Software Qual J (2016) 24:3–5
DOI 10.1007/s11219-016-9307-9

Special issue on High Assurance Systems Engineering

Peter J. Clarke[1] · **Mohammad Zulkernine**[2]

Published online: 22 January 2016

As software-based systems become more pervasive, there is a need to construct systems that in addition to meeting their functional requirements also adhere to their non-functional requirements, specifically those requirements that impact reliability, safety and security. The stated requirements all make a strong case for High Assurance Systems Engineering. Both academia and industry continue to seek new techniques and tools to develop systems that ensure their systems meet the highest assurance standards. In this special issue on High Assurance Systems Engineering, seven papers are presented that investigate problems in the areas of reliability, safety and security.

The first paper "An Empirical Exploration of Distortion in Fault Injection Experiments" by Erik van der Kouwe, Cristiano Giuffrida and Andrew S. Tanenbaum investigates whether fault injection experiments faithfully represents the fault model designed by the user. Most software researchers and practitioners accept the fact that there will always be bugs present in software, except for trivial programs; therefore the focus for highly reliable systems should also be on the mechanisms that contain faults and recover from them. This paper makes a case for carefully evaluating whether activated faults, when using a fault injection technique, match the user's fault model of the software and identifying which parameters impact the activation of faults from the fault model.

The second paper "Accelerating Temporal Verification of Simulink Diagrams Using Satisfiability Modulo Theories" by Petr Bauch, Vojtech Havel and Jiri Barnat explores the process of model checking for Simulink diagrams by developing a hybrid approach that involves the explicit and symbolic representations of sets. The approach takes into account

✉ Peter J. Clarke
clarkep@cis.fiu.edu

Mohammad Zulkernine
mzulker@cs.queensu.ca

[1] School of Computing and Information Sciences, Florida International University, Miami, FL, USA

[2] School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada

the temporal aspects of verification and set-based representation of data, thereby handling both sources of non-determinism present in verification. The hybrid approach attempts to tackle the state explosion problem by using set-based reduction with explicit sets as one representation and bit-vector formulae as another representation. More specifically, the hybrid representation entails inferring knowledge about the system under verification, which is not included by the developers, thereby accelerating the verification process.

The third paper "Towards High Assurance Software Systems with Adaptive Fault Management" by Koichiro Rinsaka and Tadashi Dohi describes an adaptive approach to estimate the optimal preventive rejuvenation schedule which maximizes the steady-state system availability. Software rejuvenation is a preventive and proactive solution that is useful for counteracting software aging, which may affect the performance of applications and eventually cause them to fail. Traditional models of preventive rejuvenation are based on fixed samples of system failure which are validated on past observations. This paper presents a statistically nonparametric algorithm to estimate the optimal preventive rejuvenation schedule, which uses upper and lower bounds of the system with a one-stage look-ahead survival function obtained from system failure, time data and derived pessimistic and optimistic rejuvenation policies.

The fourth paper "Experiences with Software-based Soft-Error Mitigation Using AN-Codes" by Martin Hoffmann, Peter Ulbrich, Christian Dietrich, Horst Schirmeier, Daniel Lohmann and Wolfgang Schröder-Preikschat describes the experiences and lessons learned from implementing AN codes in the CoRed (Combined Redundancy) dependable voter. Soft-error mitigation continues to be one of the major challenges for safety–critical applications and systems. To ameliorate transient hardware faults in software the CoRed voter, a software-based fault-tolerance approach for mixed-criticality applications, is used. However, the transition from mathematical theory to machine code results in some practical misconceptions and architecture-dependent implementation glitches. These misconceptions include binary number representation and ranges, silent assumptions about the compiler and specific characteristics of the hardware platform. This paper presents a practitioner's guide on how to deal with these challenges by illuminating typical problem areas and presenting feasible solutions.

The fifth paper "Certifying a Java Type Resolution Function Using Program Transformation, Annotation, and Reflection" by Victor Winter, Carl Reinke and Jonathan Guerrero presents a novel approach for certifying the correctness of a given type resolution function with respect to an arbitrary Java source code base. Type resolution is fundamental to static analysis which is key to accurately compiling source code in typed languages, as well as the analysis needed for many white-box testing techniques. The approach in the paper uses a test suite to comprehensively exercise a full range of resolution possibilities within Java for those corner cases that are non-trivial. A set of program transformations are used to instrument Java source code which can then be used by reflection to certify the correctness of the type resolution function.

The sixth paper "Normalizing Feature Vector Structure in Keystroke-dynamics Authentication Systems" by Zahid Syed, Sean Banerjee and Bojan Cukic tests the efficacy of a new authentication method in distinguishing users based on keystroke dynamics. Current keystroke dynamics approaches typically assume that in order to measure similarity between typing patterns of individuals, a text string is entered using the same sequence of keys. This paper uses a different approach that involves variant typing sequences and shows that variations in keystroke sequences can provide valuable discriminatory information and improve the performance of keystroke dynamics authentication systems.

The final paper "Assessing Vulnerability Exploitability Risk Using Software Properties" by Awad Younis, Yashwant Malaiya and Indrajit Ray proposes a metric that evaluates vulnerability exploitability, based on the presence of a function call connecting attack surface entry points to the vulnerability location within the software under consideration. Common Vulnerability Scoring System (CVSS) metrics have become the de facto standard for assessing the severity of security vulnerabilities. However, CVSS metrics have several limitations including the assignment of static subjective numbers to the metrics based on expert knowledge regardless of the type of vulnerability and the fact that two of its factors have the same value for almost all vulnerabilities. The proposed approach in the paper was evaluated using twenty reported vulnerabilities of the Apache HTTP server at the source code level. The results of the evaluation show promise in objectively measuring the risk of vulnerability exploitability in software systems.

The guest co-editors would like to thank the authors for their hard work in preparing the manuscripts and updating the manuscripts based on the feedback provided by the reviewers. We would also like to thank the reviewers for their time in preparing the thoughtful reviews and in some cases performing multiple reviews of the papers prior to final acceptance. Finally, we would like to thank the Editor-in-Chief, Professor Rachel Harrison, and the editorial staff for their patience in working with us; we are very appreciative of their hard work in getting this special issue on High Assurance Systems Engineering ready for publication. We hope that you enjoy this special issue.