

XIAO (LESTER) YU

Department of Computer Science, North Carolina State University ◊ Raleigh, NC 27695

(919) · 518 · 4431 ◊ xyu10@ncsu.edu

EDUCATION

North Carolina State University

Ph.D. in Computer Science

2011 - 2017 (*Expected*)

Advisor: Dr. Guoliang Jin

Visiting Student at University of Illinois at Urbana-Champaign

2013 - 2014

Host: Dr. Tao Xie

East China Normal University, Shanghai, China

M.S. in Computer Software & Theory

2008 - 2011

Advisor: Dr. Geguang Pu

B.S. in Software Engineering

2004 - 2008

RESEARCH INTERESTS

- Software Reliability and Performance
- Programming Languages
- Systems
- Automated Software Engineering

PUBLICATIONS

1. Xiao Yu, Pallavi Joshi, Jianwu Xu, Guoliang Jin, Hui Zhang, and Guofei Jiang. CloudSeer: Workflow Monitoring of Cloud Infrastructures via Interleaved Logs. In Proceedings of the 21st International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pages 489-502, 2016.
2. Mengqi Gu, Xiao Yu, Wei Yang, Tao Xie, Xintao Wu. Characterizing Performance Bugs of Application-Database Interactions in Open-Source Software. Technical Report, 2015.
3. Xiao Yu, Shi Han, Dongmei Zhang, and Tao Xie. Comprehending Performance from Real-World Execution Traces: A Device-Driver Case. In Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pages 193-206, 2014.
4. Wei Yang, Xiao Yu. AT-EASE: A Tool for Early and Quick Usability Evaluation of Smartphone Applications. Technical Report, 2013.
5. Xiao Yu, Shuai Sun, Geguang Pu, Siyuan Jiang, and Zheng Wang. A Parallel Approach to Concolic Testing with Low-cost Synchronization. Electronic Notes in Theoretical Computer Science (ENTCS), Volume 274 (August 2011), pages 83-96, 2011.
6. Kang Miao, Xiao Yu, Siyuan Jiang, and Ji Zhao, Run-time Discovery of Java Design Patterns. In Proceedings of the 2011 Second International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), pages 3329-3332, 2011.
7. Kang Miao, Xiao Yu, Ji Zhao, and Yu Shen. Java Design Pattern Recognition Based on Relational Calculus. Application Research of Computers (in Chinese), Volume 27, Issue 9, pages 3425-3430, 2010.
8. Libo Feng, Xiao Yu, Geguang Pu, Huibiao Zhu, and Bin Gu. Property Checking for Design Patterns. In Proceedings of the IASTED International Conference on Software Engineering, pages 87-94, 2010.

9. Tao Sun, Zheng Wang, Geguang Pu, Xiao Yu, Zongyan Qiu, and Bin Gu. Towards Scalable Compositional Test Generation. In Proceedings of the 2009 Ninth International Conference on Quality Software (QISC), pages 353-358, 2009.
10. Zheng Wang, Xiao Yu, Tao Sun, Geguang Pu, Zuohua Ding, and JueLiang Hu. Test Data Generation for Derived Types in C Program. In Proceedings of the 2009 Third IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE), pages 155- 162, 2009.
11. Zheng Wang, Xiao Yu, Geguang Pu, Libo Feng, Huibiao Zhu, and Jifeng He. Execution Semantics for rCOS. In Proceedings of the 2008 15th Asia-Pacific Software Engineering Conference (APSEC), pages 119-126, 2008.
12. Xiao Yu, Zheng Wang, Geguang Pu, Dingding Mao, and Jing Liu. The Verification of rCOS Using Spin. Electronic Notes in Theoretical Computer Science (ENTCS), Volume 207 (April 2008), pages 49-67, 2008.

RESEARCH EXPERIENCES

Workflow Monitoring of Cloud Infrastructures via Logs 2014 - 2015 *ASPLOS 2016* *In collaboration with NEC Laboratories America*

- In order to help cloud administrators monitor complex task executions involving distributed service components, proposed and developed a lightweight workflow monitoring approach based on logs readily available in existing cloud infrastructures.
- The proposed approach can work on interleaved log messages by attributing each log message to a possible log sequence, and incrementally check whether each log sequence conforms with predefined workflow models built from correct task executions.
- Experimentally applied the proposed approach to OpenStack for monitoring task failures and performance degradation.

Empirical Study of Performance Bugs in Database Applications 2013 - 2015 *Submission Under Review*

- Being motivated by the wide adoption of databases in contemporary applications, investigated how application-database interactions can cause adverse performance impacts by studying over 180 performance bugs in popular open-source projects.
- Designed taxonomies to categorize database-related performance bugs and their fix strategies, providing guidelines to automated tools for detecting and fixing performance bugs related to application-database interactions.

Performance Analysis of Device Drivers on Real-World Execution Traces 2012 - 2013 *ASPLOS 2014* *In collaboration with Microsoft Research (Asia)*

- From large-scale real-world execution traces, identified that the system performance can be compromised by *cost propagation*, which is an adverse effect caused by contentions on multiple locks combining with hierarchical dependencies among device drivers and the Windows kernel.
- Proposed and developed a practical two-step approach with effective data and pattern abstractions to measure performance impacts manifested through cost propagation, and discover behavioral patterns closely related to performance problems.

Dynamic Test Generation for C Language 2007 - 2011 *TASE 2009, QISC 2009, ENTCS 2011*

- With improving the efficiency and the usability of symbolic execution, implemented the automated testing tool CAUT¹ that aims at applying the technique to the industrial software testing.

¹CAUT(C Analysis and Unit Testing): <http://lab205.org/caut>

- In order to facilitate the application of automated testing in industrial software, introduced a novel program model and several particular algorithms as the supplement of the numerical constraint solver to improve the approach of the test data generation for complex data structures and pointers.
- In order to relieve the path explosion problem when testing compositional units, adopted side effect analysis to eliminate the records of variables that are irrelevant to path conditions and partial execution which can reduce the number of path exploring iterations.
- Designed parallel algorithms for dynamic test generation based on the partial order relations in the sequential exploration of program paths.

Validation of Design Patterns in Object-Oriented Programs

2009 - 2010

IASTED 2010, AIMSEC 2011

- With selecting object-oriented concepts to describe structures and behaviors of programs with design patterns, designed a relational calculus, and abstracted a specific object model in order to support the validation algorithms.
- Established a methodology and corresponding tool to detect and validate the usage of design patterns in Java program using the relational calculus and the abstract object model.

Analysis and Verification of rCOS²

2007 - 2008

ENTCS 2008, APSEC 2008

- Extended the original version of rCOS to support assertions, invariants and parallelism, and proposed an approach to transform the extended rCOS model to SPIN model in order to provide a straight way for model verification.
- Constructed the operational semantic model for rCOS including parallelism, which clarified the real effects of rCOS programs, and provided a secure guide for the implementation of rCOS.

TEACHING EXPERIENCES

Teaching Assistants

- CSC 568: Enterprise Storage Architecture *Spring 2015*
Lecturer: Dr. Vincent W. Freeh
- CSC 506: Architecture of Parallel Computers *Spring 2012, Spring 2015*
Lecturer: Dr. Edward F. Gehringer
- CSC 510: Software Engineering *Fall 2011*
Lecturer: Dr. Annie I. Antón

PROFESSIONAL ACTIVITIES

Subreviewers	ASE 2014; ICSM 2012; ICST 2014; ISSTA 2012, 2013; MSR 2014; OOPSLA 2013
Student Volunteer	FSE 2012
Conference Presentations	ASPLOS 2014; ASPLOS 2016

²rCOS: <http://rcos.iist.unu.edu/>