移动应用测试-GUI测试综述

摘要

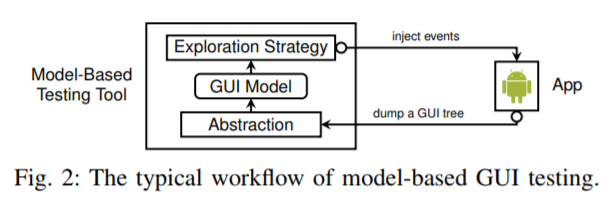
GUI是计算机软件与用户进行交互的主要方式。当前针对移动应用测试研究的一个关键方向就是移动应用的GUI测试。近年来, 移动应用GUI测试的问题吸引了大量的研究者的目光。本文将简要归纳GUI测试方面研究中的相关问题并重点归纳移动应用GUI测试方向研究的关键问题，对一般软件GUI测试简单介绍，对移动应用GUI测试研究的主要进展进行重点介绍和总结，并将二者比较，企图得到移动应用GUI测试的更好解决方案与方法。

关键词 移动应用 GUI测试 自动化测试

1引言

当今时代，GUI 软件和移动设备与人们紧密相连。自1981年Xerox公司推出首款图形用户界面电脑Xerox 8010 以来，通过GUI操纵软件已经成为人们使用软件的主要方式。除了普通应用，GUI软件也广泛应用于银行医疗航空等安全可靠关键（security-safe-critical）领域。GUI给人们带来了更加便捷的使用体验，但是却给软件开发带来了更大的工作量和复杂性。据统计，GUI软件用于GUI本身的代码量平均占到软件总代码量的48%，有时更高达总代码量的60%。因此，对GUI软件的测试成为了极具挑战性的任务。对于当前的移动设备，GUI软件覆盖率更是接近百分之百。

随着移动设备普及率迅速增长，移动应用的数量和规模逐年膨胀，对移动应用测试的需求也大规模增长。当下，移动应用无处不在，人们每天都在与移动应用打交道，且现在移动设备对用户友好度追求更加极致，手机等移动设备为几乎全部采用图形界面，而GUI正是与用户进行交互的主要方式，直接影响用户体验。移动应用GUI的测试由于其凸现出来的重要性，已日渐引起学术界和工业界的兴趣和重视。然而，目前关于移动应用GUI测试的研究还处于初级阶段：很多问题还没有解决，GUI软件测试依然需要较高人工成本，目前的技术还不能满足保证软件质量的实际需求。因此，如何提高移动应用GUI测试效率和准确性测试的可靠性和正确性也成为了一个重要问题。

当前，对于保证移动应用GUI测试的可靠性和正确性，已经有了不少探索。相关的论文工作也层出不穷，本文将针对近年来的移动应用GUI测试进行系统的介绍和总结。

【图1】经典的基于模型的Android应用GUI测试工作流

2 GUI测试的定义

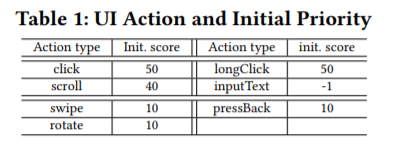
GUI测试，是图形用户界面的缩写，或用户可见的应用程序的一部分。GUI可能包括诸如菜单、按钮、文本框、图像等元素。当今的GUI测试不在局限在计算机的桌面，他可能是在所有主要的平台上可用的移动应用程序。测试人员必须通过跨平台测试来识别缺陷并确保应用程序满足所有要求。因此，GUI测试是指测试用户可见的应用程序的功能。GUI测试还会确认外观元素（如字体和图像）是否符合设计规范和要求。GUI测试发生在系统测试的级别。

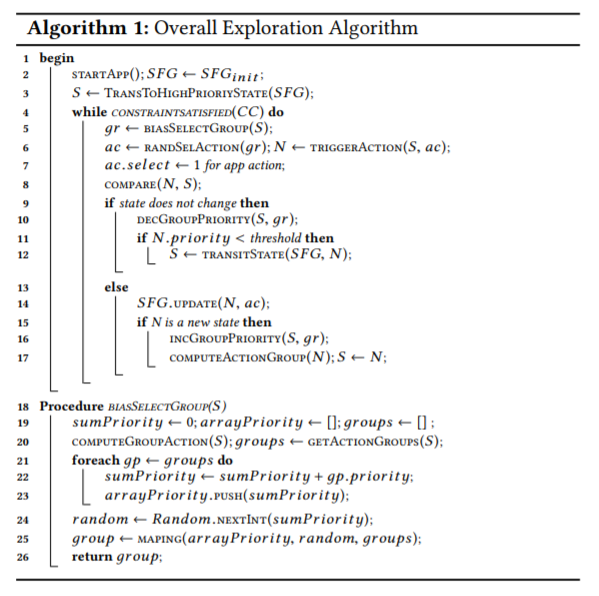
3 提升GUI测试可靠性和正确性的解决方案

最简单的方法是模糊测试，它产生随机事件来探索自动测试的行为并检测故障。然而，模糊测试的一个主要局限性是，由于随机性的本质，会产生大量的冗余事件。此外，跟踪和调试可疑路径到检测到的故障比系统方法更困难。因此我们企图寻求其他方面的解决思路。

3.1.1 基于反馈的探测策略【1】

这是一种有效的基于模型的Android应用程序GUI测试技术。为了避免局部和重复的探索,该团队的方法将等效部件打包成组，并且设计一种新颖的，基于反馈的勘探策略,这将通过基于那些已经出发的动作的执行结果，动态调整动作的优先级,并倾向于选择那些可以到达新状态的应用程序的行为。该团队研发了CrawlDroid的工具实现了该技术，并进行了实验。



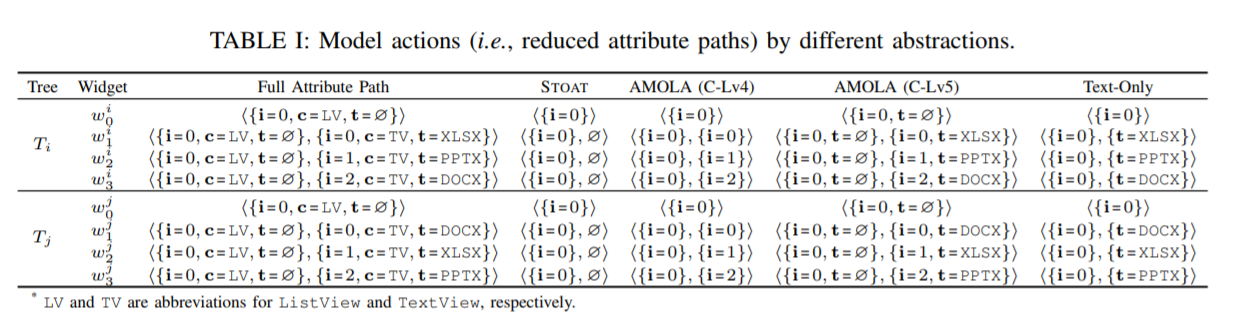
 基于模型的测试将根据预定义的遍历策略(例如深度优先搜索、广度优先搜索)对每个确定可执行小部件执行操作。该方法设计了一种新的基于反馈的探索策略，它将在一种状态下的小部件进行分组，并分配一个优先级值给每一个该组支持的动作。根据一个动作的执行结果，探索策略会调整动作的优先级(属于一个组)。通过动态地分组小部件，调整动作的优先级。

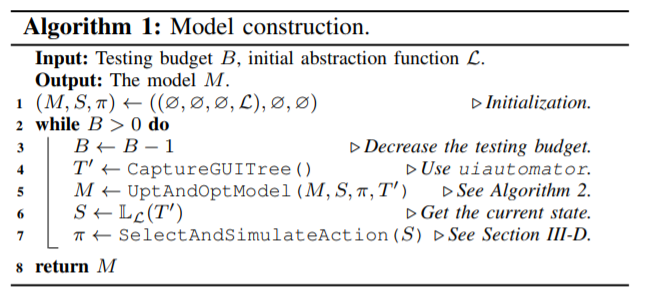
3.1.2 启发

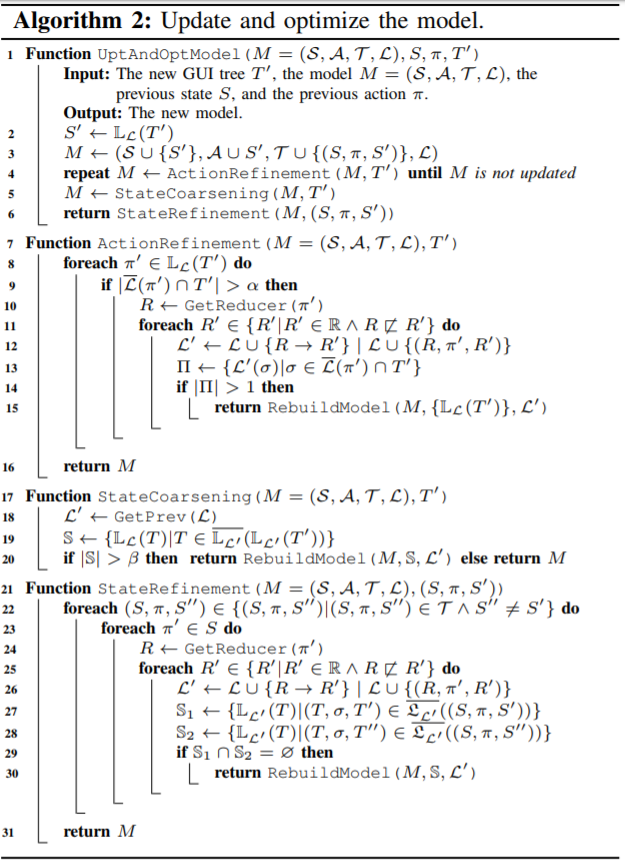
现有的基于模型的测试技术的一个主要限制是，它们花费大量的时间来探索AUTs的一些局部，而没有获得覆盖改进。这种方法可以避免局部探测，并有更多的机会达到AUTs的新状态。一定程度上缓解了基于模型的移动应用GUI的缺陷。

3.2.1模型抽象和细化【2】

该方法不同于其他现有的基于模型的GUI静态测试方法，例如，模型在测试的过程中不会演化出他的抽象，因此常常是不准确的。该方法在测试运行的过程中，利用运行时信息来动态地优化模型，与现有方法相比，这种模型演化能力显著提高了模型的精度，从而显著提高了测试的有效性。

移动应用程序测试需要大量的人力投入。例如，测试人员编写代码来模拟GUI操作(例如单击一个按钮)来驱动Android应用程序的执行。这个过程不仅耗时，而且容易出错。此外，当GUI发生变化时，测试人员必须对他们现有的测试脚本进行重要的修改。模型为GUI测试提供了三种类型的好处。一、测试工具可以使用特定的遍历模型，系统地生成动作序列，然后回放动作序列，对app进行测试。其次，基于模型的测试工具生成由高级模型操作而不是低级事件组成的输入序列，这样可以方便回放。第三，可以将适当的抽象应用于模型，从而帮助减轻GUI操作的爆炸式增长。通过抽象，许多具有相同行为的GUI操作可以映射到相同的模型操作。由于这些GUI操作的行为是相同的，所以测试工具不需要测试每一个操作，而是可以在执行模型操作时，在其中选择一个有代表性的GUI操作。对状态进行抽象。将每个GUI操作映射到一个模型操作是状态抽象中最关键的步骤。因为具有相同模型动作集合的状态可以被合并，状态通常由它的模型动作集合来标识。

基于有效的动态模型抽象思想，该研究团队提出了一种新的、实用的基于模型的Android应用程序自动GUI测试技术——APE。一开始。用默认的抽象去初始化测试，这个初始化的抽象可能是无效的，但是通过观察运行时的信息，APE通过寻找更合适的抽象来细化模型。具体来说，APE用一个决策树表示动态抽象，并使用测试期间获得的反馈动态地调整决策树。该方法和其他方法最主要的不同，就是有动态进化GUI模型的能力。

3.2.2启发

通过动态调整决策树，缓解了一般基于模型的GUI测试的问题，即：工程量大，精确度不高，可移植性差等问题。

3.3.1 基于行为模型的Android应用随机测试【4】

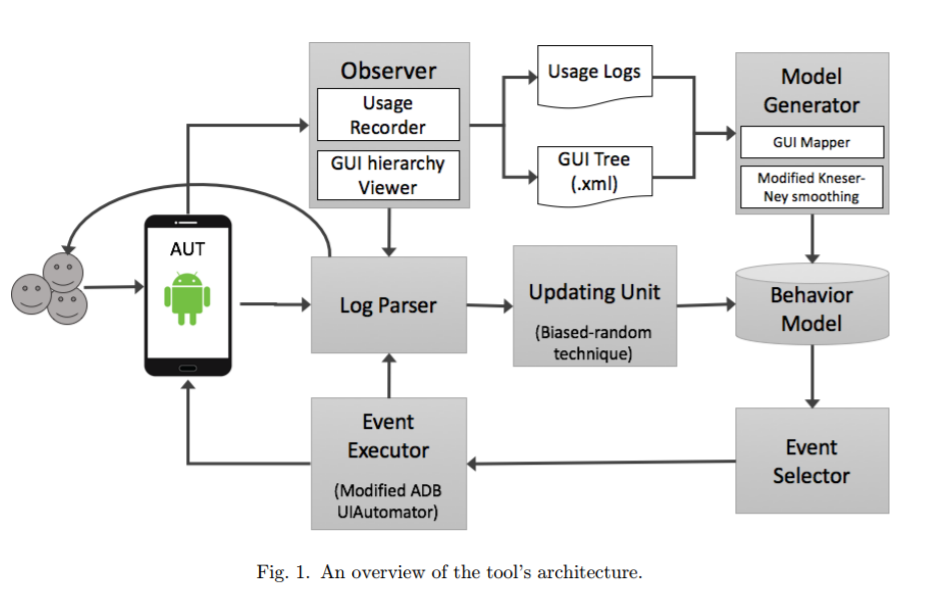
主要解决思路是，创建一个可以被简单实用的，有着很小依赖和更高效率的自动化工具。依赖基于行为的GUI自动化测试方法，但是减少预先需要的设置初始化，并仍然能够达到相同或更好的覆盖率。因此，通过以下方法实现：

(1)通过在运行时映射GUI树和行为模型来避免静态分析

(2)动态更新模型，增加用户使用应用程序时很少或从不发生的事件发生的概率。

(3)应用统计模型创建行为模型。

虽然行为模型的方法并不新颖，但是此方法的独特之处在于，我们随着测试的进展不断地更新行为模型。最初的行为模型是使用统计建模从使用日志创建的，但是同时也考虑了没有出现在使用日志中的事件。这是通过将行为模型中的事件与GUI树中的GUI组件进行映射来实现的。事件发生的概率随着测试的进展而调整。

下图，是整个系统的架构图。

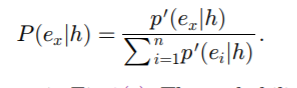
方法可以分为以下六个步骤:(1)事先收集使用日志，从AUT中提取GUI树(2)应用统计模型创建行为模型，然后与GUI树进行映射 (3)从行为模型中随机选择一个事件 (4)将选择的事件发射到AUT (5)按频率调整概率，必要时更新模型 (6)重复第(2)至第(5)项，直至达到规定的时间限制或预先决定的次数。

在用GUI树映射行为模型的过程中，该方法并没有将事件和GUI组件放在一起。行为模型中的每个事件都被映射到GUI树中的一个GUI组件，以唯一的资源id作为键。

在选择阶段，会选择随机的事件，而不是选择发生概率高的事件。

在执行阶段，事件执行器执行下一个被选择的事件，工具可移植性类似点击，滑动等时间。

在更新阶段，执行阶段后，模型将将采用有偏随机技术去调整选中或执行的事件概率。他从前面选中并执行的事件中减少因素：

在减少一些可能性后，要重新计算来维持一个合理的概率分布：

最后进行评估。

最终我们会发现，随意测试和认真测试可能会得到不同的结果。由于好处并不明显，我们没有在录制期间提供特定的限制或目标。这会对此方法的有效性造成威胁。

3.4.1

有证据表明，移动应用程序没有像桌面应用程序那样经过彻底的测试。特别是GUI测试通常是有限的。与基于web的应用程序一样，移动应用程序也受到GUI测试脆弱性的影响。GUI测试在不改变应用程序功能的情况下，会由于GUI中的微小修改而失败。【5】

1. Yuzhong Cao ，Guoquan Wu ，Wei Chen ，Jun Wei “CrawlDroid: Effective Model-based GUI Testing of Android Apps” Proceedings of the Tenth Asia-Pacific Symposium on Internetware
2. Tianxiao Gu，Chengnian Sun ，Xiaoxing Ma ，Chun Cao ，Chang Xu ，Yuan Yao Qirun Zhang ，Jian Lu ，Zhendong Su “Practical GUI Testing of Android Applications via Model Abstraction and Refinement” 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)
3. Hendrik Bünder ，Herbert Kuchen “A Model-Driven Approach for Behavior-Driven GUI Testing” ，Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing
4. Woramet Muangsiri and Shingo Takada† “Random GUI Testing of Android Application Using Behavioral Model” [International Journal of Software Engineering and Knowledge Engineering](https://www.worldscientific.com/worldscinet/ijseke)[Vol. 27, No. 09n10, pp. 1603-1612 (2017)](https://www.worldscientific.com/toc/ijseke/27/09n10)Research Notes
5. Riccardo Coppola, Maurizio Morisio and Marco Torchiano “Scripted GUI Testing of Android Apps: A Study on Diffusion, Evolution and Fragility”  Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering ，ACM