

1. A closer look at real-world patches

Kui Liu (1); Dongsun Kim (1); Koyuncu, A. (1); Li Li (2); Bissyande, T.F. (1); Le Traon, Y. (1)

Source: 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). Proceedings, p 275-86, 2018; **DOI:** 10.1109/ICSME.2018.00037; **Conference:** 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), 23-29 Sept. 2018, Madrid, Spain; **Publisher:** IEEE Computer Society, Los Alamitos, CA, USA

Author affiliation: (1) University of Luxembourg, Interdisciplinary Centre for Security, Reliability and Trust, Luxembourg (2) Monash University, Monash Software Force, Melbourne, VIC, Australia

Abstract: Bug fixing is a time-consuming and tedious task. To reduce the manual efforts in bug fixing, researchers have presented automated approaches to software repair. Unfortunately, recent studies have shown that the state-of-the-art techniques in automated repair tend to generate patches only for a small number of bugs even with quality issues (e.g., incorrect behavior and nonsensical changes). To improve automated program repair (APR) techniques, the community should deepen its knowledge on repair actions from real-world patches since most of the techniques rely on patches written by human developers. Previous investigations on real-world patches are limited to statement level that is not sufficiently fine-grained to build this knowledge. In this work, we contribute to building this knowledge via a systematic and fine-grained study of 16,450 bug fix commits from seven Java open-source projects. We find that there are opportunities for APR techniques to improve their effectiveness by looking at code elements that have not yet been investigated. We also discuss nine insights into tuning automated repair tools. For example, a small number of statement and expression types are recurrently impacted by real-world patches, and expression-level granularity could reduce search space of finding fix ingredients, where previous studies never explored. (0 refs)

Inspec controlled terms: Java - program debugging - public domain software - software maintenance

Uncontrolled terms: real-world patches - bug fixing - software repair - automated program repair techniques - APR techniques - human developer - Java open-source project - expression-level granularity

Classification Code: C6150G Diagnostic, testing, debugging and evaluating systems - C6110B Software engineering techniques - C6110J Object-oriented programming

IPC Code: G06F9/44 - G06F11/36

Treatment: Bibliography (BIB) - Practical (PRA)

Database: Inspec

Data Provider: Engineering Village

Copyright 2018, The Institution of Engineering and Technology