

# 基于模型的移动应用兼容性测试方法研究

成静<sup>1,2</sup>, 朱怡安<sup>1</sup>, 张涛<sup>2</sup>, 周文强<sup>2</sup>, 李坤<sup>2</sup>, 王海鹏<sup>1</sup>

(1.西北工业大学 计算机学院, 陕西 西安 710072; 2.西北工业大学 软件与微电子学院, 陕西 西安 710072)

**摘要:**在分析研究移动应用测试验证技术机理的基础上,提出一种基于测试价值权重语义树模型的移动应用兼容性测试方法。该方法的主要特点是能够生成所有移动应用兼容性测试环境,并可根据其测试价值进行排序和选择,从而能提高测试效率,降低测试成本,减少测试时间。

**关键词:**测试模型与分析;兼容性测试;移动应用测试;移动测试环境

**中图分类号:**TP311

**文献标志码:**A

**文章编号:**1000-2758(2015)02-0337-05

近年来,移动应用在通信、娱乐、新闻、商业、金融等各个行业均取得了快速发展。预测到2014年,全球移动应用市场规模将高达3 300亿美元,而移动应用测试技术已成为亟待研究的问题之一。由于移动设备特征和环境的差异性,需要进行充分的兼容性测试。然而移动设备多达上万种,如何选择合适的移动设备和应用环境,是当前兼容性测试迫切需要解决的难题之一<sup>[1]</sup>。

虽然,已有一些论文针对移动应用的测试方法进行研究,但对与兼容性测试相关的研究仍较少。文献[2]研究了针对Android碎片的兼容性测试方法。文献[3]提出一种综合考虑设备特征覆盖率和设备市场份额的移动应用兼容性测试策略。目前,移动应用兼容性测试仍主要依靠测试人员经验。为此,本文提出了一种基于模型的移动应用兼容性测试方法,并进行了实例验证分析。

## 1 语义树模型

语义树是一种多层次树形模型,可以用于移动应用兼容性测试建模<sup>[4]</sup>。在语义树中,每个节点用来描述一种测试特征,节点之间的链接则可用于描述测试特征之间的语义关系。于是,可以将一个兼容性测试语义树定义为三元变量 $(N, E, R)$ 。其中

1)  $N$  是一组语义树节点,包括叶子节点、中间节点和根节点,节点用于表示移动应用环境特征。

2)  $E$  是一组节点之间的连接,每个连接包括一个父节点和它的一个子节点。每个连接描述了父节点和子节点之间的语义关系。

3)  $R$  是一组关系。 $R$  有一个语义标签,描述父节点与子节点之间的语义关系。这里,定义了4个语义标签: EOR, AND, SELECT-1, and SELECT- $M$ 。图1给出语义关系示例,表1详细描述语义关系。

为了支持移动应用兼容性测试分析,本文在语义树模型基础上,提出了语义生成树概念。下面给出其定义。在图1中,给出一个语义树,以及其生成树的例子。

语义生成树是指根据语义关系,由语义树派生出的子树,上述4种语义关系所对应的派生规则分别是:

1) 对于每个具有 AND 关系的父节点,必须包含它的所有子节点以及它们的连接。

2) 对于每个具有 EOR 关系的父节点,必须且只能包含其中一个子节点以及它的连接。

3) 对于每个具有 SELECT-1 关系的父节点,必须且只能包括其中一个子节点以及它们相应的连接。

4) 对于每个具有 SELECT- $M$  关系的父节点,

必须包括它的  $M$  个子节点以及它们相应的连接。

表 1 语义关系说明表

关系	语义关系描述( $P$ 是一个父节点, $C_i$ 是它的子节点)
$EOR(P, \langle C_1, C_2 \rangle)$	EOR 关系表示 $P$ 节点有 2 个子节点, 这两个子节点二者只能选其一。
$AND(P, \langle C_1, C_2, \dots, C_n \rangle)$	AND 关系表示 $P$ 节点必须包含它所有子节点 $C_1, C_2, \dots, C_n$ 的移动应用环境特征。
$SELECT-1(P, \langle C_1, C_2, \dots, C_n \rangle)$	SELECT-1 关系表示节点 $P$ 必须从可选择的子节点 $C_1, C_2, \dots, C_n$ 中选择 1 个移动应用环境特征。
$SELECT-M(P, \langle C_1, C_2, \dots, C_n \rangle)$	SELECT- $M$ 关系表示节点 $P$ 必须从可选择的子节点 $C_1, C_2, \dots, C_n$ 中选择 $M$ 个移动应用环境特征。

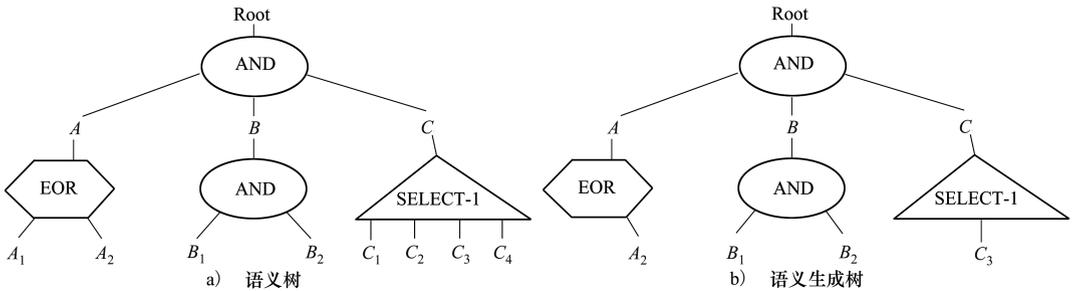


图 1 语义树及其生成树示例

## 2 测试价值语义树模型

基于语义树模型的兼容性测试方法, 通过其生成树集合能够获得所有的兼容性测试环境配置。但所生成的兼容性测试配置环境, 数量巨大, 而其中一些测试环境对移动应用兼容性的测试价值可能很低。因此, 需要研究优化策略和方法, 以减少测试环境数量, 提高测试效率。

### 2.1 测试价值权重语义树模型

影响移动应用兼容性测试的环境特征很多, 如移动应用平台、移动设备硬件特征、本地 API 接口等, 每个移动特征可能包含多个特征值。但是, 由于各个移动应用的功能、架构和运行环境不尽相同, 各个移动特征和特征值对其兼容性的影响, 即其引发移动应用故障的概率也可能不同。例如, 对 Skype 应用, 就其摄像头分辨率特征而言, 分辨率越高, 需

要传输的数据也越多, 越容易引起故障。因此, 其特征值为 200 万像素相对于 30 万像素, 会具有更高的测试价值。

因此, 在为移动应用兼容性测试环境建模时, 需要综合考虑特征和特征值对兼容性测试的价值。为此, 本文在移动应用兼容性语义树模型中, 增加了价值权重因子。图 3 为移动应用 Skype 兼容性测试权重语义树模型, 其中根据该特征对移动应用兼容性测试的价值和重要性, 即该特征引发移动应用安装运行的兼容性的故障概率与危害性, 给予其赋予价值权重因子  $w$ 。

例如, 在 Skype 应用测试价值权重语义树模型中, 其网络连接下的, wifi 的测试价值为 0.5, 而网络连接的测试价值设定为 0.9。这是因为网络连接比 wifi 的带宽更低, 网络更不稳定, 可能出现的兼容性问题更多。这里, 我们对所有的语义树模型节点设定其测试价值权重因子, 在表 2 中给出其约束规则:

表 2 权重因子约束规则表

约束规则	约束规则说明
$w \leq 1$	所有节点测试价值权重因子小于等于 1,其中根节点权重因子为 1。
$\sum_{i=1}^n w_i = 1$	语义关系为 And 的同一父节点的所有子节点的权重因子之和为 1。这里 $w_i$ 是其第 $i$ 个子节点的测试价值权重值。
$w_i \leq 1$	语义关系为 OR, SELECT-M, SELECT-1 的同一父节点的所有子节点权重因子均小于等于 1。

在语义树模型中,对每一个非叶子节点的所有子节点,按照其价值权重排序,价值权重大的排在左边,重构移动应用的语义树模型为价值权重语义树模型,该价值权重语义树模型是一个有序树。例如

对于 Skype 应用,根据其功能与兼容性测试经验,为其各个非根节点定义其价值权重,其权重语义树模型如图 3 所示,其价值权重语义生成树实例如图 4 所示。

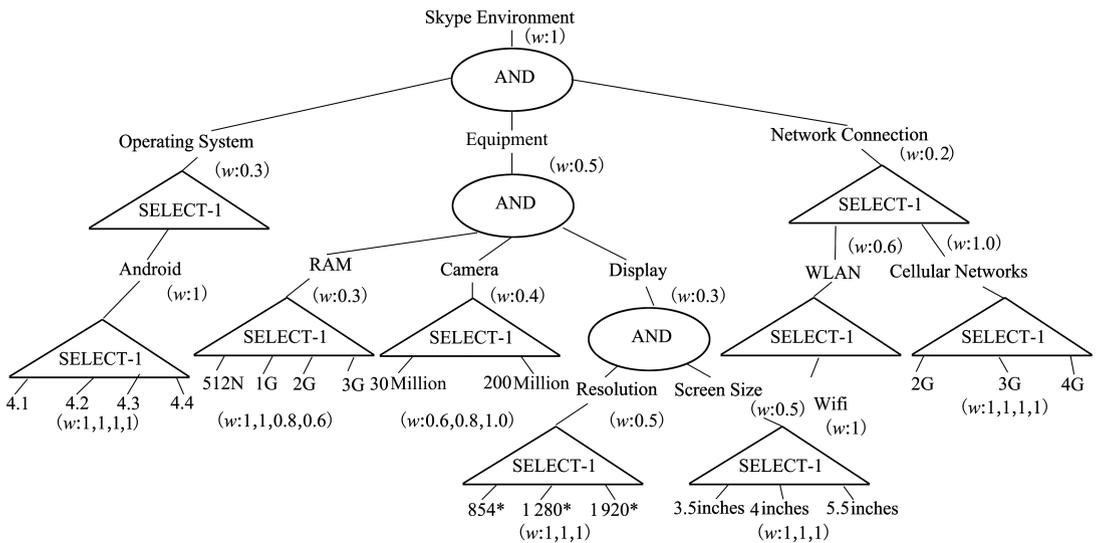


图 2 Skype 兼容性测试价值权重语义树模型

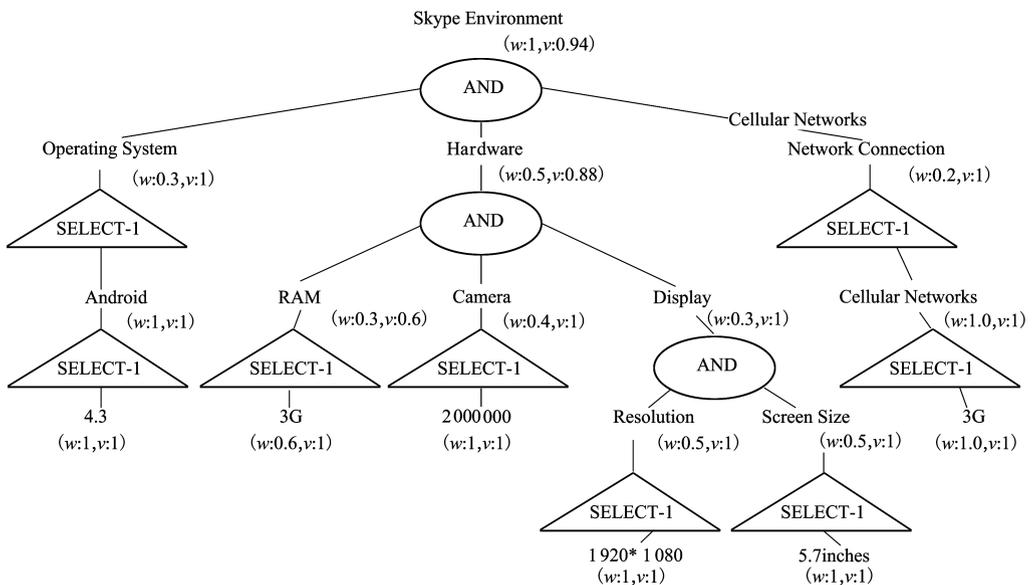


图 3 Skype 兼容性测试价值权重语义生成树实例

### 2.2 语义生成树测试价值计算

基于价值权重语义树模型,可以获得其所有的价值权重语义生成树。对每个价值权重语义生成树,可以递归计算其最终的测试价值。

首先,根据子节点测试价值计算其父节点测试价值。这里,假设父节点为  $F$ ,并设它共有  $N$  个子节点,分别为  $S_1, S_2, \dots, S_n$ ;每个子节点的测试价值权重分别为  $w_1, w_2, \dots, w_n$ ;其测试价值分别为  $v_1, v_2, \dots, v_n$ 。则父节点  $F$  的测试价值  $V$  可由下式算出:

$$V = \min\left\{1, \sum_{i=1}^n (w_i * v_i)\right\} \quad (1)$$

即父节点的测试价值为其所有子节点测试价值与测试权重之积的和,最大为 1。

对于价值权重语义生成树的叶子节点,其测试价值的缺省值为 1。

这样通过递归计算,最终可以得到测试权重语义生成树根节点的测试价值,即是该语义生成树的测试价值。例如,在图 4 中,对节点 RAM 的测试价值,根据其子节点 3G 的测试价值 1 和价值权重 0.6 计算为 0.6;对节点 Hardware 的测试价值,则根据其子节点 RAM、Camera 和 Display 的测试价值和价值

权重计算为 0.88;最终计算的生成树价值权重为 0.94。

通过上述方法,可以计算所有语义生成树的测试价值,并按照测试价值对其排序。在兼容性测试时,可以根据测试资源,优先测试价值高的测试环境,而略去测试价值低的测试环境,以期节约测试资源,降低测试成本,提高测试效率。

### 3 实例验证研究

为了验证本文方法的有效性,对 Skype、Yelp 2 个移动应用进行了基于 Android 平台的兼容性测试。首先,分别对这 2 个移动应用建立其测试价值权重语义树模型,然后分别计算各生成树的测试价值,并进行排序。再选择测试价值 > 0.7 的测试环境,构成价值权重语义树兼容性测试环境集。同时,根据经验为两个应用分别选择相同数量的兼容性测试环境,构成相对应的经验测试环境集。最后在两组测试环境上,分别执行同一组的测试用例,表 3 为最终测试结果。

表 3 实例测试结果

移动应用	语义生成树数量	测试环境数量	非兼容性故障	兼容性故障	
				模型测试环境集	经验测试环境集
Skype	1728	84	3	8	3
Yelp	1280	69	5	9	2

根据语义树模型,能够获得所有兼容性测试环境,而通过测试价值权重,能够优先测试高价值的测试环境,以提高测试效率,降低测试成本。由表 3 易见,相对于经验测试方法,本文测试方法能够发现更多移动应用兼容性缺陷,使测试工作更为有效。

### 4 结 论

目前,我国对移动应用兼容性测试的研究仍处于初期阶段,系统有效的方法尚不多见。本文构建了一种基于测试价值语义树模型的移动应用兼容性测试方法。该方法首先通过生成树算法获得所有兼容性测试环境。然后基于测试价值进行排序和过滤,能够有效提高测试效率,降低测试成本。下一步将设计开发一个自动、可视化的移动应用兼容性测试工具,并开展更多的实例验证研究。

## 参考文献:

- [1] Huang Junfei. AppACTS: Mobile App Automated Compatibility Testing Service[C]//Proceeding of 2014 2nd IEEE International conference on Mobile Cloud Computing, Services, and Engineering, Oxford, UK, 2014: 85-90
- [2] Han Dan, Zhang Chenlei, Fa Xiaochao, et al. Understanding Android Fragmentation with Topic Analysis of Vendor-Specific Bugs[C]//Proceedings of the 19th Working Conference on Reverse Engineering, Kingston, Canada, 2012: 83-92
- [3] Sergiy Vilkomir, Brandi Amstutz. Using Combinatorial Approaches for Testing Mobile Applications[C]//. Proceeding of 2014 IEEE International Conference on Software Testing, Verification, and Validation Workshops, Cleveland, OH, 2014: 78-83
- [4] Tao Chuanqi, Gao Jerry. Modeling Mobile Application Test Platform and Environment: Testing Criteria and Complexity Analysis. Proceeding of 2014 Workshop on Joining AcadeMiA and Industry Contributions to Test Automation and Model-Based Testing, San Jose, USA, 2014: 28-33

## A Model Based Compatibility Testing Method for Mobile Application

Cheng Jing<sup>1,2</sup>, Zhu Yi'an<sup>1</sup>, Zhang Tao<sup>2</sup>, Zhou Wenqiang<sup>2</sup>, Li Kun<sup>2</sup>, Wang Haipeng<sup>1</sup>

(1. Department of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China)  
(2. Department of Software Engineering, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** With the rapid development of mobile application's market, how to efficiently test and verify the quality of mobile applications has become one research focus in current academia and industry. In particular, due to the types of mobile devices, technology updates faster; how to test compatibility for mobile applications is an urgent problem. We propose a new mobile application testing method based on testing-value-weight semantic tree model. The method can drive all testing environments of mobile applications and then rank and select appropriate environments with the testing values. The experiments have proved the feasibility of the proposed method, which can reduce the amount of compatibility testing environmental configuration, thus improving the efficiency of testing and reducing the testing cost and time.

**Key words:** calculations, cost reduction, efficiency, experiments, mathematical models, semantics, software testing; compatibility test, mobile app testing, mobile testing, testing model