



Preface to the Special Issue on Program Comprehension

David Lo¹ · Alexander Serebrenik²

Published online: 17 October 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

We are delighted to present a selection of the best papers presented at the 25th IEEE International Conference on Program Comprehension (ICPC 2017) that took place in Buenos Aires, Argentina. The program committee has received 83 submissions originating from 97 abstracts and co-authored by researchers from 26 countries from Africa, Asia, Europe, North and South America and Oceania. This is more than double of the 39 submissions received back in 2000.¹

To select the papers for the special issue the PC chairs have selected the top five papers with the highest ratings from the reviewers. Each of these papers receives an average rating of ≥ 1.7 on the scale of -3 to 3 , with at least two reviewers giving 2 (“Accept”) or 3 (“Strong Accept”). Authors of the invited papers have significantly revised their work preparing the journal submission. Each of these submissions has been subject to additional reviewing. To obtain complementary perspectives, we have invited original reviewers—who reviewed the conference version of the work—and new ones. At the end of the reviewing process, we have a total of three extended papers that form this special issue.

One of the papers is by Almeida et al. who have won the best paper award. In their extended version, entitled “Investigating Whether and How Software Developers Understand Open Source Software Licensing”, the authors have conducted a survey of 375 developers that posed development scenarios involving three popular open source licenses (GNU GPL 3.0, GNU LGPL 3.0 and MPL 2.0) both alone and in combination. By comparing the answers of the developers with those of a legal expert, the authors observed that although developers clearly understood cases involving one license, presence of multiple licenses clearly hindered their understanding. Furthermore, the authors have analyzed license-related questions on a popular question-and-answer platform and interviewed several developers. The results call for tool support to help guide developers in understanding the structure of the code and the technical details of a project while taking into account the exact requirements imposed by the licenses involved.

¹While the history of the Workshop on Program Comprehension (WPC), a predecessor of ICPC, goes back to 1997 we could not find information on the submission numbers prior to 2000.

✉ David Lo
davidlo@smu.edu.sg

✉ Alexander Serebrenik
a.serebrenik@tue.nl

¹ Singapore Management University, 80 Stamford Road, Singapore 178902, Singapore

² Eindhoven University of Technology, P.O. Box 513, 5600MB, Eindhoven, The Netherlands

Another paper in this special issue is authored by Ceccato et al. entitled “Understanding the Behaviour of Hackers while Performing Attack Tasks in a Professional Setting and in a Public Challenge”. It focuses on comprehension in a very specific and less studied context of security attacks. Indeed, as noted by the authors comprehension strategies of hackers trying to attack the protected software might differ from program understanding for benign purposes. The authors have conducted three industrial case studies with the involvement of professional penetration testers and a public challenge consisting of eight attack tasks with open participation. Based on the case studies, the authors developed a taxonomy consisting of 169 concepts, covering activities related to different stages of comprehension.

The final paper included in this special issue, “Syntax, Predicates, Idioms—What Really Affects Code Complexity?” by Ajami et al. study factors that affect program comprehension. In particular, the authors report on an experiment in an online game-like environment to measure how quickly and accurately 220 professional programmers can interpret code snippets with similar functionality but different structures. The authors have observed differences in comprehension of code snippets in terms of comprehension time and correctness. For instance, *for*-loops are significantly harder to comprehend than *ifs*, loops counting down slightly require more time to comprehend than those counting up, and loops with unusual bounds result in more errors. The authors highlight implications of their findings on understanding limitations of existing complexity metrics and designing better ones.

We hope that you will enjoy reading these article extensions of some of the best papers presented at ICPC 2017 and cordially invite you to participate in and get involved with the welcoming ICPC community!



David Lo received his PhD degree from the School of Computing, National University of Singapore in 2008. He is currently an Associate Professor in the School of Information Systems, Singapore Management University. He has more than 10 years of experience in software engineering and data mining research and has more than 200 publications in these areas. He received the Lee Foundation and Lee Kong Chian Fellow for Research Excellence from the Singapore Management University in 2009 and 2018, and a number of international research and service awards including multiple ACM distinguished paper awards for his work on software analytics. He has served as general and program co-chair of several prestigious international conferences (e.g., IEEE/ACM International Conference on Automated Software Engineering), and editorial board member of a number of high-quality journals (e.g., Empirical Software Engineering).



Alexander Serebrenik (PhD, K.U. Leuven, Belgium 2003; MSc, Hebrew University, Israel, 1999), senior member of IEEE, is an Associate Professor of software evolution at Eindhoven University of Technology. His research covers a wide range of topics, from source code analysis, to collaborative and human aspects of software engineering. He has co-authored a book “Evolving Software Systems” (Springer Verlag, 2014), and more than 100 scientific papers and articles. He has won Distinguished Paper awards at the International Conference on Software Engineering (2017) and the International Conference on the Quality of Information and Communications Technology (2014), as well as special contribution awards at ESEM 2018 and MSR 2017. He is the steering committee chair of the International Conference on Software Maintenance and Evolution. He is member of the editorial boards of *Empirical Software Engineering* (Springer Verlag), *Journal of Systems and Software* (Elsevier), and *Science of Computer Programming*; he has also serves or served on the program committees of such software engineering conferences as ICSE, ESEC/FSE, ICSM(E), MSR, SANER and ICPC, winning several Distinguished Reviewer awards.