



# (12)发明专利申请

(10)申请公布号 CN 108491321 A

(43)申请公布日 2018.09.04

(21)申请号 201810183368.4

(22)申请日 2018.03.06

(71)申请人 平安科技(深圳)有限公司

地址 518000 广东省深圳市福田区八卦岭  
工业区平安大厦六楼

(72)发明人 陈晓芳

(74)专利代理机构 深圳市沃德知识产权代理事

务所(普通合伙) 44347

代理人 于志光 郭梦霞

(51)Int.Cl.

G06F 11/36(2006.01)

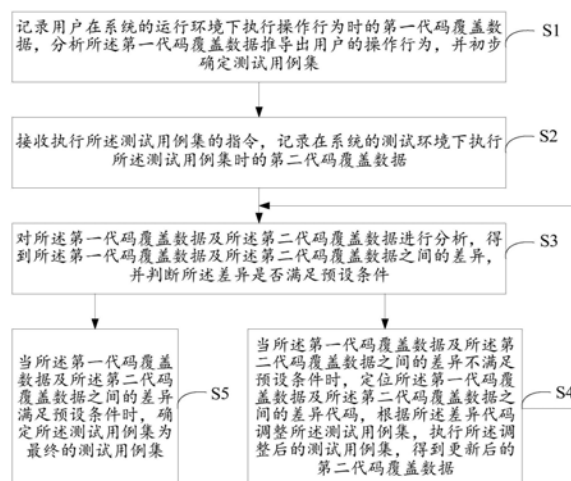
权利要求书3页 说明书9页 附图2页

## (54)发明名称

测试用例范围确定方法、装置及存储介质

## (57)摘要

本发明提出一种测试用例范围确定方法,包括:记录用户在系统的运行环境下执行操作行为时的第一代码覆盖数据,初步确定测试用例集;记录在系统的测试环境下执行所述测试用例集时的第二代码覆盖数据;计算所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异;当差异不满足预设条件时,定位所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异代码,调整所述测试用例集,执行所述调整后的测试用例集,得到更新后的第二代码覆盖数据;及,当差异满足预设条件时,确定所述测试用例集为最终的测试用例集。本发明还提出一种电子装置及存储介质。利用本发明,可快速、准确地确定测试用例范围,使测试用例范围贴近用户的实际操作,提高测试效率。



1. 一种测试用例范围确定方法,应用于电子装置,其特征在于,该方法包括:

S1、记录用户在系统的运行环境下执行操作行为时的第一代覆盖数据,分析所述第一代覆盖数据推导出用户的操作行为,并初步确定测试用例集;

S2、接收执行所述测试用例集的指令,记录在系统的测试环境下执行所述测试用例集时的第二代覆盖数据;

S3、对所述第一代覆盖数据及所述第二代覆盖数据进行分析,得到所述第一代覆盖数据及所述第二代覆盖数据之间的差异,并判断所述差异是否满足预设条件;

S4、当所述第一代覆盖数据及所述第二代覆盖数据之间的差异不满足预设条件时,定位所述第一代覆盖数据及所述第二代覆盖数据之间的差异代码,根据所述差异代码调整所述测试用例集,执行所述调整后的测试用例集,得到更新后的第二代覆盖数据,流程返回至S3;及

S5、当所述第一代覆盖数据及所述第二代覆盖数据之间的差异满足预设条件时,确定所述测试用例集为最终的测试用例集。

2. 如权利要求1所述的测试用例范围确定方法,其特征在于,所述步骤S3包括:

分别读取所述第一代覆盖数据中包含的各包在所述第一代覆盖数据和所述第二代覆盖数据中的覆盖率;

分别计算各包在所述第一代覆盖数据和所述第二代覆盖数据中的覆盖率的差异度;及

判断所述各包的差异度是否均小于或等于预设阈值。

3. 如权利要求1或2所述的测试用例范围确定方法,其特征在于,所述“当所述第一代覆盖数据及所述第二代覆盖数据之间的差异不满足预设条件时,确定所述第一代覆盖数据及所述第二代覆盖数据之间的差异代码”的步骤包括:

当所述各包中包含差异度大于预设阈值的一个或多个包时,筛选出所述一个或多个包;

分别对比所述一个或多个包中包含的类在所述第一代覆盖数据及所述第二代覆盖数据中的覆盖率,筛选出在所述第一代覆盖数据及所述第二代覆盖数据中的覆盖率不同的一个或多个类;

分别对比所述一个或多个类中包含的方法在所述第一代覆盖数据及所述第二代覆盖数据中的覆盖率,筛选出在所述第一代覆盖数据及所述第二代覆盖数据中的覆盖率不同的一个或多个方法;及

分别对比所述一个或多个方法中包含的代码行在所述第一代覆盖数据及所述第二代覆盖数据中的覆盖率,筛选出在所述第一代覆盖数据及所述第二代覆盖数据中的覆盖率不同的代码行,确定所述第一代覆盖数据及所述第二代覆盖数据之间的差异代码。

4. 如权利要求3所述的测试用例范围确定方法,其特征在于,所述第一代覆盖数据包括:用户在生产环境下执行所述操作流程,系统代码中被覆盖的包、类、方法及行信息。

5. 如权利要求4所述的测试用例范围确定方法,其特征在于,所述第二代覆盖数据包括:在测试环境下执行所述测试用例集,系统代码中被覆盖的包、类、方法及行信息。

6. 一种电子装置,其特征在于,该装置包括:存储器、处理器,所述存储器上存储有可在

所述处理器上运行的测试用例范围确定程序,该程序被所述处理器执行时实现如下步骤:

A1、记录用户在系统的运行环境下执行操作行为时的第一代码覆盖数据,分析所述第一代码覆盖数据推导出用户的操作行为,并初步确定测试用例集;

A2、接收执行所述测试用例集的指令,记录在系统的测试环境下执行所述测试用例集时的第二代码覆盖数据;

A3、对所述第一代码覆盖数据及所述第二代码覆盖数据进行分析,得到所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异,并判断所述差异是否满足预设条件;

A4、当所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异不满足预设条件时,定位所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异代码,根据所述差异代码调整所述测试用例集,执行所述调整后的测试用例集,得到更新后的第二代码覆盖数据,流程返回至A3;及

A5、当所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异满足预设条件时,确定所述测试用例集为最终的测试用例集。

7.如权利要求6所述的电子装置,其特征在于,所述步骤A3包括:

分别读取所述第一代码覆盖数据中包含的各包在所述第一代码覆盖数据和所述第二代码覆盖数据中的覆盖率;

分别计算各包在所述第一代码覆盖数据和所述第二代码覆盖数据中的覆盖率的差异度;及

判断所述各包的差异度是否均小于或等于预设阈值。

8.如权利要求6或7所述的电子装置,其特征在于,所述“当所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异不满足预设条件时,确定所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异代码”的步骤包括:

当所述各包中包含差异度大于预设阈值的一个或多个包时,筛选出所述一个或多个包;

分别对比所述一个或多个包中包含的类在所述第一代码覆盖数据及所述第二代码覆盖数据中的覆盖率,筛选出在所述第一代码覆盖数据及所述第二代码覆盖数据中的覆盖率不同的一个或多个类;

分别对比所述一个或多个类中包含的方法在所述第一代码覆盖数据及所述第二代码覆盖数据中的覆盖率,筛选出在所述第一代码覆盖数据及所述第二代码覆盖数据中的覆盖率不同的一个或多个方法;及

分别对比所述一个或多个方法中包含的代码行在所述第一代码覆盖数据及所述第二代码覆盖数据中的覆盖率,筛选出在所述第一代码覆盖数据及所述第二代码覆盖数据中的覆盖率不同的代码行,确定所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异代码。

9.如权利要求8所述的电子装置,其特征在于,所述第一代码覆盖数据包括:用户在生产环境下执行所述操作流程,系统代码中被覆盖的包、类、方法及行信息;及

所述第二代码覆盖数据包括:在测试环境下执行所述测试用例集,系统代码中被覆盖的包、类、方法及行信息。

10.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有测试用

例范围确定程序,该程序被处理器执行时实现如权利要求1至5中任一项所述的测试用例范围确定方法的步骤。

## 测试用例范围确定方法、装置及存储介质

### 技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种测试用例范围确定方法、电子装置及计算机可读存储介质。

### 背景技术

[0002] 测试是指在规定的条件下对程序进行操作,以发现程序错误,衡量软件质量,并对其是否能满足设计要求进行评估的过程。测试用例(Test Case)是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果,以便测试某个程序路径或核实是否满足某个特定需求。

[0003] 现在,各个系统或软件版本迭代周期都较短,若要在较短时间内完成测试,这就要求有明确的测试范围。目前,测试范围大都通过预估或通过拓扑的方式来获取,但获取的范围有限,使测试的范围不够全面。因此,需要提供一种能快速、准确确定测试用例范围的方法。

### 发明内容

[0004] 本发明提供一种测试用例范围确定方法、电子装置及计算机可读存储介质,其主要目的在于快速、准确地确定测试用例范围,使测试用例范围贴近用户的实际操作,提高测试效率。

[0005] 为实现上述目的,本发明提供一种测试用例范围确定方法,该方法包括:

[0006] S1、记录用户在系统的运行环境下执行操作行为时的第一代码覆盖数据,分析所述第一代码覆盖数据推导出用户的操作行为,并初步确定测试用例集;

[0007] S2、接收执行所述测试用例集的指令,记录在系统的测试环境下执行所述测试用例集时的第二代码覆盖数据;

[0008] S3、对所述第一代码覆盖数据及所述第二代码覆盖数据进行分析,得到所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异,并判断所述差异是否满足预设条件;

[0009] S4、当所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异不满足预设条件时,定位所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异代码,根据所述差异代码调整所述测试用例集,执行所述调整后的测试用例集,得到更新后的第二代码覆盖数据,流程返回至S3;及

[0010] S5、当所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异满足预设条件时,确定所述测试用例集为最终的测试用例集。

[0011] 此外,为实现上述目的,本发明还提供一种电子装置,该装置包括存储器、处理器,所述存储器上存储有可在所述处理器上运行的测试用例范围确定程序,该程序被所述处理器执行时实现如上所述的测试用例范围确定方法的任意步骤。

[0012] 此外,为实现上述目的,本发明还提供一种计算机可读存储介质,所述计算机可读存储介质上存储有测试用例范围确定程序,该程序被处理器执行时实现如上所述的测试用

例范围确定方法的任意步骤。

[0013] 相较于现有技术,本发明提出的测试用例范围确定方法、电子装置及计算机可读存储介质,根据用户在系统的运行环境下执行操作行为得到的第一代代码覆盖数据,初步确定测试用例范围,缩小了测试用例的有效,提高测试用例范围确定的效率;并在系统测试环境下执行测试用例得到第二代代码覆盖数据,对比两组代码覆盖数据得到差异代码,根据差异代码调整测试用例,确定最终的测试用例范围,使测试用例范围最大限度地接近用户实际的操作行为,提高测试效率。

## 附图说明

[0014] 图1为本发明服务器较佳实施例的示意图;

[0015] 图2为图1中测试用例范围确定程序的程序模块示意图;

[0016] 图3为本发明测试用例范围确定方法较佳实施例的流程图。

[0017] 本发明目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

## 具体实施方式

[0018] 应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0019] 本发明提供一种服务器1。参照图1所示,为本发明服务器1较佳实施例的示意图。

[0020] 在本实施例中,服务器1可以是机架式服务器、刀片式服务器、塔式服务器或机柜式服务器。

[0021] 该服务器1包括存储器11、处理器12,通信总线13,网络接口14以及显示器15。

[0022] 其中,存储器11至少包括一种类型的可读存储介质,所述可读存储介质包括闪存、硬盘、多媒体卡、卡型存储器(例如,SD或DX存储器等)、磁性存储器、磁盘、光盘等。存储器11在一些实施例中可以是所述服务器1的内部存储单元,例如该服务器1的硬盘。存储器11在另一些实施例中也可以是所述服务器1的外部存储设备,例如该服务器1上配备的插接式硬盘,智能存储卡(Smart Media Card, SMC),安全数字(Secure Digital, SD)卡,闪存卡(Flash Card)等。进一步地,存储器11还可以既包括该服务器1的内部存储单元也包括外部存储设备。存储器11不仅可以用于存储安装于该服务器1的应用软件及各类数据,例如测试用例范围确定程序10等,还可以用于暂时地存储已经输出或者将要输出的数据。

[0023] 处理器12在一些实施例中可以是一中央处理器(Central Processing Unit, CPU)、控制器、微控制器、微处理器或其他数据处理芯片,用于运行存储器11中存储的程序代码或处理数据,例如测试用例范围确定程序10等。

[0024] 通信总线13用于实现这些组件之间的连接通信。

[0025] 网络接口14可以包括标准的有线接口、无线接口(如WI-FI接口),通常用于在该服务器1与其他电子设备之间建立通信连接。例如,服务器1可以通过网络接口14从代码覆盖率工具的服务器端(图中未标出)提取代码覆盖数据,所述代码覆盖率工具的服务器端部署服务器1端,用于接收代码覆盖率工具的客户终端(图中未标出)发送的代码覆盖数据,所述代码覆盖率工具的客户终端部署在待测试系统上。

[0026] 显示器15(Display)可以是LED显示器、液晶显示器、触控式液晶显示器以及有机发光二极管(Organic Light-Emitting Diode, OLED)触摸器等。其中,显示器15也可以称为

显示屏或显示单元,用于显示在服务器1中处理的信息以及用于显示可视化的用户界面。

[0027] 图1仅示出了具有组件11-14的服务器1,但是应理解的是,并不要求实施所有示出的组件,可以替代的实施更多或者更少的组件。

[0028] 可选地,该服务器1还可以包括用户接口,用户接口可以包括输入单元比如键盘(Keyboard),可选的用户接口还可以包括标准的有线接口、无线接口。

[0029] 在图1所示的实施例中,存储器11中存储有测试用例范围确定程序10。处理器12执行存储器11中存储的测试用例范围确定程序10时实现如下步骤:

[0030] A1、记录用户在系统的运行环境下执行操作行为时的第一代码覆盖数据,分析所述第一代码覆盖数据推导出用户的操作行为,并初步确定测试用例集;

[0031] A2、接收执行所述测试用例集的指令,记录在系统的测试环境下执行所述测试用例集时的第二代码覆盖数据;

[0032] A3、对所述第一代码覆盖数据及所述第二代码覆盖数据进行分析,得到所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异,并判断所述差异是否满足预设条件;

[0033] A4、当所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异不满足预设条件时,定位所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异代码,根据所述差异代码调整所述测试用例集,执行所述调整后的测试用例集,得到更新后的第二代码覆盖数据,流程返回至A3;及

[0034] A5、当所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异满足预设条件时,确定所述测试用例集为最终的测试用例集。

[0035] 本实施例中,以系统P为例对本发明的方案进行说明。在系统P部署代码覆盖率工具的客户端,在服务器1端部署代码覆盖率工具的服务器端,例如,Jacoco代码覆盖率工具的客户端及服务器端。具体地,Jacoco代码覆盖率工具的客户端用于记录用户在系统的实际运行环境下执行某操作行为,及在系统的测试环境下执行测试用例时,系统代码的被覆盖情况;Jacoco代码覆盖率工具的服务器端用于接收Jacoco代码覆盖率工具的客户端记录的系统代码的被覆盖情况。

[0036] 在对进行版本更新后的系统P进行测试时,需确保测试用例的范围满足测试需求。启动部署在系统P的Jacoco代码覆盖率工具的客户端,对用户在实际运行环境下的操作行为进行监控。其中,操作行为包括:用户点击系统、执行系统的某个功能模块或者执行某些操作流程,Jacoco代码覆盖率工具的客户端会记录下被用户执行的功能模块或者操作流程对应的系统代码,并生成第一代码覆盖数据。Jacoco代码覆盖率工具的客户端将生成的第一代码覆盖数据发送至Jacoco代码覆盖率工具的服务器端,服务器1从Jacoco代码覆盖率工具的服务器端摘取第一代码覆盖数据,并通过服务器1的显示器向测试人员展示,供测试人员针对第一代码覆盖数据进行分析,以初步确定系统P的测试用例范围。可以理解的是,被用户执行过的功能模块或者操作流程,其在系统P中对应的代码都会被覆盖到,反之,未被执行过的功能模块或者操作流程对应的代码则不会被覆盖。具体地,所述第一代码覆盖数据包括:用户在系统P的实际运行环境下执行上述操作行为时,系统P的代码中被覆盖的包、类、方法及行信息。在第一代码覆盖数据收集过程中,计算了对应的包、类、方法的第一覆盖率,以及被覆盖代码的行数占系统P的代码的总行数的第一比例。

[0037] 进一步地,服务器1从Jacoco代码覆盖率工具的服务器端摘取出第一代码覆盖数

据后,对第一代代码覆盖数据中被覆盖的代码进行分析。可以理解的是,系统的功能模块或者操作流程由均代码实现,因此,根据第一代代码覆盖数据中被覆盖的代码,可以推导出用户在系统P的实际运行环境下执行了哪些功能模块或者操作流程。根据推导得出的操作流程分别确定各操作流程的场景,并进一步得到对应的测试用例,生成初步确定的测试用例集。例如,对第一代代码覆盖数据中的代码数据进行挖掘,发现MD5算法对应的代码被覆盖,sha-1算法对应的代码未被覆盖,那么可以确定用户执行的操作流程中调用了MD5算法,测试人员在设计测试用例时,则需要覆盖MD5算法的调用场景。确定测试用例集为本领域人员习知的技术手段,在此不作累述。

[0038] 接收执行测试用例集的指令,开启部署在系统P的测试环境下的Jacoco代码覆盖率工具的客户端,并在系统P的测试环境下执行上述步骤得到的测试用例集中的测试用例,记录执行测试用例时系统P的代码的被覆盖情况,作为第二代代码覆盖数据。具体地,第二代代码覆盖数据包括:在系统P的测试环境下执行测试用例时,系统P的代码中被覆盖的包、类、方法及行信息。同样,在第二代代码覆盖数据收集过程中,计算了对应的包、类、方法的第二覆盖率,以及被覆盖代码的行数占系统P的代码的总行数的第二比例。

[0039] 需要说明的是,在执行测试用例时,可能出现返回的测试结果与预期结果不一致的情况。在本方案中,强调的是用户的操作行为,重点关注的是用户执行的操作流程,因为输入的具体值本身就存在差异,而且,测试结果对系统代码的被覆盖情况不会有影响,故不会影响第二代代码覆盖数据。

[0040] 两组代码覆盖数据中被覆盖部分包含多个包,每个包中包含类,类中包含方法,方法中包含代码行。如果某包在两组代码覆盖数据中的覆盖率的差异不满足预设条件,说明该包中包含的类、类中包含的方法、方法中包含的代码行也存在差异,进一步说明初步确定的测试用例集中未覆盖到的、用户在系统P的实际运行环境下执行的操作行为,需进一步完善测试用例。

[0041] 具体地,分别读取第一代代码覆盖数据中包含的各包在第一代代码覆盖数据和第二代代码覆盖数据中的覆盖率,计算各包在第一代代码覆盖数据和第二代代码覆盖数据中的覆盖率的差异度;筛选出差异度小于预设阈值对应的包,确定差异小于预设阈值的该部分包对应的差异代码。其中,差异度的计算公式为:

[0042]  $M = (A - B) / A$

[0043] 其中,M为某包在第一代代码覆盖数据中的覆盖率与第二代代码覆盖数据中的覆盖率之间的差异度,A为该包在第一代代码覆盖数据中的覆盖率,B为该包在第二代代码覆盖数据中的覆盖率。

[0044] 例如,某包在第一代代码覆盖数据中的覆盖率为40%,在第二代代码覆盖数据中的覆盖率为35%,根据计算公式计算该包在两组代码覆盖数据中的覆盖率的差异度为5%,假设预设阈值为3%,差异度大于预设阈值,则需要进一步确定该包对应的差异代码。进一步对比该包所包含的类的覆盖情况,从该包中筛选出第一覆盖率与第二覆盖率不一致的一个或多个类,并进一步对该一个或多个类所包含的方法的覆盖情况进行分析,对比并挑选出在两组代码覆盖数据中覆盖率不一致的一个或多个方法,将最终的差异代码具体到代码行。例如,经过上述分析步骤,发现在第一代代码覆盖数据中被覆盖的第10行至第20行代码,在第二代代码覆盖数据中未被覆盖到,那么,第10行至第20行代码即为两组代码覆盖数据之间的



差异代码。

[0045] 至此,已定位了第一代码覆盖数据及第二代码覆盖数据之间的差异代码。对所述差异代码进行分析,确定差异代码对应的应用场景及操作流程或者功能模块,若已有的测试用例集中包含该应用场景对应的测试用例,则对该测试用例进行调整,若已有的测试用例集中不包含该应用场景对应的测试用例,则新增该应用场景对应的测试用例,并更新初步确定的测试用例集。

[0046] 继续接收执行测试用例的指令,从所述更新后的测试用例集中调取并执行新增的或者调整后的测试用例,Jacoco代码覆盖率工具的客户端记录第三代码覆盖数据。作为一种实施方式,利用第三代码覆盖数据对第二代码覆盖数据进行更新得到更新后的第二代码覆盖数据,具体地,从第二代码覆盖数据中抽取未被覆盖的差异代码,且对与第三代码覆盖数据中被覆盖的代码一致的差异代码的覆盖状态进行更新,即,由未被覆盖的状态更新为被覆盖状态。

[0047] 进一步地,获取更新后的第二代码覆盖数据中该包的覆盖率,计算各包在第一代码覆盖数据和第二代码覆盖数据中的覆盖率的差异度,当各包的差异度均小于或等于预设阈值时,说明测试用例不需要再调整,否则,继续执行上述确定差异代码、调整测试用例的步骤。

[0048] 可以理解的是,第二代码覆盖数据与第一代码覆盖数据之间各包的差异度越小,则测试用例覆盖的范围越接近用户的真实操作情况,测试人员可根据需求调整差异度预设阈值的大小。在其他实施例中,当所述差异度的预设阈值为0时,也就是对第三代码覆盖数据被覆盖的代码与上述差异代码进行对比,当第三代码覆盖数据中被覆盖的代码完全包含上述差异代码,在对第二代码覆盖数据进行更新后,计算得到的第一代码覆盖数据及第二代码覆盖数据的差异度则为0,说明新增或者调整后的测试用例范围已经完全覆盖了上述步骤中未被覆盖的应用场景及操作流程。通过该步骤可使测试用例的范围完全包含用户在系统实际运行环境下的行为,提高测试的准确性。

[0049] 上述实施例提出的服务器1,根据用户在系统的运行环境下执行操作行为得到的第一代码覆盖数据,初步确定测试用例范围,缩小了测试用例的有效,提高测试用例范围确定的效率;并在系统测试环境下执行测试用例得到第二代码覆盖数据,对比两组代码覆盖数据得到差异代码,根据差异代码调整测试用例,确定最终的测试用例范围,使测试用例范围最大限度地接近用户实际的操作行为,提高测试效率。

[0050] 可选地,在其他的实施例中,测试用例范围确定程序10还可以被分割为一个或者多个模块,一个或者多个模块被存储于存储器11中,并由一个或多个处理器(本实施例为处理器12)所执行,以完成本发明,本发明所称的模块是指能够完成特定功能的一系列计算机程序指令段。例如,参照图2所示,为图1中测试用例范围确定程序10的程序模块示意图,该实施例中,测试用例范围确定程序10可以被分割为第一记录模块110、第二记录模块120、分析模块130、更新模块140及确定模块150,所述模块110-150所实现的功能或操作步骤均与上文类似,此处不再详述,示例性地,例如其中:

[0051] 第一记录模块110,用于记录用户在系统的运行环境下执行操作行为时的第一代码覆盖数据,分析所述第一代码覆盖数据推导出用户的操作行为,并初步确定测试用例集;

[0052] 第二记录模块120,用于接收执行所述测试用例集的指令,记录在系统的测试环境

下执行所述测试用例集时的第二代码覆盖数据；

[0053] 分析模块130,用于对所述第一代码覆盖数据及所述第二代码覆盖数据进行分析,得到所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异,并判断所述差异是否满足预设条件；

[0054] 更新模块140,用于当所述第一代代码覆盖数据及所述第二代码覆盖数据之间的差异不满足预设条件时,定位所述第一代代码覆盖数据及所述第二代码覆盖数据之间的差异代码,根据所述差异代码调整所述测试用例集,执行所述调整后的测试用例集,得到更新后的第二代码覆盖数据；及

[0055] 确定模块150,用于当所述第一代代码覆盖数据及所述第二代码覆盖数据之间的差异满足预设条件时,确定所述测试用例集为最终的测试用例集。

[0056] 此外,本发明还提供一种人才管理方法。参照图3所示,为本发明人才管理方法较佳实施例的流程图。该方法可以由一个装置执行,该装置可以由软件和/或硬件实现。

[0057] 在本实施例中,人才管理方法包括步骤S1-S5:

[0058] S1、记录用户在系统的运行环境下执行操作行为时的第一代代码覆盖数据,分析所述第一代码覆盖数据推导出用户的操作行为,并初步确定测试用例集；

[0059] S2、接收执行所述测试用例集的指令,记录在系统的测试环境下执行所述测试用例集时的第二代码覆盖数据；

[0060] S3、对所述第一代码覆盖数据及所述第二代码覆盖数据进行分析,得到所述第一代码覆盖数据及所述第二代码覆盖数据之间的差异,并判断所述差异是否满足预设条件；

[0061] S4、当所述第一代代码覆盖数据及所述第二代码覆盖数据之间的差异不满足预设条件时,定位所述第一代代码覆盖数据及所述第二代码覆盖数据之间的差异代码,根据所述差异代码调整所述测试用例集,执行所述调整后的测试用例集,得到更新后的第二代码覆盖数据,流程返回至S3；及

[0062] S5、当所述第一代代码覆盖数据及所述第二代码覆盖数据之间的差异满足预设条件时,确定所述测试用例集为最终的测试用例集。

[0063] 本实施例中,以系统P为例对本发明的方案进行说明。在系统P部署代码覆盖率工具的客户端,在服务器1端部署代码覆盖率工具的服务器端,例如,Jacoco代码覆盖率工具的客户端及服务器端。具体地,Jacoco代码覆盖率工具的客户端用于记录用户在系统的实际运行环境下执行某操作行为,及在系统的测试环境下执行测试用例时,系统代码的被覆盖情况；Jacoco代码覆盖率工具的服务器端用于接收Jacoco代码覆盖率工具的客户端记录的系统代码的被覆盖情况。

[0064] 在对进行版本更新后的系统P进行测试时,需确保测试用例的范围满足测试需求。启动部署在系统P的Jacoco代码覆盖率工具的客户端,对用户在系统的实际运行环境下的操作行为进行监控。其中,操作行为包括:用户点击系统、执行系统的某个功能模块或者执行某些操作流程,Jacoco代码覆盖率工具的客户端会记录下被用户执行的功能模块或者操作流程对应的系统代码,并生成第一代代码覆盖数据。Jacoco代码覆盖率工具的客户端将生成的第一代代码覆盖数据发送至Jacoco代码覆盖率工具的服务器端,服务器1从Jacoco代码覆盖率工具的服务器端摘取第一代代码覆盖数据,并通过服务器1的显示器向测试人员展示,供测试人员针对第一代代码覆盖数据进行分析,以初步确定系统P的测试用例范围。可以理解

的是,被用户执行过的功能模块或者操作流程,其在系统P中对应的代码都会被覆盖到,反之,未被执行过的功能模块或者操作流程对应的代码则不会被覆盖。具体地,所述第一代代码覆盖数据包括:用户在系统P的实际运行环境下执行上述操作行为时,系统P的代码中被覆盖的包、类、方法及行信息。在第一代代码覆盖数据收集过程中,计算了对应的包、类、方法的第一覆盖率,以及被覆盖代码的行数占系统P的代码的总行数的第一比例。

[0065] 可以理解的是,Jacoco代码覆盖率工具的客户端在较短时间内记录用户行为生成的第一代代码覆盖数据不一定能完全代表用户在系统P执行的所有功能模块或者操作流程,为了在不浪费资源的前提下,使第一代代码覆盖数据更全面,在时间维度上对第一代代码覆盖数据进行过滤,设置预设时间区间,例如,开启Jacoco代码覆盖率工具的客户端的时间点之后两天的时间内,取该时间段内的第一代代码覆盖数据作为最终的第一代代码覆盖数据,并将其发送至Jacoco代码覆盖率工具的服务器端。

[0066] 进一步地,服务器1从Jacoco代码覆盖率工具的服务器端摘取出第一代代码覆盖数据后,对第一代代码覆盖数据中被覆盖的代码进行分析。可以理解的是,系统的功能模块或者操作流程由均代码实现,因此,根据第一代代码覆盖数据中被覆盖的代码,可以推导出用户在系统P的实际运行环境下执行了哪些功能模块或者操作流程。根据推导得出的操作流程分别确定各操作流程的场景,并进一步得到对应的测试用例,生成初步确定的测试用例集。例如,对第一代代码覆盖数据中的代码数据进行挖掘,发现MD5算法对应的代码被覆盖,sha-1算法对应的代码未被覆盖,那么可以确定用户执行的操作流程中调用了MD5算法,测试人员在设计测试用例时,则需要覆盖MD5算法的调用场景。确定测试用例集为本领域人员习知的技术手段,在此不作累述。

[0067] 接收执行测试用例集的指令,开启部署在系统P的测试环境下的Jacoco代码覆盖率工具的客户端,并在系统P的测试环境下执行上述步骤得到的测试用例集中的测试用例,记录执行测试用例时系统P的代码的被覆盖情况,作为第二代代码覆盖数据。具体地,第二代代码覆盖数据包括:在系统P的测试环境下执行测试用例时,系统P的代码中被覆盖的包、类、方法及行信息。同样,在第二代代码覆盖数据收集过程中,计算了对应的包、类、方法的第二覆盖率,以及被覆盖代码的行数占系统P的代码的总行数的第二比例。

[0068] 需要说明的是,在执行测试用例时,可能出现返回的测试结果与预期结果不一致的情况。在本方案中,强调的是用户的操作行为,重点关注的是用户执行的操作流程,因为输入的具体值本身就存在差异,而且,测试结果对系统代码的被覆盖情况不会有影响,故不会影响第二代代码覆盖数据。

[0069] 两组代码覆盖数据中被覆盖部分包含多个包,每个包中包括类,类中包括方法,方法中包括代码行。如果某包在两组代码覆盖数据中的覆盖率的差异不满足预设条件,说明该包中包含的类、类中包含的方法、方法中包含的代码行也存在差异,进一步说明初步确定的测试用例集中未覆盖到的、用户在系统P的实际运行环境下执行的操作行为,需进一步完善测试用例。

[0070] 具体地,分别读取第一代代码覆盖数据中包含的各包在第一代代码覆盖数据和第二代代码覆盖数据中的覆盖率,计算各包在第一代代码覆盖数据和第二代代码覆盖数据中的覆盖率的差异度;筛选出差异度小于预设阈值对应的包,确定差异小于预设阈值的该部分包对应的差异代码。其中,差异度的计算公式为:

[0071]  $M = (A - B) / A$

[0072] 其中,  $M$ 为某包在第一代码覆盖数据中的覆盖率与第二代码覆盖数据中的覆盖率之间的差异度,  $A$ 为该包在第一代码覆盖数据中的覆盖率,  $B$ 为该包在第二代码覆盖数据中的覆盖率。

[0073] 例如, 某包在第一代码覆盖数据中的覆盖率为40%, 在第二代码覆盖数据中的覆盖率为35%, 根据计算公式计算该包在两组代码覆盖数据中的覆盖率的差异度为5%, 假设预设阈值为3%, 差异度大于预设阈值, 则需要进一步确定该包对应的差异代码。进一步对比该包所包含的类的覆盖情况, 从该包中筛选出第一覆盖率与第二覆盖率不一致的一个或多个类, 并进一步对该一个或多个类所包含的方法的覆盖情况进行分析, 对比并挑选出在两组代码覆盖数据中覆盖率不一致的一个或多个方法, 将最终的差异代码具体到代码行。例如, 经过上述分析步骤, 发现在第一代码覆盖数据中被覆盖的第10行至第20行代码, 在第二代码覆盖数据中未被覆盖到, 那么, 第10行至第20行代码即为两组代码覆盖数据之间的差异代码。

[0074] 至此, 已定位了第一代码覆盖数据及第二代码覆盖数据之间的差异代码。对所述差异代码进行分析, 确定差异代码对应的应用场景及操作流程或者功能模块, 若已有的测试用例集中包含该应用场景对应的测试用例, 则对该测试用例进行调整, 若已有的测试用例集中不包含该应用场景对应的测试用例, 则新增该应用场景对应的测试用例, 并更新初步确定的测试用例集。

[0075] 继续接收执行测试用例的指令, 从所述更新后的测试用例集中调取并执行新增的或者调整后的测试用例, Jacoco代码覆盖率工具的客户端记录第三代码覆盖数据。作为一种实施方式, 利用第三代码覆盖数据对第二代码覆盖数据进行更新得到更新后的第二代码覆盖数据, 具体地, 从第二代码覆盖数据中抽取未被覆盖的差异代码, 且对与第三代码覆盖数据中被覆盖的代码一致的差异代码的覆盖状态进行更新, 即, 由未被覆盖的状态更新为被覆盖状态。

[0076] 进一步地, 获取更新后的第二代码覆盖数据中该包的覆盖率, 计算各包在第一代码覆盖数据和第二代码覆盖数据中的覆盖率的差异度, 当各包的差异度均小于或等于预设阈值时, 说明测试用例不需要再调整, 否则, 继续执行上述确定差异代码、调整测试用例的步骤。

[0077] 可以理解的是, 第二代码覆盖数据与第一代码覆盖数据之间各包的差异度越小, 则测试用例覆盖的范围越接近用户的真实操作情况, 测试人员可根据需求调整差异度预设阈值的大小。在其他实施例, 当所述差异度的预设阈值为0时, 也就是对第三代码覆盖数据被覆盖的代码与上述差异代码进行对比, 当第三代码覆盖数据中被覆盖的代码完全包含上述差异代码, 在对第二代码覆盖数据进行更新后, 计算得到的第一代码覆盖数据及第二代码覆盖数据的差异度则为0, 说明新增或者调整后的测试用例范围已经完全覆盖了上述步骤中未被覆盖的应用场景及操作流程。通过该步骤可使测试用例的范围完全包含用户在系统实际运行环境下的行为, 提高测试的准确性。

[0078] 在其他实施例, 对于第一代码覆盖数据中未被覆盖的系统代码, 例如, sha-1算法对应的代码, 分析sha-1算法未被调用的原因, 若sha-1算法对应的这段代码无可操作场景, 或者用户没有执行该类操作, 可将sha-1算法对应的未被覆盖的代码作为冗余代码, 将

其从系统代码中剔除,从而精简系统代码体积。

[0079] 上述实施例提出的测试用例范围确定方法,根据用户在系统的运行环境下执行操作行为得到的第一代覆盖数据,初步确定测试用例范围,缩小了测试用例的有效,提高测试用例范围确定的效率;并在系统测试环境下执行测试用例得到第二代覆盖数据,对比两组代码覆盖数据得到差异代码,根据差异代码调整测试用例,确定最终的测试用例范围,使测试用例范围最大限度地接近用户实际的操作行为,提高测试效率。。

[0080] 此外,本发明实施例还提出一种计算机可读存储介质,所述计算机可读存储介质上存储有测试用例范围确定程序10,该程序被处理器执行时实现如下操作:

[0081] A1、记录用户在系统的运行环境下执行操作行为时的第一代覆盖数据,分析所述第一代覆盖数据推导出用户的操作行为,并初步确定测试用例集;

[0082] A2、接收执行所述测试用例集的指令,记录在系统的测试环境下执行所述测试用例集时的第二代覆盖数据;

[0083] A3、对所述第一代覆盖数据及所述第二代覆盖数据进行分析,得到所述第一代覆盖数据及所述第二代覆盖数据之间的差异,并判断所述差异是否满足预设条件;

[0084] A4、当所述第一代覆盖数据及所述第二代覆盖数据之间的差异不满足预设条件时,定位所述第一代覆盖数据及所述第二代覆盖数据之间的差异代码,根据所述差异代码调整所述测试用例集,执行所述调整后的测试用例集,得到更新后的第二代覆盖数据,流程返回至A3;及

[0085] A5、当所述第一代覆盖数据及所述第二代覆盖数据之间的差异满足预设条件时,确定所述测试用例集为最终的测试用例集。

[0086] 本发明计算机可读存储介质具体实施方式与上述测试用例范围确定方法的具体实施方式基本相同,在此不作累述。

[0087] 需要说明的是,上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。并且本文中的术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、装置、物品或者方法不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、装置、物品或者方法所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括该要素的过程、装置、物品或者方法中还存在另外的相同要素。

[0088] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在如上所述的一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算机,服务器,或者网络设备等)执行本发明各个实施例所述的方法。

[0089] 以上仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

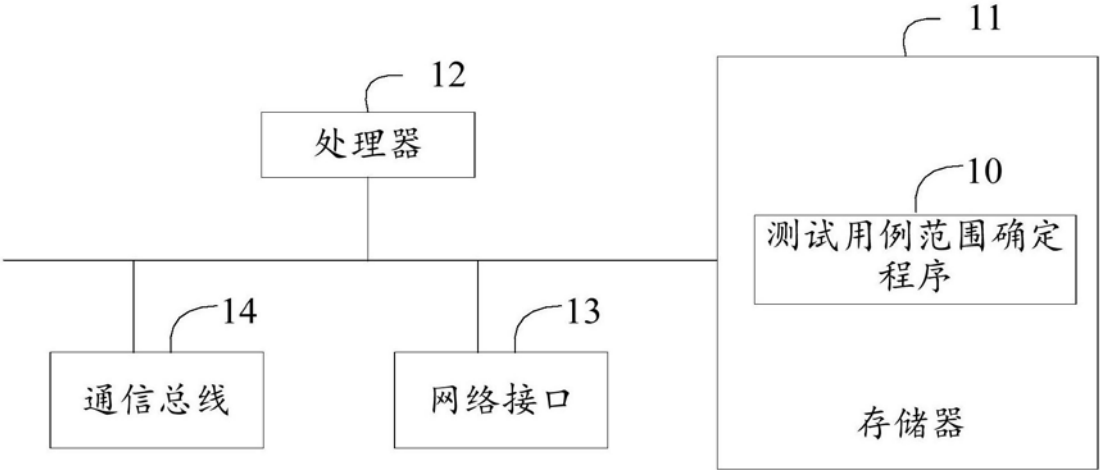


图1



图2

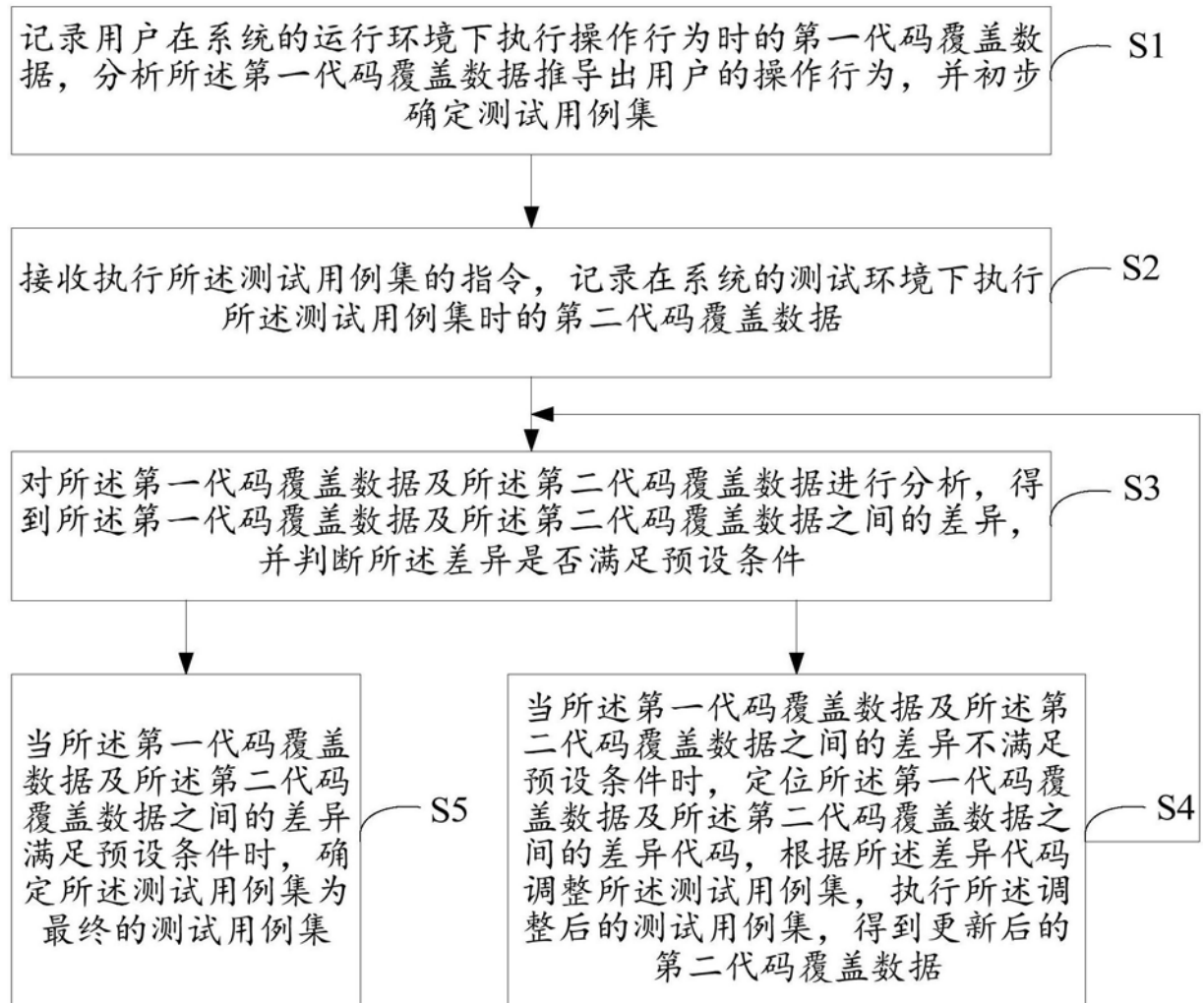


图3