Poster: A Black-Box based Android GUI Testing System

Chao-Chun Yeh*‡
avainyeh@itri.org.tw

Shih-Kun Huang† ‡
skhuang@cs.nctu.edu.tw

Sung-Yen Chang*
madchang@itri.org.tw

*Information and Communications Research Laboratories
Industrial Technology Research Institute
Hsinchu, Taiwan

† Information Technology Service Center
‡ Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan

ABSTRACT
In Android system, black box testing has risen to three key issues: S1: There is no source code and for tester to know the internal logic of the testing App. S2: There is no testing criterion for tester to know the correct behavior and testing scope of the testing App. S3: It is difficult to measure the testing coverage without instrumentation the testing App. In this paper, we provide an approach to analyze the GUI model during the testing process, implement the black-box based android GUI testing system and select 7 Apps for evaluation. Finally we compare our result of our system with the monkey tool [1] and discuss the inner App's properties that influence on the testing result.

Categories and Subject Descriptors
D.2.5 [**Testing and Debugging**]: *Testing tools*

Keywords
market App software, software testing, software quality.

1. System Prototype Design

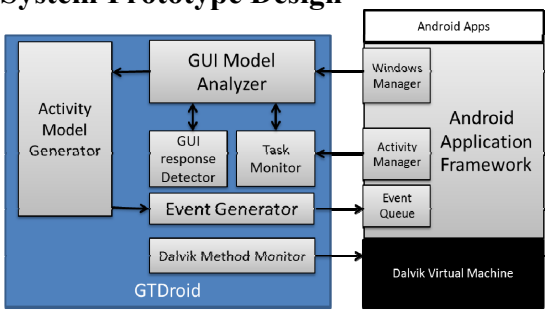

Figure 1. The Architecture of GTDroid

Figure 1 shows the architecture of our black-box Android GUI testing system– GTDroid, our proposed approach to address the issues, and explain the essential steps involved in testing an App with GTDroid. First, the App to be tested is installed and executed automatically in the emulator and then the GTDroid interacts with the android application framework modules and dalvik virtual machines. For S1, GUI model analyzer fetch GUI components information from windows manager and then analysis the layout of the current view. The analytic result sent to activity model generator for building the whole App Activity Model. For S2, the task monitor detect the current testing activity status such as task stack information and crash event with activity manager and event detector check the GUI based information from GUI model Analyzer such as ANR (Application Not Responding), exceptions and user define message. For S3, after event generator send events to the event queue and some dalvik methods might be executed by the related events. Dalvik method monitor inspects the executed methods with modified dalvik virtual machine and provide unique method count (UMC) for comparison with other test results.

2. EXPERIMENTS
Our experiment environment includes Ubuntu 11.04, Android emulator and we implement GUI model analyzer and event generator modules based on jython 2.5 and other modules on python 2.7. To evaluate GTDroid, we use 7 benchmark Apps from the bundle Apps in the emulator and the high download rate Apps in Google Play Taiwan Market [2]. We run each App in twenty minutes, collect UMC for coverage and compare the result with monkey tool. The results are presented in Fig. 2. In the result, we compare the UMC with GTDroid and Monkey tool. In average case, GTDroid can obtain more 27.02% UMC than Monkey with the base case (82.69%) and worse case (-1.82%).

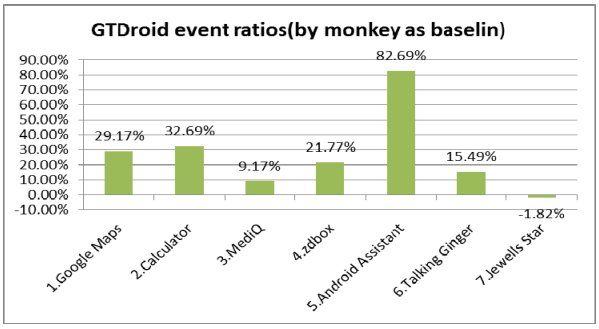

Figure 2 GTDroid UMC ratios

3. DISCUSSION & FUTURE WORK
In term of search strategy, Monkey is based on random events without any background knowledge such App module or the widgets in the top-most view. With this restraint, Monkey based test will case many duplicated tests. In GTDroid, we build App model and make cycle detection rule to prevent executing duplicated tests. In our 5th sample, we analysis the log and find there are many duplicated tests with Monkey tool like profile setting or data-filling in dialog. In our samples, some Apps (3th and 7th) are designed with single activity and simple control flow. There is no distinguished improvement between GTDroid and Monkey. While GTDroid demonstrates the results, it is still a prototype system and there are future work to overcome the limitations that it cannot detect user define widget and the coverage information should be more accuracy.

4. REFERENCES
[1] UI/Application Exerciser Monkey- http://developer.android.com/tools/help/monkey.html
[2] Android Apps on Google play- https://play.google.com/store

MobiSys'13, June 25–28, 2013, Taipei, Taiwan.
ACM 978-1-4503-1672-9/13/06.

This work is a partial result of Project No. C3522Q1200 conducted by Industrial Technology Research Institute under sponsorship of the Ministry of Economic Affairs, R.O.C., Taiwan.