MSeer – an Advanced Technique for Locating Multiple Bugs in Parallel

W. Eric Wong
Department of Computer Science
University of Texas at Dallas
http://www.utdallas.edu/~ewong

Abstract

In practice, a program may contain multiple bugs. The simultaneous presence of these bugs may deteriorate the effectiveness of existing fault-localization techniques to locate program bugs. While it is acceptable to use all failed and successful tests to identify suspicious code for programs with exactly one bug, it is not appropriate to use the same approach for programs with multiple bugs because the *due-to* relationship between failed tests and underlying bugs cannot be easily identified. One solution is to generate fault-focused clusters by grouping failed tests caused by the same bug into the same clusters. We propose *MSeer* – an advanced fault localization technique for locating multiple bugs in parallel. Our major contributions include the use of (1) a revised Kendall tau distance to measure the distance between two failed tests, (2) an innovative approach to simultaneously estimate the number of clusters and assign initial medoids to these clusters, and (3) an improved K-medoids clustering algorithm to better identify the due-to relationship between failed tests and their corresponding bugs. Case studies on 840 multiple-bug versions of seven programs suggest that MSeer performs better in terms of effectiveness and efficiency than two other techniques for locating multiple bugs in parallel.

This paper has been accepted for publication by the IEEE Transactions on Software Engineering (TSE; https://www.computer.org/csdl/trans/ts/preprint/08119545.pdf) and also presented at the 2018 IEEE International Conference on Software Engineering (ICSE) as a Journal First paper.

Bio

W. Eric Wong received his M.S. and Ph.D. in Computer Science from Purdue University, West Lafayette, Indiana, USA. He is a Full Professor and the Founding Director of Advanced Research Center for Software Testing and Quality Assurance (http://paris.utdallas.edu/stqa) in Computer Science at the University of Texas at Dallas (UTD). He also has an appointment as a guest researcher at the National Institute of Standards and Technology, an agency of the U.S. Department of Commerce. Prior to joining UTD, he was with Telcordia Technologies (formerly Bellcore) as a senior research scientist and the project manager in charge of Dependable Telecom Software Development.

Professor Wong was the recipient of the 2014 IEEE Reliability Society Engineer of the Year. He is also the Editor-in-Chief of the IEEE Transactions on Reliability. His research focuses on helping practitioners improve software quality while reducing production cost. In particular, he is working on software testing, program debugging, risk analysis, safety, and reliability.

Professor Wong is the Founding Steering Committee Chair of the IEEE International Conference on Software Security and Reliability (SERE) and the IEEE International Workshop on Program Debugging. In 2015, the SERE conference and the QSIC conference (International Conference on Quality Software) merged into one large conference, QRS, with Q representing *Quality*, R for *Reliability*, and S for *Security*. Professor Wong continues to be the Steering Committee Chair of this new conference (http://paris.utdallas.edu/qrs18).

Bayesian Methods for Reliability Data Analysis

Steven Li
Department of Industrial Engineering and Engineering Management
Western New England University
https://www1.wne.edu/engineering/faculty.cfm?uid=1031

Abstract

Bayesian inference has been increasingly accepted and used by industry practitioners for reliability analysis due to advanced computational capabilities as well as its advantage of dealing with limited testing data. The motivation of Bayesian inference and its pros and cons versus the frequentist statistical inference are first discussed and reviewed. Then, the talk provides a basic and comprehensive introduction of Bayesian inference methodologies covering topics such as prior elicitation and prior distribution, sampling models, posterior inference and simulation methods. The methodological introduction of Bayesian inference is accompanied with reliability data analysis applications and case studies using various types of reliability data, e.g., binary fail/pass data, lifetime data, and degradation data. In addition, numerical examples are also demonstrated through the commonly used statistical data analysis software of R. In summary, this tutorial presents to the audience a comprehensive understanding to the basic Bayesian statistical methods, applications and examples of Bayesian reliability data analysis, and the software implementation of Bayesian reliability estimation using open source R software.

Bio

Dr. Zhaojun (Steven) Li is Professor with the Department of Industrial Engineering and Engineering Management at Western New England University in Springfield, Massachusetts, USA. Dr. Li’s research interests focus on reliability, quality and safety engineering in product design, systems engineering and its applications in new product development, diagnostics and prognostics of complex engineered systems, and engineering management. He earned his PhD degree in Industrial Engineering from the University of Washington. He is an ASQ Certified Reliability Engineer and a Caterpillar Six Sigma Black Belt. Dr. Li’s most recent industry position was as a reliability team lead with the Caterpillar Rail Division to support the company’s Tier 4 New Four Stroke Engine and Gas-Diesel Dual Fuel Engine Development. He is currently serving as an Associate Editor for the Journal of IEEE Transactions on Reliability and is Co-Editor-in-Chief of the International Journal of Performability Engineering. He is a member of IISE, INFORMS and IEEE. He is also one of the recipients of the 1000 Young Talents Plan
(青年千人).

Towards a unified executable formal automobile OS kernel and its applications

Min Zhang
Department of Software Science and Technology
East China Normal University

Abstract

In automobile industry, it is a common approach to develop automobile real-time operating systems under some standards. For instance, OSEK/VDX is a world-widely adopted open standard. Traditional workflow is to first understand the standard, design and develop a system, then test its conformance to the standard, and finally deploy. There are several issues with the traditional workflow, e.g., ambiguities in standards may lead to incorrect design and implementation of real-world systems; the conformance of real-world systems to standards is difficult to check; bug-fixing after implementation is costly. To remedy the situation, in this paper we present a unified executable formal automobile kernel under OSEK/VDX standard by defining the operational semantics of the system services in the standard using a rewrite-based executable semantic framework called K. The formal kernel is unified in that it serves for multiple purposes such as: (1) formal modeling of the OSEK/VDX standard helps detect ambiguities in the standard; (2) the executable kernel is essentially a formal model of the standard, which can be used to verify the correctness of automobile applications; (3) verified applications can be used as test cases to check the conformance of a real-world automobile operating system against the OSEK/VDX standard. Using the formal kernel we identify several ambiguities in the OSEK/VDX standard and a potential deadlock vulnerability in an industrial automobile application.

Bio

Min Zhang is a currently an Associate Professor in the Department of Software Science and Technology, East China Normal University. His current research is focused on formal methods, particularly software modeling and verification. Min Zhang obtained his Ph.D in Japan Advanced Institute of Science and Technology in 2011, and afterward worked as a post-doc before joining ECNU.

Evaluating Static Analysis

Shiyi Wei
Department of Computer Science
University of Texas at Dallas
https://www.utdallas.edu/~swei/

Abstract

Building practical static analysis tools is a challenging task. Despite the benefits of using static analysis tools for automating code inspections for software engineers, research suggests that performance, imprecision, and unsoundness are barriers that make these tools underused in practice. On top of engineering robust tools, it requires understanding the semantics of real-world programs and the tradeoffs of design choices.

In this talk, we share our experience building practical numeric analysis tools to detect security vulnerabilities in real-world Java programs. Designing a scalable numeric static analysis presents with a multitude of design choices (e.g., numeric domain, heap abstraction, and context sensitivity), each of which may interact with others. We developed a family of abstract interpretation-based numeric static analyses for Java and systematically evaluated the impact of 162 analysis configurations on a large benchmark suite using statistical analysis. Our experiment considered the precision and performance tradeoffs of the analyses for discharging array bounds checks. Finally, we report an application of the numerical analysis to detect timing-channel attacks in Java programs.

Bio

Shiyi Wei is currently an Assistant Professor at the University of Texas at Dallas. He obtained his Ph.D. in Computer Science from Virginia Tech in 2015, and then completed a postdoc at the University of Maryland, Collage Park before coming to UTD. His research interests span the areas of programming languages, software engineering, and security. The goal of his research is make program analysis practical for improving the security and reliability of real-world software. He has published articles at top venues in his areas of interest, such as FSE, ISSTA, ESOP, and ECOOP.

Generating Specification from Timing Requirements using Patterns in Cyber-Physical Systems

Xiaohong Chen
School of Computer Science and Software Engineering
East China Normal Univeristy
https://faculty.ecnu.edu.cn/s/2150/main.jspy

Abstract

To satisfy timing requirements, the timing specifications for cyber-physical systems (CPSs) have to evolve according to the changes in the physical world.
However, it is challenging to maintain the overall satisfaction in the timing requirement problems due to potential inconsistencies between the representation of the timing requirements/specifications and the representation of the physical world. The former is typically specified in formal languages such as Clock Constraint Specification Language (CCSL), whilst the latter is typically expressed informally, e.g., in Problem Frames (PF). Therefore, in this paper, we propose TimePF-E, an automated approach to generating timing specifications while preserving their satisfaction to the timing requirements. The approach uses a new language, PF-CCSL, which unifies Problem Frames notations and CCSL constraints to specify the timing requirements problem as well as physical world changes. In additional to the syntax of representations, the operational semantics of PF-CCSL enable the transformation from PF-CCSL into the representation of behavioral models, e.g., NuSMV models, which in turn could be checked to verify the satisfaction of timing requirements in the changed models. Automated transformation patterns from timing requirements to timing specification are presented from the viewpoints of PF for requirements engineers to apply this approach. We present a case study in timing-critical domains to demonstrate the application and utility of this formalism.

Bio

Xiaohong Chen is currently an associate professor in the School of Computer Science and Software Engineering at the East China Normal University. Her current research focuses on Requirements Engineering, Knowledge Engineering, Cyber-Physical Systems, Safety-Critical Systems, and Real-Time Systems. She earned her PhD degree from Academy of Mathematics and Systems Science, Chinese Academy of Sciences. She has authored over 40 papers in software engineering or requirements engineering or knowledge engineering conferences and journals.

Using Bugs to Debug

Lingming Zhang
Department of Computer Science
University of Texas at Dallas
https://www.utdallas.edu/~lxz144130/

Abstract

Localizing failure-inducing code is essential for software debugging. Manual fault localization can be quite tedious, error-prone, and time-consuming. Therefore, a huge body of research efforts has been dedicated to automated fault localization. Spectrum-based fault localization, the most intensively studied fault localization approach based on test execution information, may have limited effectiveness, since a code element executed by a failed test may not necessarily have impacts on the test outcome and can cause the test to fail. To bridge the gap, mutation-based fault localization has been proposed to inject artificial bugs to check the impact of each code element for better fault localization. However, there are limited studies on the effectiveness of mutation-based fault localization on a sufficient number of real bugs. In this paper, we perform an extensive study to compare mutation-based fault localization techniques with various state-of-the-art spectrum-based fault localization techniques on 357 real bugs from the Defects4J benchmark suite. The study results firstly demonstrate the effectiveness of mutation-based fault localization, as well as reveals a number of guidelines to further improve mutation-based fault localization. Based on the learnt guidelines, we further transform test outputs/messages and test code to obtain various mutation information. Then, we propose TraPT, an automated Learning-to-Rank technique to fully explore the obtained mutation information for effective fault localization. The experimental results show that TraPT localizes 65.12% and 94.52% more bugs within Top-1 compared to state-of-the-art mutation and spectrum based techniques when using the default setting of LIBSVM.

Bio

Dr. Lingming Zhang is an Assistant Professor in the Computer Science Department at the University of Texas at Dallas. He obtained his Ph.D. degree from the Department of Electrical and Computer Engineering in the University of Texas at Austin in May 2014. He received his MS degree and BS degree in Computer Science from Peking University (2010) and Nanjing University (2007), respectively. His research interests lie broadly in software engineering and programming languages, including automated software analysis, testing, debugging, and verification, as well as software evolution and mobile computing. He has authored over 40 papers in premier software engineering or programming language conferences and transactions. He has also served on the program/organization committee or artifact evaluation committee for various international conferences (including ICSE, ISSTA, ASE, ICST, ICSM, and OOPSLA). His research is being supported by the NSF, Google, Huawei, and Samsung.

Towards Automated Requirement Modeling and Analysis: Practice in the Domains of Aviation and Aerospace Embedded Control Software

Weikai Miao
School of Computer Science and Software Engineering
East China Normal University

Abstract

Effective and efficient requirements modeling and analysis significantly contribute to the software quality. However, the industrial practitioner is still suffering from the lack of automated engineering methodologies that can effectively encompass precise requirements modeling and rigorous requirements analysis. Although both the research and industrial communities have reorganized that formal method is a promising solution, and some industrial standards requires the application of formal method (e.g., DO-333 standard for aviation software), to utilize it for requirements modeling and V&V (validation and verification) in large-scale industrial projects is still a challenge. To tackle this problem, in this paper, we present a formal engineering approach to the formal requirements modeling and V&V in the domains of aviation and aerospace embedded control software, which is developed based on our knowledge, researches and practices in these domain over five years. The advancements of the approach include: 1) a domain-specific light-weight formal notation and corresponding template for formal requirements modeling; 2) a systematic engineering approach for guiding the formal specification construction, V&V activities and MCDC coverage test data generation, which conforms to the major industrial standards (e.g., DO-333 for aeronautics software), and 3) a tool supporting the approach in real industrial projects. We have applied the approach and the tool in several software projects in the domains of aviation and aerospace. The experiments results shows that the approach can significantly facilitate the formal modeling, V&V and model-based testing in industry. We have also reported the experiences and lessons accumulated from our applications of the approach and the tool.

Bio

Weikai Miao is an associate professor in the School of Computer Science and Software Engineering at the East China Normal University. He received the Ph.D in Computer Science from Hosei University, Japan in 2013. His research interests include Formal Methods and Formal Engineering Methods for Software Development, Specification Verification and Validation, Requirements Engineering and Software Testing. He has published over 20 academic papers in refereed journals and international conferences including the ICSE and IEEE Trans. On Services Computing. He has also worked on several large-scale industrial software projects in various domains since 2013.

Predictable GPGPU Computing in Autonomous Driving Systems

Cong Liu
Department of Computer Science
University of Texas at Dallas
http://www.utdallas.edu/~cong/

Abstract

Graphic processing units (GPUs) have seen wide-spread use in several computing domains as they have the power to enable orders of magnitude faster and more energy-efficient execution of many applications. Unfortunately, it is not straightforward to reliably adopt GPUs in many safety-critical cyber-physical systems that require predictable timing correctness, one of the most important tenets in certification required for such systems. A key example is the advanced automotive system where timeliness of computations is an essential requirement of correctness due to the interaction with the physical world. In this talk, I will describe a systems solution for ensuring predictable timing correctness through enhancing the execution concurrency and resource utilization in a DNN-driven autonomous driving system.

Bio

Cong Liu is currently a tenure-track assistant professor in the Department of Computer Science at the University of Texas at Dallas. His current research focuses on Real-Time Systems, GPGPU Computing, and Data-driven Design of Cyber-Physical Systems. He received several best paper awards at premier conferences including RTSS, RTAS, and INFOCOM. He is a recipient of the prestigious NSF CAREER award.

SmartUnit: Empirical Evaluations for Automated Unit Testing of Embedded Software in Industry

Chengyu Zhang
School of Computer Science and Software Engineering
East China Normal University
http://www.chengyuzhang.com

Abstract

In this paper, we aim at the automated unit coverage-based testing for embedded software. To achieve the goal, by analyzing the industrial requirements and our previous work on automated unit testing tool CAUT, we rebuild a new tool, SmartUnit, to solve the engineering requirements that take place in our partner companies. SmartUnit is a dynamic symbolic execution implementation, which supports statement, branch, boundary value and MC/DC coverage.

SmartUnit has been used to test more than one million lines of code in real projects. For confidentiality motives, we select three in-house real projects for the empirical evaluations. We also carry out our evaluations on two open source database projects, SQLite and PostgreSQL, to test the scalability of our tool since the scale of the embedded software project is mostly not large, 5K-50K lines of code on average. From our experimental results, in general, more than 90% of functions in commercial embedded software achieves 100% statement, branch and MC/DC coverage, more than 80% of functions in SQLite and more than 60% of functions in PostgreSQL achieve 100% statement and branch coverage. Moreover, SmartUnit is able to find the runtime exceptions at the unit testing level. We also have reported exceptions like array index out of bounds and divided-by-zero in SQLite. Furthermore, we analyze the reasons for low coverage in automated unit testing in our setting and give a survey on the situation of manual unit testing with respect to automated unit testing in industry.

This paper has been presented at the 2018 IEEE International Conference on Software Engineering (ICSE) in Software Engineering in Practice (SEIP) Track.

Bio

Chengyu Zhang is now a second-year Ph.D. student in the School of Computer Science and Software Engineering at the East China Normal University (ECNU), China, and supervised by Prof. Geguang Pu. He graduated from East China Normal University with Excellent Graduate Award in 2016. His research interests are in automated software testing, symbolic execution, and program analysis.