Automating State to State testing of mobile applications to mobile platforms*

Edition 1†

Jonathan Sanders
Dr. Justice

ABSTRACT

This paper outlines a new way to test a devices states' affect on an application.

KEYWORDS

automata for mobile device testing,automated state testing

ACM Reference Format:
Jonathan Sanders and Dr. Justice. 1997. Automating State to State testing of mobile applications to mobile platforms: Edition 1. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97).* ACM, New York, NY, USA, Article 4, 1 page. https://doi.org/10.475/123_4

1 INTRODUCTION

A common problem with automated mobile software testing is testing mobile applications when the state of the device changes. Testing the different states of an application is already an expensive process. The problem with this issue is that often, the state of a device will cause a catastrophic failure of a feature within an application when the state of a mobile device changes; therefore, it is more expensive to not test device states against application states at all. Since the state of an application can be easily saved and the state of a device can be easily changed programmatically, a solution can be made that automatically tests application states with different device states. For applications being used in different device states, such as a mapping application used outside of network range, this becomes an extremely important issue. This author was once almost lost in the mountains because of one such issue.

We have developed a testing procedure, middle ware library, and runtime that records the state of the objects of a mobile application whenever the tester wants to and then evaluates the state and sub-states [4] of the objects again when a phone state has changed. The library is primarily concerned with identifying first exceptions being thrown and then three different types of error states [4].

The main purpose of this library is to simplify testing of different device states with the application. There are 52 preloaded "device state changes" (DVCs) that can be used in a test suite to test the app states that are chosen for testing. A DVC is a test case that starts with a device state then samples the chosen beginning app state, then changes the Devices state then sampling the app state again and evaluating if there were any exceptions thrown or errors detected. A few examples of DVCs are: airplane mode off, sample the app state, switch airplane mode to on, resample the app state to evaluate.

There are several tools developed that can capture various information of an application and store said information for later tests. These tools can be leveraged to create interesting app states instead of simple object states. For instance the authors of [2] built a tool that automatically captures UI information as a user utilizes an application. The researchers with[1] made a tool that can store UI information and then auto generate oracles that then get scripted into an Espresso test script for later use as a test case. This tool could be used to run test cases as the DVCs execute as the state instead of static properties being evaluated.

"The field of mobile specific, black-box testing, however, remains thin." [3] That is the reasoning behind this endeavor to make easier testing tools that easily test the interaction of device states with applications states.

REFERENCES

[1] M. Fazzini, E. N. D. A. Freitas, S. R. Choudhary, and A. Orso. 2017. Barista: A Technique for Recording, Encoding, and Running Platform Independent Android Tests. In *2017 IEEE International Conference on Software Testing, Verification and Validation (ICST)*. 149–160. https://doi.org/10.1109/ICST.2017.21

[2] F. Paulovsky, E. Pavese, and D. Garbervetsky. 2017. High-Coverage Testing of Navigation Models in Android Applications. In *2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)*. 52–58. https://doi.org/10.1109/AST.2017.6

[3] Fernando Paulovsky, Esteban Pavese, and Diego Garbervetsky. 2017. High-coverage testing of navigation models in Android applications. In *Proceedings of the 12th International Workshop on Automation of Software Testing*. IEEE Press, 52–58.

[4] C. D. Turner and D. J. Robson. 1993. The state-based testing of object-oriented programs. In *1993 Conference on Software Maintenance*. 302–310. https://doi.org/10.1109/ICSM.1993.366932

*Produces the permission block, and copyright information
†The full version of the author's guide is available as `acmart.pdf` document

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4