Lecture Notes in Computer Science 12153

More information about this series at http://www.springer.com/series/7408

Jun Pang · Lijun Zhang (Eds.)

Dependable Software Engineering

Theories, Tools, and Applications

6th International Symposium, SETTA 2020
Guangzhou, China, November 24–27, 2020
Proceedings

Springer

Editors
Jun Pang
University of Luxembourg
Esch-sur-Alzette, Luxembourg

Lijun Zhang
Chinese Academy of Sciences
Beijing, China

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-030-62821-5 ISBN 978-3-030-62822-2 (eBook)
https://doi.org/10.1007/978-3-030-62822-2

LNCS Sublibrary: SL2 – Programming and Software Engineering

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains the research papers presented at the 6th Symposium on Dependable Software Engineering: Theories, Tools, and Applications (SETTA 2020) series of conferences – held during November 24–27, 2020, in Guangzhou, China. The purpose of SETTA is to bring international researchers together to exchange research results and ideas on bridging the gap between formal methods and software engineering. The interaction with the Chinese computer science and software engineering community is a central focus point. The aim is to show research interests and results from different groups so as to initiate interest-driven research collaboration. Past SETTA symposiums were successfully held in Nanjing (2015), Beijing (2016), Changsha (2017), Beijing (2018), and Shanghai (2019).

SETTA 2020 attracted 20 submissions co-authored by researchers from 10 countries. Each submission was reviewed by five Program Committee members with help from additional reviewers. The Program Committee discussed the submissions online and decided to accept 10 regular papers and 1 short paper for presentation at the conference. The program also included four invited talks given by Prof. Holger Hermanns from Saarland University, Germany, Prof. Wan Fokkink from Vrije Universiteit Amsterdam, The Netherlands, Prof. Andreas Zeller from CISPA Helmholtz Center for Information Security, Germany, and Prof. Yongwang Zhao from Zhejiang University, China. This year, together with Prof. Zhiming Liu from Southwest University, China, two special tracks were organized in addition to the research papers track: one special track on Artificial Intelligence Meets Formal Methods to provide a platform for experts of both AI and FM, from both academia and industry, to discuss important research problems across these two areas, and the other journal first papers track implemented in partnership with the *Journal of Computer Science and Technology*. The conference program also consists of the presentations selected from these two special tracks.

SETTA 2020 is sponsored by the Institute of Software, Chinese Academy of Sciences, China, and organized by the Institute of Intelligent Software, Guangzhou, China. We are grateful to the Local Organizing Committee for their hard work in making SETTA 2020 a successful event. Our warmest thanks go to the authors for submitting their papers to the conference. We thank the members of the Steering Committee for their support in organizing this event. We thank all the members of the Program Committee (PC) for completing reviews on time and being active in discussions during the review process. We also thank the additional reviewers for their effort that helped the PC to decide which submissions to accept. Special thanks go to our invited speakers for presenting their research at the conference. Finally, we thank the conference general chair, Prof. Huimin Lin, the publicity chair, Dr. Fu Song, and the local organization and Web chairs, Prof. Meng Sun and Dr. Chengchao Huang.

September 2020

Jun Pang
Lijun Zhang

Organization

Program Committee

Ezio Bartocci	Vienna University of Technology, Austria
Lei Bu	Nanjing University, China
Milan Ceska	Brno University of Technology, Czech Republic
Sudipta Chattopadhyay	Singapore University of Technology and Design, Singapore
Yu-Fang Chen	Academia Sinica, Taiwan
Alessandro Cimatti	Fondazione Bruno Kessler, Italy
Yuxin Deng	East China Normal University, China
Wei Dong	National University of Defense Technology, China
Hongfei Fu	Shanghai Jiao Tong University, China
Jan Friso Groote	Eindhoven University of Technology, The Netherlands
Nan Guan	The Hong Kong Polytechnic University, Hong Kong
Dimitar Guelev	Bulgarian Academy of Sciences, Bulgaria
Xiaowei Huang	The University of Liverpool, UK
Nils Jansen	Radboud University, The Netherlands
Yu Jiang	Tsinghua University, China
Sebastian Junges	University of California, Berkeley, USA
Zhiming Liu	Southwest University, China
Stefan Mitsch	Carnegie Mellon University, USA
Jean-Francois Monin	Verimag and Université de Grenoble, France
Mohammadreza Mousavi	University of Leicester, UK
Sebastian A. Mödersheim	Technical University of Denmark, Denmark
Jun Pang	University of Luxembourg, Luxembourg
Dave Parker	University of Birmingham, UK
Mickael Randour	Université de Mons, Belgium
Zhiping Shi	Capital Normal University, China
Fu Song	ShanghaiTech University, China
Jeremy Sproston	University of Turin, Italy
Jun Sun	Singapore Management University, Singapore
Meng Sun	Peking University, China
Cong Tian	Xidian University, China
Andrea Turrini	Institute of Software, Chinese Academy of Sciences, China
Tarmo Uustalu	Reykjavik University, Iceland
Jaco van de Pol	Aarhus University, Denmark
Chenyi Zhang	Jinan University, China
Lijun Zhang	Institute of Software, Chinese Academy of Sciences, China

Additional Reviewers

Ahmadi, Mohamadreza
Becchi, Anna
Dong, Naipeng
Groß, Dennis
H. Pham, Long
Haase, Christoph
Hu, Chi
Huang, Mingzhang
Jegourel, Cyrille
Laveaux, Maurice
Li, Jiaying
Li, Ximeng
Liyun, Dai
Lu, Yuteng
Martens, Jan
Novotný, Petr
Quatmann, Tim
Sangnier, Arnaud
Shi, Hao
Suilen, Marnix
Turrini, Andrea
Wang, Rui
Willemse, Tim
Wu, Yuming
Xia, Bican
Yang, Yilong
Yang, Zhengfeng
Zhang, Liang
Zhang, Min
Zhang, Qianying
Zhang, Xiyue
Zhang, Yuanrui

Abstracts of Invited Talks

Lab Conditions for Research on Explainable Automated Decisions

Holger Hermanns[1,2]

[1] Saarland University – Computer Science, Saarland Informatics Campus, Saarbrücken, Germany
[2] Institute of Intelligent Software, Guangzhou, China

Artificial neural networks are being proposed for automated decision making under uncertainty in many visionary contexts, including high-stake tasks such as suggesting which patient to grant a life-saving medical treatment, or navigating autonomous cars through dense traffic. Against this background, it is imperative that the decision making entities meets central societal desiderata regarding dependability, perspicuity, explainability, and robustness.

Decision making problems under uncertainty are typically captured formally as variations of Markov decision processes (MDPs). This keynote discusses a set of natural and easy to-control abstractions that altogether connect the autonomous driving challenge to the modelling world of MDPs. This is then used to study the dependability and robustness of NN-based decision entities which in turn are based on state-of-the-art NN learning techniques. We argue that this approach can be regarded as providing laboratory conditions for a systematic, structured and extensible comparative analysis of NN behaviour, of NN learning performance, as well as of NN verification and analysis techniques.

Holger Hermanns—This work receives financial support by the ERC Advanced Investigators Grant 695614 (POWVER), by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Grant 389792660 as part of TRR 248, see https://perspicuous-computing.science, and by the Key-Area Research and Development Program of Guangdong Province (Grant 2018B010107004).

Rely-Guarantee Reasoning About Concurrent Reactive Systems: Framework, Languages Integration and Applications

Yongwang Zhao

School of Cyber Science and Technology, College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310007, China
zhaoyongwang@gmail.com

This talk presents PiCore, a rely-guarantee reasoning framework for formal specification and verification of concurrent reactive systems (CRSs). PiCore takes the level of abstraction and reusability of the rely-guarantee method a step further by proposing an expressive event language for complex reaction structures at the system level, as well as decoupling the system and program levels. The result is a flexible rely-guarantee framework for CRSs, which is able to integrate existing rely-guarantee implementations at program level without any change of them. PiCore introduces the notion of "events" into the rely-guarantee method for system reactions and provides a *rely-guarantee interface* which is an abstraction for common rely-guarantee components for the program level. Concrete languages used to model the behaviour of events can be easily integrated with PiCore by a *rely-guarantee adapter* which implements the rely-guarantee interface. This design allows PiCore to be independent of program languages and thus to easily reuse existing rely-guarantee frameworks. To deal with complex reaction structures, we design an event specification language in PiCore supporting structural compositions of events. An event system is a structural composition of a set of events.

We have integrated two existing languages (Hoare-Parallel and CSimpl) and their rely-guarantee proof systems into the PiCore framework. As a result we create two instances of PiCore. Then, we apply the instances of the PiCore framework to two case studies, i.e. a real-world concurrent RTOS (Zephyr) and a standard of business process execution language (BPEL). We have applied PiCore to the formal specification and mechanized proof of the concurrent buddy memory allocation of Zephyr RTOS. The formal specification is fine-grained providing a high level of detail. It closely follows the Zephyr C code, covering all the data structures and imperative statements present in the implementation. We use the rely-guarantee proof system of PiCore for the formal verification of functional correctness and invariant preservation in the model, revealing three bugs in the C code. We have applied PiCore to interpret the semantics of the BPEL language by translating BPEL into PiCore. To show the correctness of this translation, we prove a strong bisimulation between the source BPEL program and the translated PiCore specification. In this way, formal verification of BPEL programs can be conducted in the PiCore framework. The strong bisimulation implies the soundness and completeness of formal verification of BPEL program in PiCore.

Contents