

基于 Petri 网的图形用户界面测试*

林 涛, 高建华

(上海师范大学 计算机科学与工程系, 上海 200234)

摘 要: 在不确定上下文相关的环境中, 图形用户界面测试主要依靠随机测试以及测试人员的从业经验, 其有效性低。提高图形用户界面测试的效率是一个未解难题。因此, 通过引入离散并行系统的 Petri 网理论, 定义了图形用户界面的事件、事件序列和事件分解等概念, 将可达性、有界性、活性以及强连通性等 Petri 网的重要性质延伸至此领域, 以提高图形用户界面测试的覆盖面和效率。并且试图解决不可达、不强连通、死锁、无界、不符合初始模型和跳转错误等六类图形用户界面缺陷问题。实验证明基于 Petri 网的图形用户界面测试在事件覆盖数、代码行覆盖数以及缺陷发现数等方面均优于其他方法。

关键词: 软件缺陷; 测试用例生成; 黑盒测试; 事件序列; 离散并行系统

中图分类号: TP311.53 **文献标志码:** A **文章编号:** 1001-3695(2016)03-0768-05

doi: 10.3969/j.issn.1001-3695.2016.03.029

Graphic user interface testing based on Petri net

Lin Tao, Gao Jianhua

(Dept. of Computer Science & Technology, Shanghai Normal University, Shanghai 200234, China)

Abstract: In environment of undetermined and context-sensitive, graphic user interface testing mainly depends on random testing and testers' professional experience, the effectiveness being low. It is an unresolved puzzle for graphic user interface testing. Therefore, it is brought in the Petri net theory in the discrete and parallel system, defining concepts of event, events sequence, and events decomposition in graphic user interface. The paper introduced some significant properties of Petri net, such as reachability, roundedness, liveness and strong connectedness to this field, so as to improve the coverage and efficiency of graphic user interface testing. In addition, an attempt to solve six categories bugs in graphic user interface, such as non-reachability, not strong connected, dead-lock, unbounded, not suitable to the original model and error jumping is conducted. The experiment proves that graphic user interface testing based on Petri net is more effective than other test methods in coverage of events, code lines as well as the number of fault detection.

Key words: software bug; test case generation; black-box test; event sequences; discrete parallel system

随着计算机软件复杂度的提高, 图形用户界面 (graphic user interface, GUI) 测试面临愈来愈多的难题。主要有以下困境: a) 图形用户界面不确定的运行环境^[1], 相较以前, 现今界面更加错综复杂, 图形用户界面事件成倍增加; b) 需要耗用大量时间^[2], 过多依赖随机测试以及测试人员的从业经验; c) 图形用户界面测试难以应对用户满意度^[3], 尽管软件公司投入大量人力、物力, 努力增加测试覆盖率, 但是依然不能满足需求规约。

尽管对于测试用例的通用生成方法已经成熟^[4], 但是应对图形用户界面测试的理论与测试用例生成方法依然是一个难题。很多学者提出了一些有待完善的理念, 如 Nguyen 等人^[5]提出了 Observe-Model-Exercise * (OME, “*”表示该方法的迭代本质) 方法以解决图形用户界面难以自动化测试的问题。与该方法类似, 一种基于模型的黑盒测试方法对图形界面事件序列的测试也具有一定的有效性^[6]。Mayo 等人^[7]认为 Combinatorial-based 的评价方式可以有效用于图形用户界面的测试。Nakajima 等人^[8]设计了一种用于多个图形用户界面窗

口的测试算法。Rauf 等人^[9]研究了 GUI 测试的覆盖率, 并且使用了事件流图技术。然而文献中都未能提出系统化的基于严格数学理论的图形用户界面测试方法, 提高图形用户界面测试有效性的关键在于研究上下文相关的自动化或半自动化的动态测试。

本文提出一种基于 Petri 网的图形用户界面测试方法的目的在于解决前述三点问题。本文主要贡献为以下三点:

- a) 提出并总结在图形用户界面中应用的 Petri 网性质;
- b) 应对六类图形用户界面缺陷问题, 提出自动化测试的新方法;
- c) 通过测试用例数量等指标, 验证基于 Petri 网的图形用户界面测试的效率优于传统的随机测试和 OME 方法。

本文将 Petri 网理论比较完整地应用于图形用户界面测试。

1 图形用户界面测试中 Petri 网相关知识

Petri 网主要用于离散并行系统和软件工作流等的建模。

收稿日期: 2014-12-13; **修回日期:** 2015-02-03 **基金项目:** 国家自然科学基金资助项目(61073163); 上海市企业自主创新专项资金项目(沪 CXY-2013-88); 上海师范大学研究生重点项目(SNU14001)

作者简介: 林涛(1988-), 男, 甘肃兰州人, 硕士研究生, 主要研究方向为软件测试、离散系统(lintaoo@hotmail.com); 高建华(1963-), 男, 教授, 博导, 博士, 主要研究方向为软件可靠性理论与设计、软件开发环境与开发技术、数据安全与计算机安全、网络测试、LSI/VLSI 测试等。

本章主要研究 Petri 网理论迁移到图形用户界面测试的有关概念。

1.1 相关定义

在此,定义 Petri 网与图形用户界面的相关定义。

1.1.1 Petri 网定义

Petri 网为一个五元组^[10], $PN = (P, T, I, O, M_i)$ 。其中:

$P = \{p_1, p_2, \dots, p_m\}$ 为库所(place)的集合。

$T = \{t_1, t_2, \dots, t_n\}$ 为变迁(transition)的集合。

$I: P \times T \rightarrow N$ 是输入函数,它定义了从 P 到 T 的有向弧的重数,这里 N 为非负整数集。

$O: T \times P \rightarrow N$ 是输出函数,它定义了从 T 到 P 的有向弧的重数。

$M_i: PN$ 的标志,它为列向量,表示第 i 个库所中 token 或标志数目。

t 是 t 的前集^[11],对于 $t \in T$,如果 $\forall s \in S: s \in {}^{\cdot}t \rightarrow M(s) \geq 1$,则称变迁 t 在标志 M 有发生权,记为 $M[t >]$ 。

1.1.2 事件、事件序列

用户对图形用户界面的一个操作,为一个事件,包括但不限于点击按钮。

一个或者多个事件按照先后次序,组合成为一个有效的任务活动,就称之为事件序列。例如通过多次点击不同选项框,最后提交之前所有选择,就是一个事件序列。

1.1.3 事件分解

考虑上下文因素,即使在图形用户界面中看似是同一事件,事实上,由于之前事件序列(前驱事件)不同会发生不同激发,所以需要将这一事件分解为不同子事件。

1.2 应用于图形用户界面测试的 Petri 网性质

为了将 Petri 网有效建模于图形用户界面测试,需要将 Petri 网的特定性质与图形用户界面对应,着重于以下性质:可达性、有界性、活性、前驱事件、强连通性及孤岛。其中,图形用户界面测试中的可达性、强连通性不同于 Petri 网中的相关性质。

1.2.1 可达性

从 Petri 网的初始标志 M_0 开始激发一个变迁序列并且产生一个标志 M_r ,则称 M_r 是从 M_0 可达的^[12]。更进一步,可达标志集定义如下:

对于 Petri 网 $PN = (P, T, I, O, M)$,若存在 $t \in T$,使 $M[t > M']$,则称 M 到 M' 是直接可达的。若存在变迁序列 t_1, t_2, \dots, t_k 和标志序列 M_1, M_2, \dots, M_k ,使得

$$M[t_1 > M_1][t_2 > M_2] \dots [t_k > M_k]$$

则 M 到 M_k 是可达的。 M 可达的所有标志的集合记为 $R(M)$ 。

换言之,可达标志集是 Petri 网任何可能发生序列所能进入的全部状态的集合。在本研究中,可达标志集指软件的图形用户界面满足于需求规约的事件序列集合。

1.2.2 有界性

Petri 网模型的有界性是指网中任何位置的 token 数是有界的,给定 PN 及其可达集 $R(M_0)$,对于库所 $p \in P$,若 $\forall M \in R(M_0): M(p) \leq k$,则称 p 是 k 有界的。若 Petri 网的所有库所都是 k 有界的,则 Petri 网是有界的。当 $k = 1$ 时,称库所或 Petri 网是安全的。

对于图形用户界面测试,每个事件都只能由有限个事件触发。在本文中只研究安全的图形用户界面测试,即任何违反多

余一个事件触发生成的事件都是软件缺陷。

1.2.3 活性

对于 $t \in T, \forall M \in R(M_0)$,若存在特定变迁序列(series),该变迁序列的激发使得此变迁 t 可触发,则称该变迁是活的^[13]。更进一步,如果 Petri 网的所有变迁都是有活性的,则该 Petri 网是具有活性的;否则称为死锁。

对于图形用户界面测试,不存在一个事件序列,使得如果进入该事件序列,则系统无法跳出,即不能再进入其他事件序列。

1.2.4 强连通性

若一个 Petri 网具有强连通性,当且仅当对于 Petri 网中的每一对节点(即库所和变迁) x 和 y ,都存在一条从 x 到 y 的路径;否则,就称存在孤岛。

对于图形用户界面测试,不存在一个事件是无法由任何事件序列到达,则称此图形用户界面满足强连通性。若存在软件缺陷,则缺陷事件就是孤岛。

在图形用户界面测试中不可达与不强连通并非同一概念。事实上,强连通是可达的必要非充分条件。

2 图形用户界面缺陷问题与实例分析

首先举一个实例,然后引出本文研究聚焦的五种图形用户界面缺陷问题。

2.1 订书模型图形用户界面分析

使用一个简化的订书系统图形用户界面,共有三个界面,依次为 homepage、confirm、congratulation,如图 1 所示。

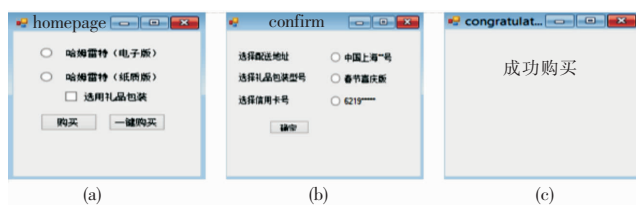


图1 实例图形用户界面

顾客购买《哈姆雷特》,可以选择电子版或者纸质版(点击选择框)。若选择购买纸质版,可以更进一步选择是否使用礼品包装。之后,顾客确定购买,有两种方法,即购买和一键购买。区别如下:若顾客在 homepage 界面选择一键购买,则系统自动调用该顾客的预存信息,进入成功购买界面。若是点击“购买”按钮,则进入下一个界面。在这个界面(在不影响理论说明的情况下,本界面的选择配送地址、选择礼品包装、选择信用卡号都仅有一个选项),若之前顾客购买纸质版,可以选择配送地址和信用卡号;若顾客之前还选择了使用礼品包装,那么顾客在这个界面可以挑选各种不同版式的礼品包装,否则该选项禁用。若顾客购买的是电子版,则选择配送地址和选择礼品包装,这两个选项都是禁用的。最后确定,进入购物成功界面。流程如图 2 所示。

用 Petri 网对该订书系统图形用户界面进行建模,如图 3 所示。

虽然在图 1 中的 homepage 界面“购买”和“一键购买”按钮各仅有一个,但是考虑到上下文的影响,不同选择指向不同的 confirm 界面,意味着前一界面选项的不同,导致后一界面部分按钮的禁用。因此,在图 5 中,将“购买”按钮分为三种库所进行描绘,即 P_3 、 P_6 以及 P_{11} 购买;一键购买分为 P_{17} 、 P_{18} 和 P_{19} 一键购买。

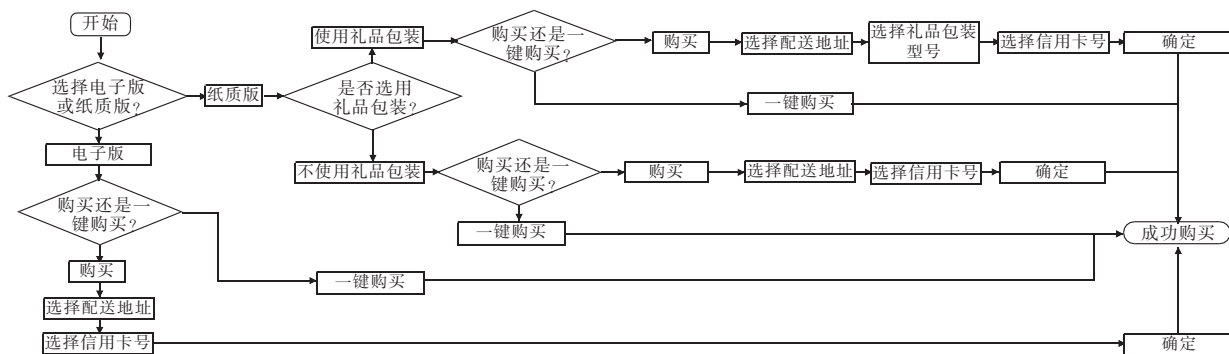


图2 实例的流程图

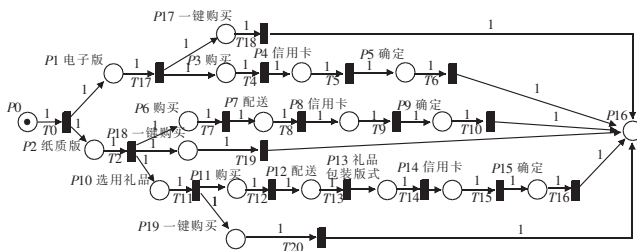


图3 初始实例的 Petri 网建模

事实上,设置“一键购买”事件,目的在于研究 Petri 网能否检测图形用户界面的正确跳转问题。因而在下文中除非特别说明,该研究简化图1的 homepage 为 simple homepage,如图4所示。

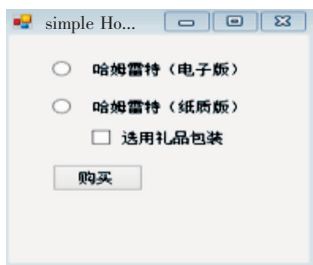


图4 简单首页

图3初始实例的 Petri 网建模也可简化为图5。

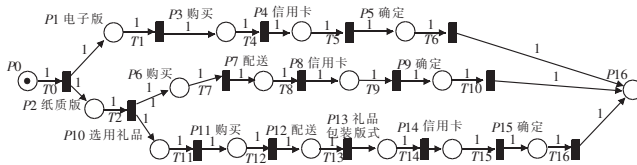


图5 简化实例的 Petri 网建模

对该应用系统图形用户界面的事件序列构造可达树,该事件序列是满足可达的、有界的,且是强连通的。

2.2 六类图形用户界面缺陷问题

在图形用户界面测试的分类领域,文献中提出了不同的分类角度。Issa 等人^[14]研究 visual testing(VT),将 GUI 测试分为两大类,即单一对象测试和多个对象测试。

在本文中,订书应用系统模型设计完成后,如果存在缺陷,特别是在图形用户界面发生事件序列的错误中,根据1.2节,将关注以下六类问题,并在之后的实验部分给出具体的设计与结果。

2.2.1 不可达问题

实际图形用户界面某些事件不可达。比如 homepage 中的“选用礼品包装”选框存在缺陷,无法使用,那么,confirm 界面

中的“选择礼品包装型号”事件将不可达。此实际图形用户界面的 Petri 网描绘如图6所示。

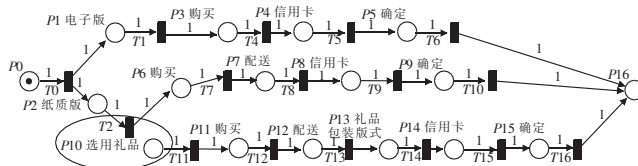


图6 不可达事件举例

2.2.2 不强连通问题

实际图形用户界面不强连通,存在孤岛。如果图1中 homepage 的哈姆雷特(电子版)选项存在缺陷,并且 Petri 网描绘如图7所示,则“哈姆雷特(电子版)”事件就为孤岛。

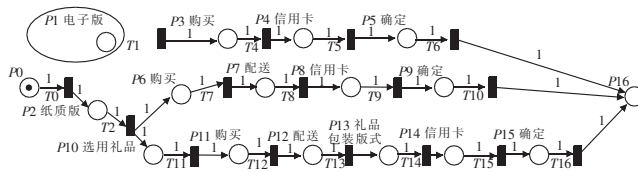


图7 非强连通事件举例

实际应用中还存在一种孤岛问题,将在本节最后介绍。

2.2.3 死锁问题

实际图形用户界面死锁。购买纸质版之后,如果 confirm 界面要求同时满足以下两点,“确定”按钮方能有效:

- 客户必须在勾选配送地址之前选择信用卡号;
- 客户必须选择完配送地址才能选择信用卡号。

则图形用户界面存在死锁,如图8所示。

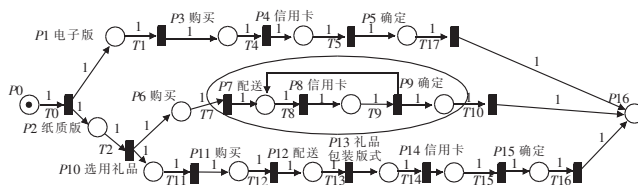


图8 死锁事件举例

2.2.4 无界问题

实际用户界面是无界的,特例就是不安全的。在本简化模型中,默认一次购买一本书,若客户通过多次点击“哈姆雷特(电子版)”和“哈姆雷特(纸质版)”按钮,系统收到多次响应,则该图形用户界面就是无界的 Petri 网描绘,如图9所示。

2.2.5 不符合初始模型问题

不满足预设的 Petri 网模型。此种情形主要是由于图形用户界面的设计不符合需求规约。例如,不论是否在 homepage 界面选择“选用礼品包装”,购买纸质版之后的 confirm 界面都

要求勾选“选择礼品包装型号”。在本情形下的 Petri 网描绘如图 10 所示。

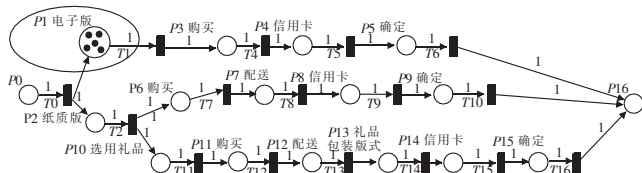


图 9 无界事件举例

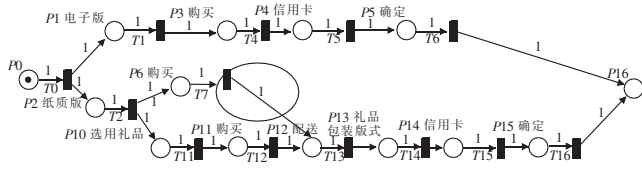


图 10 不满足需求规约的事件举例

还有一类重要的不满足初始模型的问题是在设计实现系统过程中人为加入其他库所,尤其是在图形用户界面测试时引入辅助界面事件,但是在提交系统后未删除。

例如在设计时,为了验证 confirm 界面的“选择信用卡号”并且“确定”后,系统能够正常与银行系统关联,测试人员添加以下测试界面,如图 11 所示。点击 feedback 界面的“确定”按钮后,跳转至 congratulation 界面。

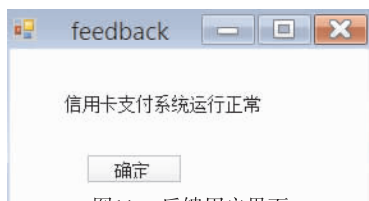


图 11 反馈用户界面

若设计完成后,未删除 feedback 界面,直接提交该系统,则出现软件缺陷,不符合初始模型。事实上,该问题因为不满足预设的 Petri 网模型,可以快速发现。此情形下,实际用户界面的 Petri 网描述,如图 12 所示额外添加了“P17 收到回复”库所,与预设 Petri 网模型不相等。

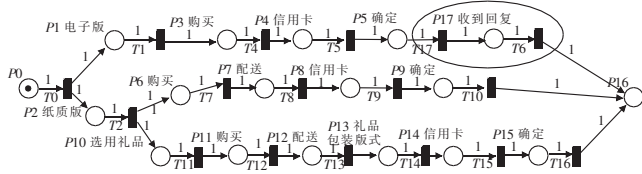


图 12 额外添加事件举例

在实际中,完成测试后,有时虽将事件序列恢复为需求规约的要求,但是并未删除用于测试的代码,这是另一种非可达的孤岛情形,需要代码重构,删除冗余。具体如图 13 所示。在此虽然删除了图 11 中冗余的事件序列,但是并未删除“收到回复”的代码。

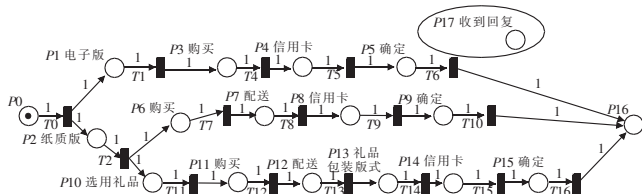


图 13 非重构事件举例

根据以上分析,则可构造实际图形用户界面的 Petri 网模

型,并完成可达树分析,与初始设计的比较,就可发现图形用户界面的潜在错误。

2.2.6 跳转错误

多个图形用户界面中的一类重要缺陷是不正确的跳转。如图 3 中“一键购买”点击后不能到达“成功购买”界面,而是与点击“购买”按钮到达同一界面,如图 14 所示。

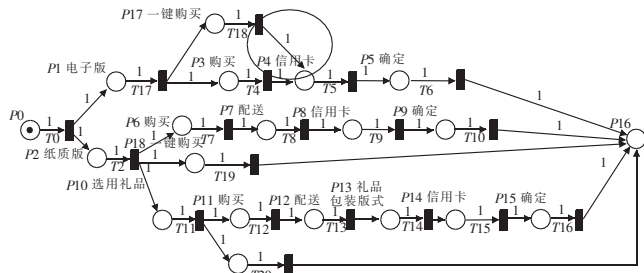


图 14 错误跳转举例

3 实验

在本章实现实例中的订书系统,并对其图形用户界面进行测试。

本实验重点关注以下两个问题:

- 基于 Petri 网的图形用户界面测试能否在不确定的运行环境中发现软件缺陷。
- 基于 Petri 网的图形用户界面测试是否优于随机测试和 OME 方法。

3.1 实验设计

实验环境如下:操作系统为 Windows 8.1 64 bit,处理器为 Intel Core i5 3.13 GHz,内存为 8 GB。开发语言为 C#,开发工具为 Visual Studio 2013 ultimate。实现订书系统图形用户界面。之后,向该系统中植入实例所示的六类缺陷,每类各三个错误,如表 1 所示。

表 1 缺陷以及检测

| 缺陷类型 | 缺陷描述 | 基于 Petri 网测试能否检测 | 随机测试能否检测 | OME 方法 |
|--------------------|--------------------------|------------------|----------|--------|
| 不可达问题 | “选择礼品包装型号” | ✓ | ✓ | × |
| | “选择信用卡号” | ✓ | × | ✓ |
| | confirm 界面“确定” | ✓ | × | × |
| 不强连通问题 (成为孤岛事件) | “哈姆雷特(电子版)” | ✓ | ✓ | ✓ |
| | “选择礼品包装型号” | ✓ | × | ✓ |
| | “选择配送地址” | ✓ | ✓ | × |
| 死锁问题 | “选择配送地址”与“选择信用卡号” | ✓ | × | ✓ |
| | “哈姆雷特(电子版)”与“哈姆雷特(纸质版)” | ✓ | ✓ | ✓ |
| | “选择配送地址”与“确定” | ✓ | × | ✓ |
| 无界问题 | “哈姆雷特(电子版)” | ✓ | × | ✓ |
| | “选用礼品包装” | ✓ | × | ✓ |
| 不符合初始 模型问题 | “选择信用卡号” | × | ✓ | × |
| | 总是要求勾选“选择礼品包装型号” | ✓ | × | ✓ |
| | 额外增加 feedback 界面 | ✓ | × | ✓ |
| 错误跳转 | 总是可以勾选“选择配送地址” | × | ✓ | × |
| | 点击“一键购买”按钮跳转至 confirm 界面 | ✓ | × | ✓ |
| | 点击“购买”按钮跳转至“成功购买”界面 | ✓ | × | ✓ |
| | 点击“一键购买”按钮不跳转 | ✓ | ✓ | × |

设计了三类测试环境,分别为使用基于 Petri 网的图形用户界面测试、随机测试以及 OME 方法^[5],并对系统进行测试。OME 方法的原理是动态迭代监测测试用例中的新事件,与此同时,拓展图形用户界面的输入集模型,最后计算新的覆盖集,

得到新的测试用例以训练新的元素。基于 Petri 网的图形用户界面测试的流程如图 15 所示。

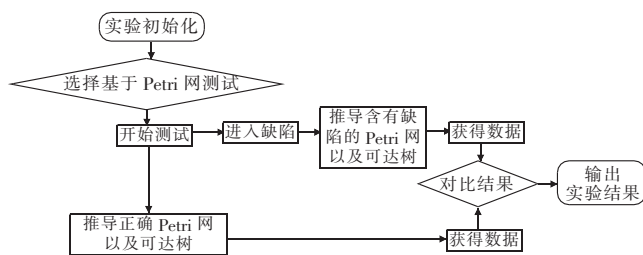


图 15 测试环境

3.2 结果与分析

通过分析表 1, 得出实验结果, 如表 2 所示。统计事件覆盖数、代码行覆盖数(通过使用代码覆盖检测工具 NCover 获得测试结果)以及缺陷发现数。

表 2 结果比较

| 内容 | 基于 Petri 网测试 | 随机测试 | OME 方法 |
|----------|--------------|-------|--------|
| 总测试用例数 | 3 219 | 4 872 | 6 856 |
| 事件覆盖率/% | 96.4 | 33.7 | 87.6 |
| 代码行覆盖率/% | 93.8 | 38.2 | 85.4 |
| 总计检测缺陷数量 | 16 | 7 | 12 |
| 缺陷发现比率/% | 88.9 | 38.9 | 66.7 |

随机测试和 OME 方法的用例数分别比基于 Petri 网的方法的用例数多 51.4% 和 113%, 但后者的事件覆盖率比前者高 62.7%, 代码行覆盖率高 55.6%, 检测到的缺陷比率高 50%。

基于 Petri 网测试方法和 OME 方法均明显优于随机测试。相较而言, OME 方法所用的测试用例数比基于 Petri 网的方法多 113%。因而基于 Petri 网的测试方法优于随机测试和 OME 方法, 如图 16 所示。

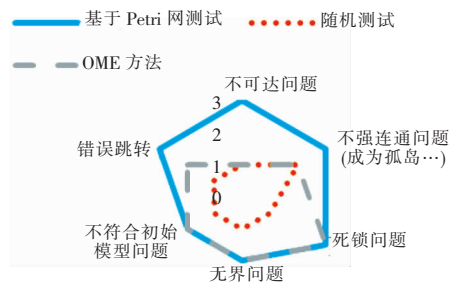


图 16 基于 Petri 网测试与随机测试检测缺陷的有效性比较

结果表明, 基于 Petri 网的图形用户界面测试是有效的, 符合理想预期; 传统的随机测试不能满足当今的图形用户界面需求; 使用基于 Petri 网的测试的总用例数较少、覆盖率高, 达到了优化测试的要求。

事实上, 基于 Petri 网的测试仅仅未能发现两点缺陷(这两个缺陷 OME 方法亦未能找到), 分析如下:

a) “选择信用卡号”, 由于测试的多次点击, 发送多次信息, “确定”事件后, 测试未对此无界事件敏感, 因而误认为通过。

b) 即使购买电子版, 总是可以勾选“选择配送地址”。这是由于在 homepage 界面后, 测试直接跳过“选择配送地址”事件, 遗漏此事件, 进入下一测试, 而未发现缺陷。

以上两点, 至今仍然是软件自动化测试的难题。

4 结束语

图形用户界面在软件开发中越显重要, 与此同时, 对其的测试工作也逐步重视。但是不确定的输入与基于上下文的环

境使得其测试面临诸多困难。

本研究将并发系统领域的 Petri 网理论引入图形用户界面测试, 总结并提出了相关性质, 解决了有关缺陷问题。

至于将来的工作有两个方向: a) 在短期, 移动客户端的开发与研究正在工业界与学界如火如荼的展开, 因而将本研究的相关成果应用于移动客户端是可行的, 尤其是 Windows phone 客户端, 因为本研究实验基于 C#语言与该系统同源; b) 本文尚不能完全自动化测试, 在更长远时间, 研究时序层次化等高级 Petri 网并与图形用户界面测试相结合, 有可能取得更好的测试效果, 因而这也是一个方向。

参考文献:

- [1] Lin Y D, Chu E T H, Yu Shangche, et al. Improving the accuracy of automated GUI testing for embedded systems[J]. IEEE Software, 2014, 31(1): 39-45.
- [2] Bauersfeld S, De Rojas A, Vos T E J. Evaluating rogue user testing in industry: an experience report[C]//Proc of the 8th International Conference on Research Challenges in Information Science, 2014: 1-10.
- [3] Nedyalkova S, Bernardino J. Comparative study of open source capture and replay tools[J]. IEEE (Revista IEEE America Latina) Latin America Transactions, 2014, 12(4): 675-682.
- [4] 赵咏斌, 朱嘉钢, 陆晓. SVM 应用于测试用例生成的方法[J]. 计算机应用研究, 2015, 32(1): 115-120.
- [5] Nguyen B N, Memon A M. An observe-model-exercise* paradigm to test event-driven systems with undetermined input spaces[J]. IEEE Trans on Software Engineering, 2014, 40(3): 216-234.
- [6] Darvish A, Chang C K. Black-box test data generation for GUI testing[C]//Proc of the 14th International Conference on Quality Software. 2014: 133-138.
- [7] Mayo Q, Michaels R, Bryce R. Test suite reduction by combinatorial-based coverage of event sequences[C]//Proc of the 7th International Conference on Software Testing, Verification and Validation Workshops. 2014: 128-132.
- [8] Nakajima H, Masuda T, Takahashi I. GUI ferret: GUI test tool to analyze complex behavior of multi-window applications[C]//Proc of the 18th International Conference on Engineering of Computer Systems. 2013: 163-166.
- [9] Rauf A, Anwar S, Jaffer M A, et al. Automated GUI test coverage analysis using GA[C]//Proc of the 7th International Conference on Information Technology: New Generations. 2010: 1057-1062.
- [10] Cabasino M P, Giua A, Laforune S, et al. A new approach for diagnosability analysis of Petri nets using verifier nets[J]. IEEE Trans on Automatic Control, 2012, 57(12): 3104-3117.
- [11] Mahulea C, Seatzu C, Cabasino M P, et al. Fault diagnosis of discrete-event systems using continuous petri nets[J]. IEEE Trans on Systems, Man and Cybernetics, Part A: Systems and Humans, 2012, 42(4): 970-984.
- [12] Miyamoto T, Horiguchi K. Modular reachability analysis of petri nets for multiagent systems[J]. IEEE Trans on Systems, Man, and Cybernetics: Systems, 2013, 43(6): 1411-1423.
- [13] Sreenivas R S. On the existence of supervisory policies that enforce liveness in partially controlled free-choice petri nets[J]. IEEE Trans on Automatic Control, 2012, 57(2): 435-449.
- [14] Issa A, Sillito J, Garousi V. Visual testing of graphical user interfaces: an exploratory study towards systematic definitions and approaches[C]//Proc of the 14th IEEE International Symposium on Web Systems Evolution. 2012.