Lecture Notes in Computer Science 10998

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

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

Xinyu Feng · Markus Müller-Olm
Zijiang Yang (Eds.)

Dependable Software Engineering

Theories, Tools, and Applications

4th International Symposium, SETTA 2018
Beijing, China, September 4–6, 2018
Proceedings

Springer

Editors
Xinyu Feng
Nanjing University
Nanjing
China

Zijiang Yang
Western Michigan University
Kalamazoo, MI
USA

Markus Müller-Olm
Westfälische Wilhelms-Universität Münster
Münster
Germany

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-99932-6 ISBN 978-3-319-99933-3 (eBook)
https://doi.org/10.1007/978-3-319-99933-3

Library of Congress Control Number: 2018953821

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 papers presented at the 4th International Symposium on Dependable Software Engineering: Theories, Tools, and Applications (SETTA 2018), held during September 4–6, 2018, in Beijing. The purpose of SETTA is to provide an international forum for researchers and practitioners to share cutting-edge advancements and strengthen collaborations in the field of formal methods and its interoperability with software engineering for building reliable, safe, secure, and smart systems. Past SETTA symposiums were successfully held in Nanjing (2015), Beijing (2016), and Changsha (2017).

SETTA 2018 solicited submissions in two categories, regular papers and short papers. Short papers could discuss ongoing research at an early stage, or present systems and tools. There were 22 submissions in total. Each submission was reviewed by at least three, and on average 3.7, Program Committee (PC) members, with the help of external reviewers. After thoroughly evaluating the relevance and quality of each paper through online PC meetings, the PC decided to accept nine regular papers and three short papers. The program also included three invited talks given by Prof. Moshe Vardi from Rice University, Prof. Tao Xie from University of Illinois at Urbana-Champaign, and Prof. Hongseok Yang from KAIST. Prof. Moshe Vardi was a joint keynote speaker of CONFESTA 2018, a joint event comprising the international 2018 conferences CONCUR, FORMATS, QEST, and SETTA, alongside with several workshops and tutorials.

This program would not have been possible without the unstinting efforts of many people, whom we would like to thank. First, we would like to express our gratitude to the PC and the external reviewers for their hard work put in toward ensuring the high quality of the proceedings. Our thanks also go to the Steering Committee for its advice and help. We would like to warmly thank the general chair of SETTA 2018, Prof. Chaochen Zhou, the general chair of CONFESTA 2018, Prof. Huimin Lin, the local organizers including Dr. David N. Jansen, Dr. Andrea Turrini, Dr. Shuling Wang, Dr. Peng Wu, Dr. Zhilin Wu, Dr. Bai Xue, Prof. Lijun Zhang, and all others on the local Organizing Committee.

We also enjoyed great institutional and financial support from the Institute of Software, Chinese Academy of Sciences, without which an international conference like CONFESTA and the co-located events could not have been successfully organized. We also thank the Chinese Academy of Sciences and the other sponsors for their financial support. Furthermore, we would like to thank Springer for sponsoring the Best Paper Award. Finally, we are grateful to the developers of the EasyChair system, which significantly eased the processes of submission, paper selection, and proceedings compilation.

July 2018

Xinyu Feng
Markus Müller-Olm
Zijiang Yang

Organization

General Chair

Chaochen Zhou	Institute of Software, CAS, China

Program Chairs

Xinyu Feng	Nanjing University, China
Markus Müller-Olm	Westfälische Wilhelms-Universität Münster, Germany
Zijiang Yang	Western Michigan University, USA

Program Committee

Farhad Arbab	CWI and Leiden University, The Netherlands
Sanjoy Baruah	Washington University in St. Louis, USA
Lei Bu	Nanjing University, China
Michael Butler	University of Southampton, UK
Yan Cai	Institute of Software, CAS, China
Taolue Chen	Birkbeck, University of London, UK
Yuxin Deng	East China Normal University, China
Xinyu Feng	Nanjing University, China
Yuan Feng	University of Technology, Sydney, Australia
Ernst Moritz Hahn	Institute of Software, CAS, China
Dan Hao	Peking University, China
Maritta Heisel	University of Duisburg-Essen, Germany
Raymond Hu	Imperial College London, UK
He Jiang	Dalian University of Technology, China
Yu Jiang	Tsinghua University, China
Einar Broch Johnsen	University of Oslo, Norway
Guoqiang Li	Shanghai Jiao Tong University, China
Ting Liu	Xi'an Jiaotong University, China
Tongping Liu	University of Texas at San Antonio, USA
Yang Liu	Nanyang Technological University, Singapore
Xiapu Luo	The Hong Kong Polytechnic University, Hong Kong, SAR China
Stephan Merz	Inria Nancy and LORIA, France
Markus Müller-Olm	Westfälische Wilhelms-Universität Münster, Germany
Jun Pang	University of Luxembourg, Luxembourg
Davide Sangiorgi	University of Bologna, Italy
Oleg Sokolsky	University of Pennsylvania, USA
Fu Song	ShanghaiTech University, China
Zhendong Su	University of California, Davis, USA

Jun Sun	Singapore University of Technology and Design, Singapore
Walid Mohamed Taha	Halmstad University and University of Houston, Sweden
Sofiene Tahar	Concordia University, Canada
Cong Tian	Xidian University, China
Bow-Yaw Wang	Academia Sinica, Taiwan
Chao Wang	University of Southern California, USA
Ji Wang	National University of Defense Technology, China
Heike Wehrheim	University of Paderborn, Germany
Xin Xia	Monash University, Australia
Zijiang Yang	Western Michigan University, USA
Shin Yoo	KAIST, Korea

Local Organizing Committee

Jansen, David N.
Lv, Yi
Turrini, Andrea
Wang, Shuling
Wu, Peng
Wu, Zhilin (chair)
Xue, Bai
Yan, Rongjie
Zhu, Xueyang

Additional Reviewers

Chen, Liqian
Chen, Zhe
Colley, John
Dongol, Brijesh
Elderhalli, Yassmeen
Gu, Tianxiao
Gutsfeld, Jens
Kaur, Ramneet
Kenter, Sebastian
Kharraz, Karam
König, Jürgen
Liu, Hongyu
Petre, Luigia
Santen, Thomas
Silvestro, Sam
Soualhia, Mbarka
Steffen, Martin
Tang, Enyi
Zhang, Min
Zhu, Chenyang

Abstracts of Invited Talks

Intelligent Software Engineering: Synergy between AI and Software Engineering

Tao Xie

University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
taoxie@illinois.edu

Abstract. As an example of exploiting the synergy between AI and software engineering, the field of intelligent software engineering has emerged with various advances in recent years. Such field broadly addresses issues on *intelligent* [software engineering] and [*intelligence software*] engineering. The former, *intelligent* [software engineering], focuses on instilling intelligence in approaches developed to address various software engineering tasks to accomplish high effectiveness and efficiency. The latter, [*intelligence software*] engineering, focuses on addressing various software engineering tasks for intelligence software, e.g., AI software. In this paper, we discuss recent research and future directions in the field of intelligent software engineering.

This work was supported in part by National Science Foundation under grants no. CNS-1513939 and CNS1564274, and a grant from the ZJUI Research Program.

Formal Semantics of Probabilistic Programming Languages: Issues, Results and Opportunities

Hongseok Yang

KAIST, Daejeon, South Korea
hongseok00@gmail.com

Probabilistic programming refers to the idea of developing a programming language for writing and reasoning about probabilistic models from machine learning and statistics. Such a language comes with the implementation of several generic inference algorithms that answer various queries about the models written in the language, such as posterior inference and marginalisation. By providing these algorithms, a probabilistic programming language enables data scientists to focus on designing good models based on their domain knowledge, instead of building effective inference engines for their models, a task that typically requires expertise in machine learning, statistics and systems. Even experts in machine learning and statistics may get benefited from such a probabilistic programming system because using the system they can easily explore highly advanced models.

In the past three years, I and my colleagues have worked on developing so called denotational semantics of such probabilistic programming languages, especially those that support expressive language features such as higher-order functions, continuous distributions and general recursion. Such semantics describe what probabilistic model each program in those languages denotes, serve as specifications for inference algorithms for the languages, and justify compiler optimisations for probabilistic programs or models. In this talk, I will describe what we have learnt so far, and explain how these lessons help improve the design and implementation of these probabilistic programming languages and their inference engines.

Contents