

## Special section on realizing artificial intelligence synergies in software engineering

Çetin Meriçli<sup>1</sup> · Burak Turhan<sup>2</sup>

Published online: 1 February 2017  
© Springer Science+Business Media New York 2017

With its boundaries expanding into other disciplines and fields as computers become more and more ubiquitous, software engineering (SE) is expected to solve a plethora of increasingly complex questions that are dynamic, automated, and adaptive or must execute at a very large scale. In theory, there could be another layer of collaboration between SE and other disciplines in addition to employing software systems of varying complexity. For example, artificial intelligence (AI) technologies can support the development of increasingly complex SE systems as in the case of recommendation systems. Conversely, in theory, SE might also play a role in alleviating development costs and the development effort associated with AI tools and applications such as robotics where proper development and testing practices are of utmost importance from both cost and robustness perspectives. Unfortunately, in practice, such collaborations between SE and AI are rarely achieved.

This special section is the joint result of invited papers from the RAISE'14 Workshop on Realizing Artificial Intelligence Synergies in Software Engineering and an open call for contributions to address the issues above. All submissions were reviewed by experts in the field. Finally, three papers were selected for inclusion in this special section. These papers cover specific AI/SE synergies such as experiments with boosting-based cost-sensitive transfer learning for cross-project defect prediction, a systematic review on the applications of Bayesian networks in software quality estimation, and case studies in learning developer cues in the context of program comprehension.

We hope that these papers will be useful in reflecting the type of discussions in the RAISE Workshop series, as well as in providing a view of the state of the art in the research areas covered by each paper. Before we provide brief summaries of the papers, we would like to extend our sincere thanks to the authors for their contributions, reviewers for their help and high quality feedback, and the Editor-in-Chief for making this special issue possible.

---

✉ Çetin Meriçli  
cetin@cmu.edu

<sup>1</sup> Carnegie Mellon University, Pittsburgh, PA, USA

<sup>2</sup> University of Oulu, Oulu, Finland

In “A transfer cost-sensitive boosting approach for cross-project defect prediction,” Duksan Ryu, Jong-In Jang, and Jongmoon Baik provide an experimental benchmark study in cross-project defect prediction (CPDP). The authors propose a new learning technique called transfer cost-sensitive boosting (TCSBoost), evaluate its performance on 15 datasets, and benchmark their results against the state-of-the-art cross-project defect prediction techniques. Addressing class imbalance problems with SMOTE, TCSBoost shows a significantly better performance than the state-of-the-art approaches. The study also reports that using limited amounts of within project data does not improve the performance of TCSBoost further. The authors conclude that their approach can help reduce the cost of software quality assurance activities.

In “A systematic literature review on the applications of Bayesian networks to predict software quality,” Ayse Tosun, Ayse Basar Bener, and Shirin Akbarinasaji report on a systematic literature review on the applications of Bayesian networks (BNs) to predict software quality. Motivated by the premise on the use of BNs in decision-making, the review identifies 10 primary studies and evaluates dataset characteristics, techniques used for parameter learning and structured learning, and use of tools and model validation techniques in these primary studies. Highlighting the shortcomings in reporting the applications of BNs in the software engineering domain, the authors provide a set of guidelines for reporting the essential contextual and methodological details in reporting BN applications in software quality prediction to support further studies including replications.

In “Supporting comprehension of unfamiliar programs by modeling cues,” Naveen Kulkarni and Vasudeva Varma report on two case studies on the use of cues for program comprehension. In the first case study, authors use standard comprehension tasks with the goal of identifying a set of cues that developers use to explore new programs. In the second case study, they investigate the effect of IDEs on the comprehension process. Then, the authors propose an approach to construct task-specific program summaries based on the identified cue choices. Their evaluation leads to the conclusion that their approach can be beneficial for novice developers when investigating unfamiliar programs by suggesting the right cues for them for better program comprehension.



**Dr. Çetin Meriçli** is a Special Faculty at the Special Faculty at the National Robotics Engineering Center (NREC), part of Carnegie Mellon University’s Robotics Institute, where he leads the Machine Learning and Perception group.

His research focus is on self-driving cars, specifically on urban and offroad autonomous and semiautonomous driving. Some of his other current research interests can be listed as machine learning, computer vision, imitation learning, robot learning from demonstration, robust high fidelity outdoor perception, efficient behavior acquisition, customization, and correction through human feedback, situational awareness, human-robot interaction, data-driven large scale accurate simulation systems for testing robot autonomy, and safe and efficient human-robot collaboration. He has over 40 publications on these subjects. Apart from his academic achievements, he also has extensive hands-on experience in designing, developing, and maintaining high performance high availability robot and software systems ranging from service robots and interactive expressive humanoid torsos to unmanned military ground vehicles. He additionally has had successes at RoboCup International including a world championship.

Prior to joining NREC, he was a post-doctoral fellow in the Computer Science Department at Carnegie Mellon University. Dr. Meriçli received his Ph.D. in Computer Science from Boğaziçi University, Turkey.



**Burak Turhan** is a full Professor of Software Engineering at the Faculty of Information Technology and Electrical Engineering at the University of Oulu, Finland. His research interests are focused on: empirical studies of software quality and programmer productivity, software analytics through the application of machine learning and data mining methods for defect and cost modeling, studying human factors in software engineering – particularly cognitive biases –, and mining software repositories for grounded decision making, as well as agile/lean software development with a special focus on test-driven development. He is a member of the editorial board of Empirical Software Engineering Journal, a steering group member and General Chair for PROMISE conference, and PC co-chair for PROMISE'13, ESEM'17 and PROFES'17 conferences. He is a member of IEEE, IEEE Computer Society, ACM, and ACM SIGSOFT. For more information and details please visit <https://turhanb.net>.