

基于成对测试的 GUI 测试用例生成方案及实现

李 峻, 袁兆山

(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

摘 要:在软件开发过程中软件测试的成本很高。其中,测试用例是软件测试的主要组成部分,因此产生合适有效的测试用例是减少软件测试成本的一个主要手段。文章对一个基于成对测试 GUI 测试用例生成方案及实现展开研究;给出了 GUI 对象描述、GUI 测试用例生成、自动执行以及结果比较的过程;以项目 CQMIS 为例,使用自动化测试工具 QTP,依据所提出的方案实施测试用例生成,效果良好。

关键词:软件测试;成对测试;测试用例;QTP

中图分类号:TP311.53

文献标识码:A

文章编号:1003-5060(2009)11-1719-04

Scheme for GUI test case generation based on pairwise testing and its implementation

LI Jun, YUAN Zhao-shan

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: The testing cost is high in software development, and the test case is the most important part of software testing, so producing effective test cases is an important means to reduce the cost of software testing. This paper researches a scheme for test case generation based on pairwise testing and its implementation. A process is put forward which includes GUI object description, GUI test case generation, automated execution and comparison of the results. An example of the project CQ-MIS is given, and the automation testing tool QTP is used. The scheme is proved to be effective by practical implementation.

Key words: software testing; pairwise testing; test case; QTP

图形用户界面 GUI(Graphical User Interface)在现今软件中的应用已经相当普遍,并成为大部分应用软件与用户交互的重要手段。GUI 应用程序的主要特征是对象驱动,用户通过鼠标选择、键盘输入和打开关闭窗口等手段与软件发生交互^[1]。GUI 给软件使用者带来了极大的便利,使软件易用性增强。因此,GUI 的测试也越来越受到重视,而测试用例是软件测试中的重要部分。测试用例是为了某个目标而设计的一组测试输入,执行和实际结果与预期结果比较的过程^[2]。对于 GUI 的软件测试来说,即使能通过覆盖准则使测试集中到 GUI 的某个部分,但是由于测试用例状态空间的巨大,要为这些选定的部分

产生所有的测试用例也不太现实,只能选取所有测试用例的一个子集作为测试用例集^[3]。因此,一个很重要的问题是如何产生大小合适的 GUI 测试用例集,使测试用例能尽可能覆盖所有 GUI 对象的不同情况。

1 基于成对测试 GUI 测试用例方案

1.1 成对测试含义

成对测试是一种面向应用的测试方案^[4],所要达到的目标是要求在最终生成的测试用例集合中对系统的所有对象可能取值的两两组合(配对)至少覆盖一次。研究表明,软件的大多数错误不是由于多个参数的复杂相互交互引起的,而是由

收稿日期:2008-11-17

基金项目:安徽省淮北矿业集团公司科技资助项目(06201);合肥工业大学科学研究发展基金资助项目(050502F)

作者简介:李 峻(1982—),男,安徽合肥人,合肥工业大学硕士生;

袁兆山(1945—),男,山东苍山人,合肥工业大学教授,硕士生导师。

于一个参数或 2 个参数的简单相互作用引起的。目前,基于成对测试的 GUI 测试用例生成的研究还不够成熟,绝大多数有关成对测试都是在理论上探讨,很少用于实际使用。本文结合自动化测试工具 QTP,对 GUI 测试用例生成进行研究,提出一个基于成对测试 GUI 测试用例生成方案。该方案对完整的 GUI 测试用例生成过程,包括 GUI 对象描述、对象取值、测试用例生成、测试用例自动执行以及结果自动比较的过程,进行较详细的论述。最后以项目 CQMIS (Coal Quality Management Information System) 为例,结合自动化测试工具 QTP,说明方案实施的可行性。

成对测试也叫两两测试或 2 路测试^[5],其定义如下:

定义 1 假设测试模型涉及到 K 个参数,且每个参数的可能取值为有限个。所谓成对测试是指测试用例集覆盖这 K 个参数中的所有两两参数组合。也即对于任意 2 个参数 P_1 和 P_2 ,在测试中存在一个测试用例满足参数 P_1 和 P_2 出现在测试用例中。

成对测试用例生成策略,主要表现为以一个空测试用例集开始的递归过程,即每步添加一个新参数,在前一步所产生的测试用例集基础上,依据算法生成新的用例集,以完全覆盖当前已加入的所有参数各个值之间的两两组合。在最后一个参数添加完成后所产生的测试用例集中,即可实现对所有参数值两两组合的完全覆盖。

1.2 GUI 对象表示

GUI 反映了窗口、会话、功能菜单等一系列对象之间相互组合^[6]。在 GUI 场景中用户关注能够打开一个特定的窗口或点击一个下拉菜单,从而能够实现某种功能。GUI 的重要特征包括 GUI 的对象位置、对象类型、对象属性以及输入要求等^[7]。

GUI 定义:图形用户界面(GUI)是一个包括等级制度,以图形为前端表示的软件系统。其中包括可接受的用户输入产生和系统事件产生,并且根据一组确定对象可以生成预定的图形输出值。GUI 中每个对象都有一组固定的属性,在执行 GUI 的任何时间中,这些属性都有离散值,这些具有离散值的对象构成了一个 GUI 的状态。

GUI(图形用户界面)由 3 部分组成:① 对象(如用户名、密码),用符号 O 表示,一个对象可以表示为 $O_i \in O$ 。② 对象属性(如文本输入框、下拉选择框),用符号 P 表示,对象的属性表示为 $P_i \in$

P 。③ 值(如日期、百分数及文字),每一个对象都拥有相应值,用 V 表示,对象的值表示为 $V_i \in V$ 。因此,任何时刻 GUI 的某一对象可以用三元组集合表示为 (O_i, P_i, V_i) 。

1.3 基于成对测试的测试用例生成

典型的有若干个对象的 GUI 由两大类对象组成:必填对象和选填对象。这两部分测试对象的测试集合如定义 2 和定义 3。

定义 2 对于必须填写的对象取值,当对象是多选一类型时,即对象中的每一个选择都是可选择填写的且对象必须填。此对象集合用三元组的方式表示为: $\{(对象名, 属性, 值 1), \dots, (对象名, 属性, 值 n)\}$ 。当对象是单一时,将这些对象统一成一个测试对象集合。

定义 3 对于可选填写的对象,即对象可填可不填时,当对象是多选一类型时,此对象集合用三元组的方式表示为: $\{(对象名, 属性, 值 1), \dots, (对象名, 属性, 值 n), (对象名, 属性, 空白)\}$ 。其中,对象名的含义是在测试用例中不选择此对象进行测试。当对象是单一时,用 $\{(对象名, 属性, 值), (对象名, 属性, 空白)\}$ 表示。这里的对象是可选填的,因此对于单一填写的对象不可统一为一个测试对象集合,必须分开。

如图 1 所示,如果此对象是必填对象,此对象集合是 $\{(区(队)别, 下拉框, 一区), (区(队)别, 下拉框, 二区), (区(队)别, 下拉框, 三区)\}$ 。如果是可填对象,其集合为 $\{(区(队)别, 下拉框, 一区), (区(队)别, 下拉框, 二区), (区(队)别, 下拉框, 三区), (区(队)别, 下拉框, 空白)\}$ 。

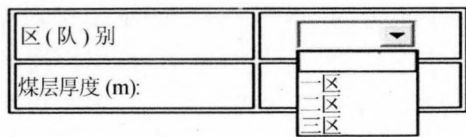


图 1 测试对象集合示例

对列出的所有对象集合,使用成对测试的方法生成测试用例,具体步骤如下:

(1) 将 GUI 对象的第一和第二个对象集合中所有三元组做两两的完全组合,生成测试用例集。

(2) 取下一个对象 $i (i = 3, 4, \dots, m)$, 把第 i 对象集合的所有三元组依次添加到第(1)步中的测试用例集中。考察此时的测试用例集中是否存在未两两配对的情况。若有未配对的情况,则对这些未配对的情况,使用水平扩展算法(IPO-

H)和垂直扩展算法(IPO-V)生成新的测试用例^[8]。在使用垂直扩展算法产生新的测试用例时,在对象三元组不受选择影响时,选择那些在已有的测试用例中出现次数较少的三元组。

(3) 若 GUI 的所有对象都处理完毕,结束并产生测试用例。若还有对象未处理,转入第(2)步继续执行。

1.4 测试用例自动执行

测试用例生成之后,可以使用流行的自动化测试工具 QTP 执行测试脚本达到自动执行。

(1) 动作分割技术。通常在建立一个全新的测试脚本时,测试脚本中只有一个操作流程即动作。如果把整个系统中所有的 GUI 全部放入一个动作中,不仅测试脚本冗长不易维护,而且再次使用的价值也不大。动作技术分割实现了测试脚本的模块化且更容易被重复使用。可以认为每个 GUI 是一个独立动作,并对其中的对象编写测试用例。执行时可以是一个动作执行,也可以在一个动作之前或后插入若干动作形成一个测试流程。这种方法最大的好处是:同一个脚本可以重复使用,提高脚本利用率。测试用例中对象数量较小,容易管理 GUI 对象,脚本的维护变得方便。

(2) 场景恢复问题。在测试脚本执行中一个很重要的问题是运行期间的意外事件、错误和应用程序崩溃会破坏运行并扭曲结果。尤其是在无人值守情况下运行时,测试会挂起,直到执行恢复所需的操作。可以通过以下策略解决此问题:① 恢复场景机制:创建恢复场景并将它们保存在恢复文件中,该文件是将多个恢复场景组织和存储在一起的一种简便方法。创建恢复场景后,将它们与选定的测试用例相关联,以便在发生触发器事件时执行相应的场景。② 建立完善的异常处理机制:出现异常情况是导致运行出错的主要原因。当检测到异常情况时,首先记录异常消息类型,通过发送特殊消息或选择有焦点的按钮关闭异常消息框,然后清除当前测试用例的执行现场,包括恢复 GUI、文件和数据库等,最后启动下一条测试用例。③ 对象控制机制:可以创建 Recovery 对象在运行期间以编程方式控制恢复。在运行的特定部分启用 Recovery 对象,检索有关测试用例的状态信息。

1.5 测试用例运行结果验证

测试用例经过执行后会产生一些实际输出值,而要判断其测试用例能否通过测试就需要和预期的标准输出相比较^[9]。可以通过在测试过程

中加入“检查点”来检验输出的指定状态与预期状态是否一致。测试人员不必编写测试准则程序,通过使用自动化测试工具即可以实现测试用例结果自动比较。可以在“检查点”时根据需要使用万用字符,以方便检查某个特定文字,使测试脚本更加有弹性、更能适合应用程序的变化。

2 项目实施

煤炭质量管理信息系统(CQMIS)是为淮北集团公司开发的应用系统,开发环境为 Eclipse3.1,采用 C/S 结构,涉及到的 GUI 较多且逻辑性强。本节将以 CQMIS 中的煤质计划管理模块的年度煤质计划信息的工作面煤质计划信息部分,作为例子来说明。

2.1 建立 GUI 对象

如图 2 所示,本 GUI 中共有 2 个必填的选择框对象、3 个必填的单一文本输入框对象、2 个可选填的选择输入框对象和 6 个可选填的单一输入框对象。

此 GUI 对象集合表示如下:

工作面: {(工作面,下拉框,1 采区), (工作面,下拉框,2 采区), (工作面,下拉框,3 采区)}。
区队: {(区队,下拉框,一区), (区队,下拉框,二区), (区队,下拉框,三区)}。
夹矸岩性: {(夹矸岩性,下拉框,炭泥岩), (夹矸岩性,下拉框,泥岩), (夹矸岩性,下拉框,空白)}。
落煤方式: {(落煤方式,下拉框,炮采), (落煤方式,下拉框,空白)}。
开始日期: {(开始日期,文本框,2007-10-10)}。
结束日期: {(开始日期,文本框,2007-12-01)}。
剔除率: {(剔除率,数字框,90%), (剔除率,数字框,空)}。
影响灰分: {(影响灰度,数字框,60%), (剔除前灰度,数字框,空)}。
剔除前灰度: {(剔除前灰度,数字框,60%), (光矸层数,数字框,空)}。
剔除后灰度: {(剔除后灰度,数字框,70%), (剔除后灰度,数字框,空)}。

工作面:	<input type="text" value="1采区"/>	区队:	<input type="text" value="一区"/>
夹矸岩性:	<input type="text" value="炭泥岩"/>	落煤方式:	<input type="text" value="炮采"/>
开始日期:	<input type="text" value=""/>	结束日期:	<input type="text" value=""/>
剔除率(%)	<input type="text" value=""/>	影响灰分(%)	<input type="text" value=""/>
夹矸层数(层)	<input type="text" value=""/>	夹矸厚度(m)	<input type="text" value=""/>
剔除前灰分(%)	<input type="text" value=""/>	剔除后灰分(%)	<input type="text" value=""/>
可采煤层厚度(m)	<input type="text" value=""/>		<input type="text" value=""/>

图 2 含有 13 个对象的 GUI

2.2 测试用例生成

对上述的对象集合使用基于成对测试的 GUI

测试用例生成方案来测试用例,如图 3 所示。

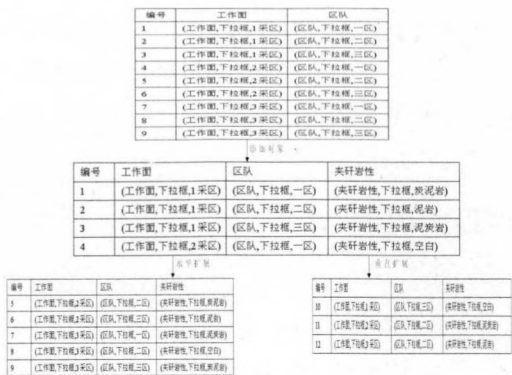


图 3 基于成对测试的测试用例生成

2.3 具体实现

使用自动化测试工具 QTP,把这些测试用例编写成测试脚本进行测试。编写方式如下:

(1) 用 Widget 类存储 GUI 中的对象。创建 Widget 类来存储 GUI 对象,在 Widget 类中定义的内容如下:用来表示每个对象对应 QTP 中对象名的变量 widgetName;用来表示每个对象的类型的方法 widgetType;用来标记此对象是否必须填写的变量 widgetUse,默认为 false,当该对象必须填写时为 ture。这里使用的对象名与 QTP 中定义的对象名相同,在 QTP 中可以把 GUI 中所有的对象加入到对象仓库中,并且可以根据需要修改名称。

(2) 用 Value 类存储对象的值。创建 Value 类描述对象的所有可能取值,定义各种不同类型的属性如字符串、数字和百分数等,用来表示每个对象属性和需要的属性值相联系的方法 valueConnect。

(3) 用 PairWise 类组合测试对象。创建 PairWise 类对对象进行测试用例组合生成,创建用来获取对象信息的 pairwiseObject 方法;使用水平扩展的 pairwiseLevel 方法;使用垂直扩展的 pairwiseVertical 方法。

(4) QTP 测试脚本。可以首先创建一个 Test,把所有相关对象仓库的 tsr 文件添加到这个 Test 中;然后利用上面的 3 个 Java 类,生成 GUI 测试用例对象组合,把组合过的测试对象用 Java 语言以 QTP 测试脚本的书写规范生成 mts 文件;最后把 mts 文件导入到 Test 中,利用 QTP 回放技术运行测试用例。

将基于成对测试生成测试用例的技术应用在 CQMIS 项目测试中。以上述项目模块为样例,利

用 QTP 进行自动化测试。原先需要的测试用例数为 $18\,432(3 \times 3 \times 4 \times 4 \times 2^7)$ 个测试用例。现在仅需要 86 个测试用例就可以完成测试。通过 QTP 回放脚本,这 86 个测试用例发现了 6 个 Bug,比以前多发现了 3 个 Bug 并且比手工测试缩短了 1 d 时间。实践证明,使用这种方案确实提高了发现 Bug 的能力,并大大减少了测试成本,提高了测试效率。

3 结束语

本文针对 GUI 测试用例繁多,执行效率低下问题,提出一个基于成对测试的测试用例生成方案,并且实现了测试用例生成和测试自动化。该方案成功应用于 CQMIS 的测试中,简化了测试用例的数量,提高了测试效率和发现错误能力。但是软件测试中往往 GUI 的对象很复杂,构建 GUI 对象的工作量比较大,所以可进一步研究和探讨如何使用本体库构建 GUI 对象。

[参 考 文 献]

[1] Yuan Xun, Memon A M. Using GUI run-time state as feedback to generate test cases[C]//Proceedings of the 29th International Conference on Software Engineering, 2007: 396—405.

[2] Reid S, Titterton T. Working draft of BS 7925-1[EB/OL]. <http://www.testingstandards.co.uk/bs-7925-1-online.html>, 2008-11-10.

[3] Pollack M E, Soffa M L. Automated test oracles for GUIs [C]//Proceedings of Eighth International Symposium on the Foundations of Software Engineering, 2000: 6—10.

[4] 向 渝, 张 骏, 王文勇. 成对测试中的一种用例生成算法[J]. 电子科技大学学报, 2005, 34(7): 821—824.

[5] 曾劲涛, 陈建明. 有参数约束的两两组合覆盖测试用例生成的研究[J]. 苏州大学学报(自然科学版), 2008, 24(10): 45—49.

[6] Brooks P A, Memon A M. Automated GUI testing guided by usage profiles[C]//Proceedings of 22th IEEE/ACM International Conference on Automated Software Engineering, 2007: 333—342.

[7] Cai Kai-Yuan, Zhao Lei. On the test case for GUI testing[C]//Proceedings of the 5th International Conference on Quality Software, 2005: 19—28.

[8] Tai K C, Lei Y. A test generation strategy for pairwise testing [J]. IEEE Transactions on Software Engineering, 2002, 28(1): 109—111.

[9] Hayat M U, Qadeer M. Intra Component GUI test case generation technique[C]//Information and Emerging Technologies, IC IET, 2007: 1—5.

(责任编辑 张秋娟)