Faculty Research Award, a 2011 Microsoft Research Software Engineering Innovation Foundation (SEIF) Award, 2008, 2009, and 2010 IBM Faculty Awards, and a 2008 IBM Jazz Innovation Award. He was the Program Chair of ISSTA 2015.

Overview: This course presented the latest research and practice on principles, techniques, and applications of parameterized unit testing in practice, highlighting success stories, research and education achievements, and future research directions in developer testing. The course will help improve developer skills and knowledge for writing PUTs and give overview of tool automation in supporting PUTs. Attendees will acquire the skills and knowledge needed to perform research or conduct practice in the field of developer testing and to integrate developer testing techniques in their own research, practice, and education.

Seminars

Alan Turing: Founder of Computer Science

Lecturer: Prof. Jonathan Bowen, London South Bank University, UK.

Biography Jonathan Bowen, FBCS FRSA, is Chairman of Museophile Limited (founded in 2002) and an Emeritus Professor at London South Bank University, where he established and headed the Centre for Applied Formal Methods in 2000. During 2013–15, he was Professor of Computer Science at Birmingham City University. His interests have ranged from formal methods, safety-critical systems, the Z notation, provably correct systems, rapid prototyping using logic programming, decompilation, hardware compilation, software/hardware co-design, linking semantics, and software testing, to the history of computing, museum informatics, and virtual communities.

Overview: Alan Turing (1912–1954) has been increasingly recognised as an important mathematician and philosopher who despite his short life developed ideas that have led to foundational aspects of computer science and related fields. This seminar talk provided an overview of the diverse aspects related to Turing's remarkable achievements, with respect to the production of a book, *The Turing Guide*, a collected volume of 42 chapters, published in Oxford University Press in 2017. Although the story of Turing can be seen as one of tragedy, with his life cut short while still at the height of his intellectual powers, just short of his 42nd birthday, from a historical viewpoint Turing's contribution to humankind has been triumphant.

Formal Reasoning about Infinite Data Values: An Ongoing Quest

Lecturer: Dr Zhilin Wu, Institute of Software, Chinese Academy of Sciences, China.

Biography: Zhilin Wu is an associate research professor in State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. His main research interests include program analysis and verification, computational logic, automata theory, and database theory.

Overview: Infinite data values are pervasive in computer systems, e.g., process identifiers, file names, integer or floating variables in programs, data parameters in network messages, records in databases, etc. Nevertheless, reasoning about them formally is notoriously difficult, since the infinity of data domains easily induces the undecidability of the reasoning tasks. The usual practice in most of the current approaches or tools is to ignore or abstract away the data infinity. A long-term goal is to show that in many scenarios, proper formalisms can be found, so that on the one hand the infinite data values, instead of being abstracted away, can be handled directly and explicitly, and on the other hand, the reasoning process can still be largely automated and made efficient. In this seminar, a summary of efforts towards this goal over the previous five years was given.

From the lectures and seminars, a record of the School has been distilled in six chapters within this volume as follows:

- Jonathan Bowen: *Alan Turing: Founder of Computer Science*
- Jim Woodcock and Simon Foster: *UTP by Example: Designs*
- Michael Butler: *Reasoned Modelling with Event-B*
- Ana Cavalcanti, Alvaro Miyazawa, Andy Wellings, Jim Woodcock and Shuai Zhao: *Java in the Safety-Critical Domain*
- Martin Leucker: *Runtime Verification for Linear-time Temporal Logic*
- Taolue Chen, Fu Song and Zhilin Wu: *Formal Reasoning On Infinite Data Values: An Ongoing Quest*

Acknowledgments: We would like to thank the lecturers and their coauthors for their professional commitment and effort, the strong support of Southwest University, and the enthusiastic work of the local organization team, without which SETSS 2016 would not have been possible. Thank you to Xin Chen (Nanjing University) for help with assembling the Proceedings. We are grateful for the support of Alfred Hofmann and Anna Kramer of Springer Lecture Notes in Computer Science in the publication of this volume.

February 2017

Zhiming Liu
Jonathan P. Bowen
Co-chairs, SETSS 2016

Attendees, organizers, and lecturers at SETSS 2016. Front row, left to right: Tao Xie, Michael Butler, Jonathan Bowen, Yanqiang Cui, Zili Zhang, ?, Zhiming Liu.

Table of Contents

Alan Turing: Founder of Computer Science

Jonathan P. Bowen

London South Bank University, London SE1 0AA, UK,
jonathan.bowen@lsbu.ac.uk,
WWW home page: http://www.jpbowen.com

UTP by Example: Designs

Jim Woodcock and Simon Foster

University of York, UK

Reasoned Modelling with Event-B

Michael Butler

University of Southampton, UK

Java in the Safety-Critical Domain

Ana Cavalcanti, Alvaro Miyazawa, Andy Wellings, Jim Woodcock, and Shuai Zhao

University of York, UK

Runtime Verification for Linear-time Temporal Logic

Martin Leucker

University of Lübeck, Germany

Formal Reasoning On Infinite Data Values: An Ongoing Quest

Taolue Chen[1], Fu Song[2], and Zhilin Wu[3]

[1] Department of Computer Science, Middlesex University London, UK
[2] School of Information Science and Technology, ShanghaiTech University, Shanghai, China
[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

Author Index