











































SwiftHand<sup>[16]</sup>通过动态建立 GUI 的有限状态机模型,寻求以最少事件数达到最高的测试覆盖率。ORBIT<sup>[41]</sup>基于动态 GUI 抓取和静态代码分析方法,以避免生成不相关的 UI 事件。然而最近实证研究表明<sup>[7]</sup>,所有现有技术的覆盖率都很低(小于 50%)。

## 6.2 AndroidRecord/Replay框架

目前,工业界和学术界已有多针对 Android 应用交互行为录制/重放的框架。工业界相关研究如 TestIn<sup>[18]</sup>,以在不同型号的真机上进行测试为主,通过将应用运行在不同分辨率、不同尺寸和其他硬件配置的真机上完成对应用适配性的测试。在学术界,以 RERAN<sup>[12]</sup>为代表的记录底层日志方法,通过记录 Android 底层接口得到系统操作日志流,以完成测试用例的重放,其精确度可以达到毫秒级。但单纯记录操作日志流方法只能在录制的智能设备上重放,无法实现跨设备重放。Mosaic<sup>[13]</sup>是 RERAN<sup>[12]</sup>工作的进一步扩展,通过将移动设备的指令集进行抽象化处理,将设备事件抽象为 Press、Move、Release 等有限的几个动作,来描述用户的所有行为,以达到跨设备重放用户行为的目的,其系统开销仅为 0.2%,并能跨设备和平台重放 GooglePlay 中的 45 个应用。但 Mosaic 存在两方面的局限性:(1) 在指令集抽象方面,将移动设备复杂的指令集抽象为简单几个动作,以这种方式生成测试用例不具备可读性,无法重用;(2) 在跨设备自适应方面,由于 Mosaic 采用成比例缩放方式实现不同屏幕尺寸和分辨率的适配,当应用界面布局随着屏幕旋转或屏幕尺寸、分辨率发生变化后,该方法会失效。

以 VALERA<sup>[42]</sup>为代表的插桩方法虽然可以在多个智能设备上重放,但这种方法只能测试单一应用。当前,多个应用交互的场景越来越普及,单纯针对单个应用测试已经无法满足自动化测试需求。同时,对字节码插桩会造成系统额外开销,不能准确地评估应用运行情况;另外,对应用重签名会破坏应用自身的安全机制,如对支付宝、在线商城等应用程序重签名,会对应用程序自身的安全机制造成破坏。

SPAG<sup>[43]</sup>是基于录制/重放技术的 Android 应用测试工具,它通过事件批处理和智能等待等功能以减少重放过程中的不确定性,并整合 Sikuli IDE 以在每个 GUI 操作后通过断言(截图图片)验证应用是否正确执行。SPAG-C<sup>[44]</sup>是 SPAG 工作的进一步扩展,其目的是增加测试断言的可重用性而不影响测试精度。它从外部摄像机捕获程序截图,以进一步减少记录测试用例所需时间,并提高测试断言的可重用性而不影响准确性。

与以上录制重放框架不同:AppCheck 在录制阶段支持众包方式收集用户交互事件序列,并转换成平台无关的测试脚本;在重放阶段,借助于 Android 辅助功能服务支持直接在众包用户的设备上执行测试用例。

## 6.3 兼容性检测

当前,针对 Android 生态系统的碎片化问题已有一些工作进行研究。Han 等人研究了 HTC 和摩托罗拉在 Android 问题跟踪系统中的错误报告,指出,Android 生态系统是零散、缺乏可移植性的<sup>[45]</sup>。为了更好地理解 Android 应用程序中碎片诱导的兼容性问题,文献[33]对 191 个开源软件的兼容性问题进行了实证研究,总结了一些常见模式来检测 Android 应用程序中的兼容性问题,并设计实现了 FicFinder。Erfani 等人提出了 CHECKCAMP<sup>[46]</sup>自动化测试技术针对 iOS 和 Android 平台的原生应用,CHECKCAMP 通过抽象每个平台的执行序列,通过比较不同平台的执行序列来检测兼容性问题。X-Checker<sup>[47]</sup>是一种跨平台的应用程序开发框架(例如 Xamarin),它开发了一种差异测试技术,以识别这些框架在源平台和目标平台 API 的不一致性。

与以上兼容性检测方法相比,AppCheck 通过收集测试用例在不同设备上运行时生成的各种运行时相关数据(例如截图和布局信息)以检测兼容性问题,支持检测 Android 应用在不同设备上运行时产生的行为、布局以及性能等方面的一致性问题。

## 7 结 论

本文提出了一种基于录制/重放的 Android 应用众包测试方法,该方法的一个重要特点是,可以通过互联网以众包测试方式支持测试用例的生成和执行。基于该方法,我们实现了一种基于录制/重放的 Android 应用众包测试工具 AppCheck。通过我们的初步实验结果表明,AppCheck 可以有效实现众包测试环境下 Android 应用用户交互行为的跨设备录制/重放,并在重放完成后识别兼容性问题,改进了当前方法的不足。

**致谢** 在此感谢中科院软件研究所软件工程技术研究开发中心的老师们以及北京城市学院 2016 级徐自强、李林青、汪伟光同学,2017 级吕田田、潘泽、吕冉、徐舒宁、何祎昕、李月华同学,2018 级羡喻杰、梁倩、王兆钧同学,2019 级袁迈、曹艳琦、赵一萱、夏凡、侯钰坤同学对本文实验的大力协助。

### References:

- [1] There are 18796 different Android devices according to OpenSignals latest fragmentation report. 2018. <https://thenextweb.com/mobile/2014/08/21/18796-different-android-devices-according-opensignals-latest-fragmentation-report/>
- [2] An eye-opening report from OpenSignal found that there are at least 24 093 distinct Android devices in the wild. 2018. <http://www.techspot.com/news/61666-android-fragmentation-least-24093-distinct-devices-wild.html>
- [3] Google I/O 2017 by the numbers: 2 billion Android devices, 500 million photos users, and more. 2017. <http://gadgets.ndtv.com/apps/news/google-io-2017-android-drive-photos-users-youtube-maps-1695234>
- [4] Choi W, Necula G, Sen K. Guided GUI testing of Android apps with minimal restart and approximate learning. In: Proc. of the OOPSLA. 2013. 623–640.
- [5] Machiry A, Tahiliani R, Naik M. Dynodroid: An input generation system for Android apps. In: Proc. of the FSE. 2013.
- [6] Mahmood R, Mirzaei N, Malek S. EvoDroid: Segmented evolutionary testing of Android apps. In: Proc. of the FSE. 2014.
- [7] Choudhary SR, Gorla A, Orso A. Automated test input generation for Android: Are we there yet? In: Proc. of the ASE. 2015. 429–440.
- [8] Google espresso. 2018. <https://developer.android.com/training/testing/espresso/>
- [9] Robotium. 2018. <https://github.com/RobotiumTech/robotium>
- [10] Google UI automator. 2018. <https://developer.android.com/training/testing/ui-automator>
- [11] Appium. 2018. <http://appium.io/>
- [12] Gomez L, Neamtiu I, Azim T, Millstein T. RERAN: Timing- and touch-sensitive record and replay for Android. In: Proc. of the ICSE. 2013.
- [13] Zhu MHY, *et al.* Mosaic: Cross-platform user-interaction record and replay for the fragmented Android ecosystem. In: Proc. of the ISPASS. 2015.
- [14] Chen JC, Xue YZ, Zhao C. Approach for GUI testing based on event handler function. Ruan Jian Xue Bao/Journal of Software, 2013,24(12):2830–2842 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4399.htm> [doi: 10.3724/SP.J.1001.2013.04399]
- [15] Amalfitano D, Fasolino AR, Tramontana P, De Carmine S, Memon AM. Using GUI ripping for automated testing of Android applications. In: Proc. of the ASE. 2012.
- [16] Choi W, Necula G, Sen K. Guided GUI testing of Android Apps with minimal restart and approximate learning. In: Proc. of the OOPSLA. 2013.
- [17] Firebase. 2018. <https://firebase.google.com/docs/test-lab/>
- [18] Testin. 2018. <http://testin.cn/>
- [19] Utest. 2018. <https://www.utest.com/>
- [20] Crowdsourced testing. 2018. [https://en.wikipedia.org/wiki/Crowdsourced\\_testing](https://en.wikipedia.org/wiki/Crowdsourced_testing)
- [21] Wang JJ, *et al.* Towards effectively test report classification to assist crowdsourced testing. In: Proc. of the ASE. 2016.
- [22] Smartphone test farm (STF). 2018. <https://openstf.io/>
- [23] Google accessibility. 2018. <https://www.google.com/accessibility/>
- [24] Didi chuxing. 2018. <https://www.didiglobal.com/>
- [25] JD app. 2018. <https://app.jd.com/>
- [26] JD app bug. 2018. <http://www.miui.com/thread-4545053-1-1.html>
- [27] Wei L, Liu Y, *et al.* Taming Android fragmentation: Characterizing and detecting compatibility issues for Android apps. In: Proc. of the ASE. 2016.
- [28] Choudhary SR, Prasad MR, Orso A. X-Pert: Accurate identification of cross-browser issues in Web applications. In: Proc. of the ICSE. 2013.
- [29] Fazzini M, *et al.* Automated cross-platform inconsistency detection for mobile apps. In: Proc. of the ASE. 2017.
- [30] Minitouch. 2018. <https://github.com/openstf/minitouch>

- [31] Hierarchy-Viewer. 2018. <https://developer.android.com/studio/profile/hierarchy-viewer>
- [32] UIAutomator. 2018. [http://developer.android.com/tools/testing/testing\\_ui.html](http://developer.android.com/tools/testing/testing_ui.html)
- [33] Kang Y, *et al.* DiagDroid: Android performance diagnosis via anatomizing asynchronous executions. In: Proc. of the SIGSOFT. 2016.
- [34] Kang Y, Zhou YF, Gao M, Sun YX, Lyu MR. Experience report: Detecting poor-responsive UI in Android applications. In: Proc. of the 2016 IEEE 27th Int'l Symp. on Software Reliability Engineering. 2016.
- [35] Tesseract-OCR. 2018. <https://github.com/tesseract-ocr/tesseract>
- [36] Blink-Diff. 2018. <https://github.com/yahoo/blink-diff>
- [37] Wu G, He M, *et al.* Detect cross-browser issues for JavaScript-based Web applications based on record/replay. In: Proc. of the ICSME. 2016.
- [38] Application/UI exerciser monkey. 2018. <https://developer.android.com/studio/test/monkey.html>
- [39] Amalfitano D, Carmine SD, Memon A, *et al.* Using GUI ripping for automated testing of Android applications. In: Proc. of the ASE. 2012.
- [40] Azim T, Neamtiu I. Targeted and depth-first exploration for systematic testing of Android apps. In: Proc. of the OOPSLA. 2013.
- [41] Yang W, Prasad MR, Xie T. A grey-box approach for automated GUI model generation of mobile applications. In: Proc. of the FASE. 2013.
- [42] Hu Y, Neamtiu I. Valera: An effective and efficient record-and- replay tool for Android. In: Proc. of the Int'l Workshop on Mobile Software Engineering and Systems. ACM, 2016. 285–286.
- [43] Lin YD, Chu ETH, *et al.* Improving accuracy of automated GUI testing for embedded systems. In: Proc. of the IEEE Software. 2014.
- [44] Lin YD, Rojas JF, *et al.* On the accuracy, efficiency, and reusability of automated test oracles for Android devices. In: Proc. of the IEEE TSE, Vol.10. 2014.
- [45] Han D, Zhang C, Fan X, Hindle A, Wong K, Stroulia E. Understanding Android fragmentation with topic analysis of vendor-specific bugs. In: Proc. of the WCRE. 2012.
- [46] Joorabchi ME, Li M, Mebah A. Detect inconsistencies in multi-platform mobile apps. In: Proc. of the ISSRE. 2015.
- [47] Boushehrinejadmoradi N, *et al.* Testing cross-platform mobile app development framework. In: Proc. of the ASE. 2015.

#### 附中文参考文献:

- [14] 陈军成, 薛云志, 赵琛. 一种基于事件处理函数的 GUI 测试方法. 软件学报, 2013, 24(12): 2830–2842. <http://www.jos.org.cn/1000-9825/4399.htm> [doi: 10.3724/SP.J.1001.2013.04399]



曹羽中(1984—),男,博士生,高级工程师,主要研究领域为软件工程,软件测试与维护.



魏峻(1970—),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为软件工程,分布式计算.



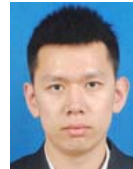
吴国全(1979—),男,博士,副研究员,CCF 专业会员,主要研究领域为软件工程,软件测试与维护,面向服务的计算.



黄涛(1965—),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为软件工程,分布式计算.



陈伟(1980—),男,博士,副研究员,CCF 专业会员,主要研究领域为软件演化与维护,服务计算,持续软件工程.



王溯(1997—),男,学士,主要研究领域为软件测试.