

**Review of Z. Gao, Z. Chen, Y. Zou, and A.M. Memon, 2016,
SITAR: GUI Test Script Repair
IEEE Transactions on Software Engineering 42 (2): 170–186**

**T.H. Tse
Department of Computer Science
The University of Hong Kong
Pokfulam, Hong Kong**

The testing of a graphical user interface (GUI) often requires extensive human interactions with the system, which is tedious, time consuming, and costly. Even if failures are detected, testers cannot tell whether they are due to human errors or system faults. To solve the problem, GUI test cases are often simulated by automatic test scripts. However, when any GUI widget is redesigned, parts of the original test scripts will not be usable. This paper proposes a SITAR script repairer process to maintain the test scripts.

SITAR consists of three phases: (1) ripping, which extracts the key attribute transitions described by the test scripts into an abstract event-flow graph model through reverse engineering; (2) mapping, which further associates the events between the scripts and the model; and (3) automatic repair, which may be confirmed, modified, or supplemented by additional events by the human tester. An empirical study has been conducted to measure the percentage of unusable test scripts, the percentage of automatic ripping, the effectiveness of mappings, the percentage of stable events, the cost of repair, and the percentage of scripts not repaired. Results show that 41 to 89 percent of the unusable test scripts are repaired. Human cost is significantly reduced. The paper should be of great interest to readers faced with human resource issues in GUI software testing.

Is there any shortcoming of this study? The response by the authors to this practical question is rather intriguing. They state that weaknesses are due to the large number of human subjects, the large pool of test scripts, the repeated repairs to avoid errors, and the manual review of unsuccessful cases. Readers are, however, not given any clue about the real deficiencies. I cannot help but provide the young authors with the same advice that I give to graduating students when they attend an interview: Do not disguise your strengths as weaknesses. Instead, present the true weaknesses with an open mind and discuss how they are being overcome.