



(12) 发明专利申请

(10) 申请公布号 CN 117130903 A

(43) 申请公布日 2023. 11. 28

(21) 申请号 202310388997.1

(22) 申请日 2023.04.12

(71) 申请人 中国工商银行股份有限公司

地址 100140 北京市西城区复兴门内大街  
55号

(72) 发明人 侯文龙 杨洋 陈溪 刘康婷

(74) 专利代理机构 中科专利商标代理有限责任  
公司 11021

专利代理师 张琛

(51) Int. Cl.

G06F 11/36 (2006.01)

G06F 18/22 (2023.01)

G06F 8/34 (2018.01)

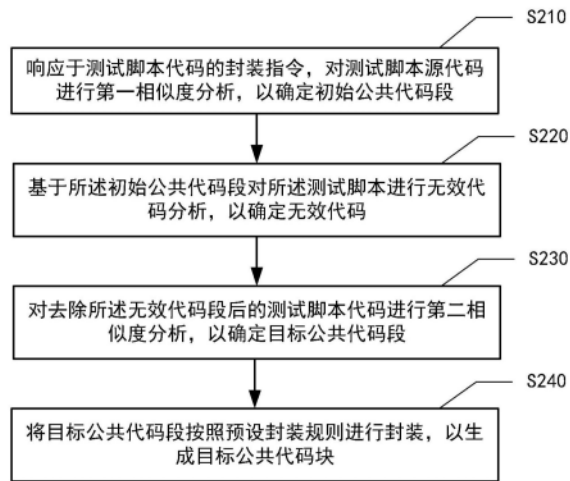
权利要求书3页 说明书19页 附图7页

(54) 发明名称

测试脚本代码的封装方法及装置

(57) 摘要

本公开提供了一种测试脚本代码的封装方法,涉及软件测试领域,可以应用于金融技术领域。该方法包括:响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段;基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码;对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共代码段;以及将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。本公开还提供了一种测试脚本代码的封装装置、设备、存储介质和程序产品。



1. 一种测试脚本代码的封装方法,其特征在于,所述方法包括:  
响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段;  
基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码;  
对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共代码段;以及  
将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。
2. 根据权利要求1所述的方法,其特征在于,所述基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码包括:  
根据初始公共代码段的执行顺序对所述测试脚本进行相似性分析,以生成测试脚本组;以及  
对所述测试脚本组中的测试脚本进行无效代码规则分析,以确定无效代码段。
3. 根据权利要求2所述的方法,其特征在于,所述根据初始公共代码段的执行顺序对所述测试脚本进行相似性分析,以生成测试脚本组包括:  
获取测试脚本中初始公共代码段的数量和执行顺序;以及  
将初始公共代码段相同数量大于第一预设阈值且所述初始公共代码段的执行顺序相同的测试脚本归类为同一测试脚本组。
4. 根据权利要求2所述的方法,其特征在于,所述对所述测试脚本组中的测试脚本进行无效代码规则分析,以确定无效代码段包括:  
获取测试脚本组中相邻公共代码段之间的代码片段;以及  
确定所述代码片段中无效操作对应的无效代码段,其中,所述无效操作包括重复点击操作、重复输入操作和无效拖拽操作。
5. 根据权利要求1所述的方法,其特征在于,所述将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块包括:  
根据所述目标公共代码段中的操作对象类型确定目标封装规则;以及  
根据所述目标封装规则对所述目标公共代码段进行封装,以生成目标公共代码块。
6. 根据权利要求5中所述的方法,其特征在于,所述根据所述公共代码中的操作对象类型确定目标封装规则包括:  
若确定所述操作对象类型为获取浏览器连接操作,则确定所述目标封装规则为在获取浏览器连接操作后增加等待和窗口最大化操作;  
若确定所述操作对象类型为点击操作,则确定所述目标封装规则为在点击操作后增加等待时间;以及  
若确定所述操作对象类型为输入操作,则确定所述目标封装规则为在输入操作前增加同元素的点击操作。
7. 根据权利要求1所述的方法,其特征在于,在生成目标公共代码块之后,还包括:  
对所述目标公共代码块的有效性进行验证;以及  
根据有效性验证结果生成公共代码块管理清单。
8. 根据权利要求7所述的方法,其特征在于,所述对所述目标公共代码块的有效性进行验证包括:

将所述目标公共代码块替换测试脚本中对应公共代码片段,以生成目标测试脚本;  
执行所述目标测试脚本,以获取所述目标测试脚本的执行成功率;  
当确定所述执行成功率大于等于预设阈值,则确定所述目标公共代码块有效;以及  
当确定所述执行成功率小于预设阈值,则确定所述目标公共代码块无效。

9. 根据权利要求1至8中任一项所述的方法,其特征在于,所述对测试脚本源代码进行第一相似度分析,以确定初始公共代码段包括:

对测试脚本源代码进行预处理,以生成目标测试脚本代码信息;

使用文本关键词抽取算法对所述目标测试脚本代码信息进行词频分析,以生成词频分析结果;以及

确定所述目标测试脚本代码信息中词频大于预设阈值的代码为初始公共代码段。

10. 根据权利要求9所述的方法,其特征在于,所述对测试脚本源代码进行预处理,以生成目标测试脚本代码信息包括:

对测试脚本源代码进行扫描,以识别所述测试脚本源代码中的测试页面操作对象;

根据所述测试页面操作对象确定所述测试页面操作对象的操作动作信息和处理方法信息;

对所述测试页面操作对象进行哈希运算,以生成哈希值;以及

根据所述测试页面操作对象信息、所述操作动作信息、所述处理方法信息和所述哈希值生成目标测试脚本代码信息。

11. 根据权利要求9所述的方法,其特征在于,所述使用文本关键词抽取算法对所述目标测试脚本代码信息进行词频分析,以生成词频分析结果包括:

根据预设步长按照脚本执行顺序对所述目标测试脚本代码进行抽取,以生成第一目标代码;

计算所述第一目标代码的词频信息;以及

根据所述词频信息生成词频分析结果。

12. 一种测试脚本代码的封装装置,其特征在于,所述装置包括:

初始公共代码段确定模块,用于响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段;

无效代码确定模块,用于基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码;

目标公共代码段确定模块,用于对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共代码段;以及

目标公共代码块生成模块,用于将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。

13. 一种电子设备,包括:

一个或多个处理器;

存储装置,用于存储一个或多个程序,

其中,当所述一个或多个程序被所述一个或多个处理器执行时,使得所述一个或多个处理器执行根据权利要求1~11中任一项所述的测试脚本代码的封装方法。

14. 一种计算机可读存储介质,其上存储有可执行指令,该指令被处理器执行时使处理

器执行根据权利要求1~11中任一项所述的测试脚本代码的封装方法。

15. 一种计算机程序产品,包括计算机程序,所述计算机程序被处理器执行时实现根据权利要求1~11中任一项所述的测试脚本代码的封装方法。

## 测试脚本代码的封装方法及装置

### 技术领域

[0001] 本公开涉及自动化测试技术领域,具体涉及页面测试技术领域,更具体地涉及一种测试脚本代码的封装方法、装置、设备、存储介质和程序产品。

### 背景技术

[0002] 在UI端自动化测试脚本编写的过程中,为了提高工作效率,测试人员通常会采用一些带脚本录制功能的自动化测试工具来进行自动化脚本编写,当测试人员开始编制脚本时,通过自动化测试工具记录测试人员在浏览器中完成操作,测试工具自动形成对应的自动化测试脚本,相较于手工逐行编写的方式,使用该方式能够大幅提升自动化测试脚本的编写效率。

[0003] 由于用户在测试的过程中,难免会出现一些误点击或者重复点击的动作,这些操作对于整个业务功能使用来说是无效操作,但因为自动化测试工具通过录制的方式自动形成对应的自动化测试脚本,会导致自动化测试脚本中也会有对应的脚本语句出现,每次自动化执行中均需要执行,浪费宝贵的执行时间和执行资源。

[0004] 需要说明的是,在上述背景技术部分公开的信息仅用于加强对本公开的背景的理解,因此可以包括不构成对本领域普通技术人员已知的现有技术的信息。

### 发明内容

[0005] 鉴于上述问题,本公开提供了提高自动化测试工作的效率的测试脚本代码的封装方法、装置、设备、介质和程序产品。

[0006] 根据本公开的第一个方面,提供了一种测试脚本公共代码的生成方法,所述方法包括:

[0007] 响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段;

[0008] 基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码;

[0009] 对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共代码段;以及

[0010] 将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。

[0011] 根据本公开的实施例,所述基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码包括:

[0012] 根据初始公共代码段的执行顺序对所述测试脚本进行相似性分析,以生成测试脚本组;以及

[0013] 对所述测试脚本组中的测试脚本进行无效代码规则分析,以确定无效代码段。

[0014] 根据本公开的实施例,所述根据初始公共代码段的执行顺序对所述测试脚本进行相似性分析,以生成测试脚本组包括:

[0015] 获取测试脚本中初始公共代码段的数量和执行顺序;以及

- [0016] 将初始公共代码段相同数量大于第一预设阈值且所述初始公共代码段的执行顺序相同的测试脚本归类为同一测试脚本组。
- [0017] 根据本公开的实施例,所述对所述测试脚本组中的测试脚本进行无效代码规则分析,以确定无效代码段包括:
- [0018] 获取测试脚本组中相邻公共代码段之间的代码片段;以及
- [0019] 确定所述代码片段中无效操作对应的无效代码段,其中,所述无效操作包括重复点击操作、重复输入操作和无效拖拽操作。
- [0020] 根据本公开的实施例,所述将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块包括:
- [0021] 根据所述目标公共代码段中的操作对象类型确定目标封装规则;以及
- [0022] 根据所述目标封装规则对所述目标公共代码段进行封装,以生成目标公共代码块。
- [0023] 根据本公开的实施例,所述根据所述公共代码中的操作对象类型确定目标封装规则包括:
- [0024] 若确定所述操作对象类型为获取浏览器连接操作,则确定所述目标封装规则为在获取浏览器连接操作后增加等待和窗口最大化操作;
- [0025] 若确定所述操作对象类型为点击操作,则确定所述目标封装规则为在点击操作后增加等待时间;以及
- [0026] 若确定所述操作对象类型为输入操作,则确定所述目标封装规则为在输入操作前增加同元素的点击操作。
- [0027] 根据本公开的实施例,在生成目标公共代码块之后,还包括:
- [0028] 对所述目标公共代码块的有效性进行验证;以及
- [0029] 根据有效性验证结果生成公共代码块管理清单。
- [0030] 根据本公开的实施例,所述对所述目标公共代码块的有效性进行验证包括:
- [0031] 将所述目标公共代码块替换测试脚本中对应公共代码片段,以生成目标测试脚本;
- [0032] 执行所述目标测试脚本,以获取所述目标测试脚本的执行成功率;
- [0033] 当确定所述执行成功率大于等于预设阈值,则确定所述目标公共代码块有效;以及
- [0034] 当确定所述执行成功率小于预设阈值,则确定所述目标公共代码块无效。
- [0035] 根据本公开的实施例,所述对测试脚本源代码进行第一相似度分析,以确定初始公共代码段包括:
- [0036] 对测试脚本源代码进行预处理,以生成目标测试脚本代码信息;
- [0037] 使用文本关键词抽取算法对所述目标测试脚本代码信息进行词频分析,以生成词频分析结果;以及
- [0038] 确定所述目标测试脚本代码信息中词频大于预设阈值的代码为初始公共代码段。
- [0039] 根据本公开的实施例,所述对测试脚本源代码进行预处理,以生成目标测试脚本代码信息包括:
- [0040] 对测试脚本源代码进行扫描,以识别所述测试脚本源代码中的测试页面操作对

象；

[0041] 根据所述测试页面操作对象确定所述测试页面操作对象的操作动作信息和处理方法信息；

[0042] 对所述测试页面操作对象进行哈希运算，以生成哈希值；以及

[0043] 根据所述测试页面操作对象信息、所述操作动作信息、所述处理方法信息和所述哈希值生成目标测试脚本代码信息。

[0044] 根据本公开的实施例，所述使用文本关键词抽取算法对所述目标测试脚本代码信息进行词频分析，以生成词频分析结果包括：

[0045] 根据预设步长按照脚本执行顺序对所述目标测试脚本代码进行抽取，以生成第一目标代码；

[0046] 计算所述第一目标代码的词频信息；以及

[0047] 根据所述词频信息生成词频分析结果。

[0048] 本公开的第二方面提供了一种测试脚本代码的封装装置，所述装置包括：

[0049] 初始公共代码段确定模块，用于响应于测试脚本代码的封装指令，对测试脚本源代码进行第一相似度分析，以确定初始公共代码段；

[0050] 无效代码确定模块，用于基于所述初始公共代码段对所述测试脚本进行无效代码分析，以确定无效代码；

[0051] 目标公共代码段确定模块，用于对去除所述无效代码段后的测试脚本代码进行第二相似度分析，以确定目标公共代码段；以及

[0052] 目标公共代码块生成模块，用于将目标公共代码段按照预设封装规则进行封装，以生成目标公共代码块。

[0053] 根据本公开的实施例，无效代码确定模块包括：测试脚本组生成子模块和无效代码分析子模块。

[0054] 测试脚本组生成子模块，用于根据初始公共代码段的执行顺序对所述测试脚本进行相似性分析，以生成测试脚本组；以及

[0055] 无效代码分析子模块，用于对所述测试脚本组中的测试脚本进行无效代码规则分析，以确定无效代码段。

[0056] 根据本公开的实施例，测试脚本组生成子模块包括：第一获取单元和分类单元。

[0057] 第一获取单元，用于获取测试脚本中初始公共代码段的数量和执行顺序；以及

[0058] 分类单元，用于将初始公共代码段相同数量大于第一预设阈值且所述初始公共代码段的执行顺序相同的测试脚本归类为同一测试脚本组。

[0059] 根据本公开的实施例，无效代码分析子模块包括：第二获取单元和确定单元。

[0060] 第二获取单元，用于获取测试脚本组中相邻公共代码段之间的代码片段；以及

[0061] 确定单元，用于确定所述代码片段中无效操作对应的无效代码段，其中，所述无效操作包括重复点击操作、重复输入操作和无效拖拽操作。

[0062] 根据本公开的实施例，目标公共代码块生成模块包括：第一确定子模块和目标公共代码块生成子模块。

[0063] 第一确定子模块，用于根据所述目标公共代码段中的操作对象类型确定目标封装规则；以及

[0064] 目标公共代码块生成子模块,用于根据所述目标封装规则对所述目标公共代码段进行封装,以生成目标公共代码块。

[0065] 根据本公开的实施例,第一确定子模块包括第一确定单元、第二确定单元和第三确定单元。

[0066] 第一确定单元,用于若确定所述操作对象类型为获取浏览器连接操作,则确定所述目标封装规则为在获取浏览器连接操作后增加等待和窗口最大化操作;

[0067] 第二确定单元,用于若确定所述操作对象类型为点击操作,则确定所述目标封装规则为在点击操作后增加等待时间;以及

[0068] 第三确定单元,用于若确定所述操作对象类型为输入操作,则确定所述目标封装规则为在输入操作前增加同元素的点击操作。

[0069] 根据本公开的实施例,还包括:验证模块和公共代码块管理清单生成模块。

[0070] 验证模块,用于对所述目标公共代码块的有效性进行验证;以及

[0071] 公共代码块管理清单生成模块,用于根据有效性验证结果生成公共代码块管理清单。

[0072] 根据本公开的实施例,公共代码块管理清单生成模块包括:目标测试脚本生成子模块、获取子模块、第二确定子模块和第三确定子模块。

[0073] 目标测试脚本生成子模块,用于将所述目标公共代码块替换测试脚本中对应公共代码片段,以生成目标测试脚本;

[0074] 获取子模块,用于执行所述目标测试脚本,以获取所述目标测试脚本的执行成功率;

[0075] 第二确定子模块,用于当确定所述执行成功率大于等于预设阈值,则确定所述目标公共代码块有效;以及

[0076] 第三确定子模块,用于当确定所述执行成功率小于预设阈值,则确定所述目标公共代码块无效。

[0077] 根据本公开的实施例,初始公共代码段确定模块包括:预处理子模块、词频分析子模块和第四确定子模块。

[0078] 预处理子模块,用于对测试脚本源代码进行预处理,以生成目标测试脚本代码信息;

[0079] 词频分析子模块,用于使用文本关键词抽取算法对所述目标测试脚本代码信息进行词频分析,以生成词频分析结果;以及

[0080] 第四确定子模块,用于确定所述目标测试脚本代码信息中词频大于预设阈值的代码为初始公共代码段。

[0081] 根据本公开的实施例,预处理子模块包括:扫描单元、第四确定单元、哈希值计算单元和目标测试脚本代码信息生成单元

[0082] 扫描单元,用于对测试脚本源代码进行扫描,以识别所述测试脚本源代码中的测试页面操作对象;

[0083] 第四确定单元,用于根据所述测试页面操作对象确定所述测试页面操作对象的操作动作信息和处理方法信息;

[0084] 哈希值计算单元,用于对所述测试页面操作对象进行哈希运算,以生成哈希值;以



及

[0085] 目标测试脚本代码信息生成单元,用于根据所述测试页面操作对象信息、所述操作动作信息、所述处理方法信息和所述哈希值生成目标测试脚本代码信息。

[0086] 根据本公开的实施例,词频分析子模块包括:第一目标代码生成子模块、计算子模块和词频分析结果生成子模块。

[0087] 第一目标代码生成子模块,用于根据预设步长按照脚本执行顺序对所述目标测试脚本代码进行抽取,以生成第一目标代码;

[0088] 计算子模块,用于计算所述第一目标代码的词频信息;以及

[0089] 词频分析结果生成子模块,用于根据所述词频信息生成词频分析结果。

[0090] 本公开的第三方面提供了一种电子设备,包括:一个或多个处理器;存储器,用于存储一个或多个程序,其中,当所述一个或多个程序被所述一个或多个处理器执行时,使得一个或多个处理器执行上述测试脚本代码的封装方法。

[0091] 本公开的第四方面还提供了一种计算机可读存储介质,其上存储有可执行指令,该指令被处理器执行时使处理器执行上述测试脚本代码的封装方法。

[0092] 本公开的第五方面还提供了一种计算机程序产品,包括计算机程序,该计算机程序被处理器执行时实现上述测试脚本代码的封装方法。

[0093] 通过本公开的实施例提供的一种测试脚本代码的封装方法,响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段;为降低无效冗余代码,在生成目标公共代码块之前对初始公共代码段进行无效代码分析,对无效代码进行自动剔除,即基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码;对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共代码段;以及将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。相比于相关技术,本公开实施例提供的方法在封装的代码片段中删除掉原有无效操作脚本语句,简化了自动化测试脚本结构,降低无效冗余代码量,同时使公共代码块最大化,避免了公共代码的碎片化,节约了测试执行资源,提高自动化脚本公共模块封装质量及执行效率。

## 附图说明

[0094] 通过以下参照附图对本公开实施例的描述,本公开的上述内容以及其他目的、特征和优点将更为清楚,在附图中:

[0095] 图1示意性示出了根据本公开实施例的测试脚本代码的封装方法、装置、设备、介质和程序产品的应用场景图;

[0096] 图2示意性示出了根据本公开实施例提供的一种测试脚本代码的封装方法的流程图;

[0097] 图3a示意性示出了根据本公开实施例提供的一种初始公共代码段的确定方法的流程图之一;

[0098] 图3b示意性示出了根据本公开实施例提供的一种初始公共代码段的确定方法的流程图之二;

[0099] 图3c示意性示出了根据本公开实施例提供的一种初始公共代码段的确定方法的流程图之三;

- [0100] 图4a示意性示出了根据本公开实施例提供的一种确定无效代码方法的流程图之一；
- [0101] 图4b示意性示出了根据本公开实施例提供的一种确定无效代码方法的流程图之二；
- [0102] 图4c示意性示出了根据本公开实施例提供的一种确定无效代码方法的流程图之三；
- [0103] 图5a示意性示出了根据本公开实施例提供的一种目标公共代码块的封装方法的流程图之一；
- [0104] 图5b示意性示出了根据本公开实施例提供的一种目标公共代码块的封装方法的流程图之二；
- [0105] 图6a示意性示出了根据本公开实施例提供的目标公共代码块验证方法的流程图之一；
- [0106] 图6b示意性示出了根据本公开实施例提供的目标公共代码块验证方法的流程图之二；
- [0107] 图7示意性示出了根据本公开实施例的一种测试脚本代码的封装装置的结构框图；以及
- [0108] 图8示意性示出了根据本公开实施例的适于实现测试脚本代码的封装方法的电子设备的方框图。

### 具体实施方式

[0109] 以下,将参照附图来描述本公开的实施例。但是应该理解,这些描述只是示例性的,而并非要限制本公开的范围。在下面的详细描述中,为便于解释,阐述了许多具体的细节以提供对本公开实施例的全面理解。然而,明显地,一个或多个实施例在没有这些具体细节的情况下也可以被实施。此外,在以下说明中,省略了对公知结构和技术的描述,以避免不必要地混淆本公开的概念。

[0110] 在此使用的术语仅仅是为了描述具体实施例,而并非意在限制本公开。在此使用的术语“包括”、“包含”等表明了所述特征、步骤、操作和/或部件的存在,但是并不排除存在或添加一个或多个其他特征、步骤、操作或部件。

[0111] 在此使用的所有术语(包括技术和科学术语)具有本领域技术人员通常所理解的含义,除非另外定义。应注意,这里使用的术语应解释为具有与本说明书的上下文相一致的含义,而不应以理想化或过于刻板的方式来解释。

[0112] 在使用类似于“A、B和C等中至少一个”这样的表述的情况下,一般来说应该按照本领域技术人员通常理解该表述的含义来予以解释(例如,“具有A、B和C中至少一个的系统”应包括但不限于单独具有A、单独具有B、单独具有C、具有A和B、具有A和C、具有B和C、和/或具有A、B、C的系统等)。

[0113] 在UI端自动化测试脚本编写的过程中,为了提高工作效率,测试人员通常会采用一些带脚本录制功能的自动化测试工具来进行自动化脚本编写,当测试人员开始编制脚本时,通过自动化测试工具记录测试人员在浏览器中完成操作,测试工具自动形成对应的自动化测试脚本,相较于手工逐行编写的方式,使用该方式能够大幅提升自动化测试脚本的

编写效率,但因为用户在测试的过程中,难免会出现一些误点击或者重复点击的动作,这些操作对于整个业务功能使用来说是无效操作,但因为自动化测试工具通过录制的方式自动形成对应的自动化测试脚本,会导致自动化测试脚本中也会有对应的脚本语句出现,每次自动化执行中均需要执行,浪费宝贵的执行时间和执行资源。

[0114] 在自动化测试脚本智能封装的过程中,采用从代码操作对象是否完全一致角度进行公共重复代码相似度的分析和封装的方案模式下,在遇到自动化测试脚本中存在误操作或者重复点击等操作的脚本语句时,传统的相似度比较方法已经无法区分,只能实现操作对象完全一致代码片段进行封装。若同一个业务操作流程在多个脚本中存在的误操作或重复点击操作较多时,会将一个业务操作流程封装碎片化,导致封装的代码片段多且散乱,不利于后续的维护保鲜。

[0115] 基于上述技术问题,本公开的实施例提供了一种测试脚本代码的封装方法,所述方法包括:响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段;基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码;对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共代码段;以及将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。

[0116] 图1示意性示出了根据本公开实施例的测试脚本代码的封装方法、装置、设备、介质和程序产品的应用场景图。

[0117] 如图1所示,根据该实施例的应用场景100可以包括页面测试脚本自动化测试场景。网络104用以在终端设备101、102、103和服务器105之间提供通信链路的介质。网络104可以包括各种连接类型,例如有线、无线通信链路或者光纤电缆等等。

[0118] 用户可以使用终端设备101、102、103通过网络104与服务器105交互,以接收或发送消息等。终端设备101、102、103上可以安装有各种通讯客户端应用,例如购物类应用、网页浏览器应用、搜索类应用、即时通信工具、邮箱客户端、社交平台软件等(仅为示例)。

[0119] 终端设备101、102、103可以是具有显示屏并且支持网页浏览的各种电子设备,包括但不限于智能手机、平板电脑、膝上型便携计算机和台式计算机等等。

[0120] 服务器105可以是为页面测试脚本管理服务器,针对用户通过应用服务器发起的测试脚本代码的封装指令,例如响应于用户利用终端设备101、102、103所发出的测试脚本代码的封装指令,该页面测试脚本管理服务器对存量测试脚本代码进行预处理,词频分析抽取出公共代码部分,对公共代码进行无效代码分析,对剔除无效代码后的公共代码并按照预设封装规则进行封装生成公共代码块。

[0121] 需要说明的是,本公开实施例所提供的测试脚本代码的封装方法一般可以由服务器105执行。相应地,本公开实施例所提供的测试脚本代码的封装装置一般可以设置于服务器105中。本公开实施例所提供的测试脚本代码的封装方法也可以由不同于服务器105且能够与终端设备101、102、103和/或服务器105通信的服务器或服务器集群执行。相应地,本公开实施例所提供的测试脚本代码的封装装置也可以设置于不同于服务器105且能够与终端设备101、102、103和/或服务器105通信的服务器或服务器集群中。

[0122] 应该理解,图1中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要,可以具有任意数目的终端设备、网络和服务器。

[0123] 需要说明的是,本公开实施例确定的测试脚本代码的封装方法和装置可用于自动

化测试技术领域,也可用于金融技术领域,还可用于除金融领域之外的任意领域,本公开实施例确定的测试脚本代码的封装方法和装置的应用领域不做限定。

[0124] 以下将基于图1描述的场景,通过图2~图6对本公开实施例的测试脚本代码的封装方法进行详细描述。

[0125] 图2示意性示出了根据本公开实施例提供的一种测试脚本代码的封装方法的流程图。如图2所示,该实施例的测试脚本代码的封装方法包括操作S210~操作S240,该方法可以由自动化测试脚本代码模块智能封装装置执行。

[0126] 在操作S210,响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段。

[0127] 一个示例中,在类似功能或者同交易类型的业务场景所对应的自动化测试脚本中,存在大量的重复代码片段;此外由于不同测试开发人员的开发习惯,导致相同功能的代码不完全一致,当该重复代码片段中对应功能发生变动时,所有相关自动化测试脚本都需要进行调整,产生了大量的重复冗余工作量。为了减少测试脚本源代码中存在的大量冗余代码,对源代码进行相似度分析,将相似的代码片段作为公共代码段封装使用,在某个系统功能发生变更,对应的自动化测试脚本代码进行保鲜维护时,大大降低测试开发人员对存量自动化测试脚本的保鲜维护成本,以及对高频使用公共模块进行模块化封装的工作成本,提高自动化测试工作的效率,降低资源投入。具体相似度分析过程可参见图3a~图3c所示出的操作步骤,在此不再赘述。

[0128] 在操作S220,基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码。

[0129] 一个示例中,为了进一步降低冗余代码,需对测试脚本中初始公共代码段之间的代码片段进行无效代码分析,确定是否存在误点击、重复点击等无效操作,确定对应的无效代码。无效代码分析过程可参见图4a~图4c所示出的操作步骤,在此不再赘述。

[0130] 在操作S230,对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共代码段。

[0131] 在操作S240,将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。

[0132] 一个示例中,在剔除无效代码后,对测试脚本代码进行第二相似度分析,确定目标公共代码段,第二相似度分析方案和技术原理与操作S210中的相似度分析过程相同,在此不再赘述。

[0133] 一个示例中,为了提高公共代码模块化的执行成功率,在确定目标公共代码段后,将操作S230中最终得到的目标公共代码段中存在的代码信息按照预设封装规则进行智能化方法封装,通过公共代码调用的方式,完成存量自动化测试脚本的模块化封装,具体封装规则和封装过程可参见图5a~图5b所示出的操作步骤,在此不再赘述。

[0134] 通过本公开的实施例提供的一种测试脚本代码的封装方法,响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段;为降低无效冗余代码,在生成目标公共代码块之前对初始公共代码段进行无效代码分析,对无效代码进行自动剔除,即基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码;对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共

代码段;以及将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。相比于相关技术,本公开实施例提供的方法在封装的代码片段中删除掉原有无效操作脚本语句,简化了自动化测试脚本结构,降低无效冗余代码量,同时使公共代码块最大化,避免了公共代码的碎片化,节约了测试执行资源,提高自动化脚本公共模块封装质量及执行效率。

[0135] 图3a示意性示出了根据本公开实施例提供的一种初始公共代码段的确定方法的流程图之一,图3b示意性示出了根据本公开实施例提供的一种初始公共代码段的确定方法的流程图之二,图3c示意性示出了根据本公开实施例提供的一种初始公共代码段的确定方法的流程图之三。

[0136] 如图3a所示,操作S210包括操作S211~操作S213。

[0137] 在操作S211,对测试脚本源代码进行预处理,以生成目标测试脚本代码信息。

[0138] 如图3b所示,操作S211包括操作S2111~操作S2114。

[0139] 在操作S2111,对测试脚本源代码进行扫描,以识别所述测试脚本源代码中的测试页面操作对象。

[0140] 在操作S2112,根据所述测试页面操作对象确定所述测试页面操作对象的操作动作信息和处理方法信息。

[0141] 在操作S2113,对所述测试页面操作对象进行哈希运算,以生成哈希值。

[0142] 在操作S2114,根据所述测试页面操作对象信息、所述操作动作信息、所述处理方法信息和所述哈希值生成目标测试脚本代码信息。

[0143] 一个示例中,扫描测试脚本代码中的测试页面操作对象,所述测试页面操作对象包括页面中的控件元素,例如可以是输入框、点击按钮等等,根据测试页面操作对象的类型结合测试脚本代码确定每一测试页面操作对象对应的操作动作信息和处理方法信息,其中处理方法信息包括判断循环语句,例如ifelse if条件判断语句。为了方便后期的词频分析统计,需对测试页面操作对象进行哈希运算得到哈希值,即同一测试页面操作对象哈希值相同,根据测试页面操作对象信息、操作动作信息、处理方法信息和哈希值生成目标测试脚本代码信息,可见下表1。

[0144] 表1目标测试脚本代码信息表

序号	操作对象信息	操作动作信息	处理方法信息	操作对象信息对应哈希值
1	http://XXX.us.XXX.com:7791/sso/login.jsp?	get		-1048494915
2	Name:ssouusername	<u>clear</u>		-1551596138
3	Name:ssouusername	sendKeys		-1551596138
4	Name:password	<u>clear</u>		1017135754
5	Name:password	sendKeys		1017135754
6	className:submitButton	click		1909411681
7			if	
8			Else if	

[0145] [0146] 在操作S212,使用文本关键词抽取算法对所述目标测试脚本代码信息进行词频分析,以生成词频分析结果。

[0147] 在操作S213,确定所述目标测试脚本代码信息中词频大于预设阈值的代码为初始公共代码段。

[0148] 如图3c所示,操作S212包括操作S2121~操作S2123。

[0149] 在操作S2121,根据预设步长按照脚本执行顺序对所述目标测试脚本代码进行抽取,以生成第一目标代码。

[0150] 在操作S2122,计算所述第一目标代码的词频信息。

[0151] 在操作S2123,根据所述词频信息生成词频分析结果。

[0152] 一个示例中,对预处理后的自动化测试脚本进行相似度分析,使用文本关键词抽取算法,将公共代码块进行提取,封装,封装后的代码块提交给公共代码定义模块进行存放处理,状态置为待验证,并且记录对应关系。文本关键词抽取算法是对降噪后的代码进行处理,按照正向+词频+步长的方式对脚本内容进行抽取。正向是指必须按照脚本的执行顺序进行抽取,预设步长,例如可以是6行,按照脚本执行顺序抽取6行代码作为第一目标代码,计算该第一目标代码在脚本中出现的频率,得到词频信息,对词频信息进行分析,词频达到一定条件的代码即可作为公共代码进行封装。

[0153] 图4a示意性示出了根据本公开实施例提供的一种确定无效代码方法的流程图之一,图4b示意性示出了根据本公开实施例提供的一种确定无效代码方法的流程图之二,图4c示意性示出了根据本公开实施例提供的一种确定无效代码方法的流程图之三。

[0154] 如图4a所示,操作S220包括操作S221和操作S222。

[0155] 在操作S221,根据初始公共代码段的执行顺序对所述测试脚本进行相似性分析,

以生成测试脚本组。

[0156] 如图4b所示,操作S221包括操作S2211和操作S2212。

[0157] 在操作S2211,获取测试脚本中初始公共代码段的数量和执行顺序。

[0158] 在操作S2212,将初始公共代码段相同数量大于第一预设阈值且所述初始公共代码段的执行顺序相同的测试脚本归类为同一测试脚本组。

[0159] 一个示例中,对获取到的初始公共代码段和调用该初始公共代码段的脚本进行分析,每个脚本中如果存在两个以上相同的初始公共代码段且出现的顺序相同,即执行顺序相同时,记录当前每个初始公共代码段的行数以及每个脚本中相同初始公共代码段的个数,并将其标记为一类新的分组,如果两个脚本存在两个以上的初始公共代码段,且初始公共代码段出现的顺序一致,则将该类脚本归为一类自动化脚本组,可以理解的是,一个自动化脚本可以归属于多个脚本组。

[0160] 在操作S222,对所述测试脚本组中的测试脚本进行无效代码规则分析,以确定无效代码段。

[0161] 如图4c所示,操作S222包括操作S2221和操作S2222。

[0162] 在操作S2221,获取测试脚本组中相邻公共代码段之间的代码片段。

[0163] 在操作S2222,确定所述代码片段中无效操作对应的无效代码段。

[0164] 根据本公开的实施例,所述无效操作包括重复点击操作、重复输入操作和无效拖拽操作。

[0165] 一个示例中,取调用公共代码段数量达到预设阈值且当前调用数量最多的脚本组,例如公共代码段数量达到脚本总数的5%,在脚本组内,基于公共脚本两两相邻分析的原则,逐个对相邻两个公共代码块之间的自动化脚本片段进行无效代码规则分析,直至调用公共脚本的脚本组不存在两个以上的公共代码段或达不到脚本总量的5%,则无效代码分析结束。无效代码分析的具体规则包括:(1)若相邻两个公共代码块间的代码存在和公共脚本对同一个控件的点击操作,且中间无其他有效操作,则只保留一条点击操作;(2)若相邻两个公共代码块间的代码存在和公共脚本对同一个控件的输入操作,且中间无其他有效操作,则保留最后一条输入操作;(3)删除无效操作,包括对单独坐标点击操作和无效拖拽操作(例如原对象和目标对象是同一个),处理方式为将对应语句直接删除。

[0166] 图5a示意性示出了根据本公开实施例提供的一种目标公共代码块的封装方法的流程图之一。图5b示意性示出了根据本公开实施例提供的一种目标公共代码块的封装方法的流程图之二。如图5a所示,操作S240包括操作S241~操作S242。

[0167] 在操作S241,根据所述目标公共代码段中的操作对象类型确定目标封装规则。

[0168] 在操作S242,根据所述目标封装规则对所述目标公共代码段进行封装,以生成目标公共代码块。

[0169] 如图5b所示,操作S241包括操作S2411~操作S2413。

[0170] 在操作S2411,若确定所述操作对象类型为获取浏览器连接操作,则确定所述目标封装规则为在获取浏览器连接操作后增加等待和窗口最大化操作。

[0171] 在操作S2412,若确定所述操作对象类型为点击操作,则确定所述目标封装规则为在点击操作后增加等待时间。

[0172] 在操作S2413,若确定所述操作对象类型为输入操作,则确定所述目标封装规则为

在输入操作前增加同元素的点击操作。

[0173] 一个示例中,当任一操作对象对应的第一目标代码的词频达到当前脚本总数量的预设阈值时,例如脚本总数的5%,则对该第一目标代码进行封装,为了提高封装后的公共代码执行的成功率,保证测试脚本的顺利执行,需要使用预设封装规则进行封装。具体规则有:

[0174] (1) 获取浏览器连接操作(`driver.get`)后增加等待和窗口最大化操作(`driver.manage().window().maximize()`;)。模拟页面打开的操作,增加等待时间等待页面加载,当页面加载后增加窗口最大化操作。

[0175] (2) 点击操作(`click`)后增加等待时间。为了屏蔽网络差异对页面测试产生的影响,在进行点击操作后增加等待时间,等待服务器响应。

[0176] (3) 输入操作(`sendKeys`)前增加同元素的点击操作。为了保证能够成功在输入框输入信息,在输入操作前增加同元素的点击操作,确保当前输入框处于激活状态。

[0177] (4) `if`条件判断结束后增加屏幕截图操作。

[0178] 根据不同的操作对象类型确定对应的封装规则进行封装生成目标公共块。在生成目标公共代码块后需对该公共代码块进行有效性的验证。

[0179] 图6a示意性示出了根据本公开实施例提供的目标公共代码块验证方法的流程图之一。图6b示意性示出了根据本公开实施例提供的目标公共代码块验证方法的流程图之二,如图6a所示,包括操作S310和操作S320。

[0180] 在操作S310,对所述目标公共代码块的有效性进行验证。

[0181] 如图6b所示,操作S310包括操作S311~操作S314。

[0182] 在操作S311,将所述目标公共代码块替换测试脚本中对应公共代码片段,以生成目标测试脚本。

[0183] 在操作S312,执行所述目标测试脚本,以获取所述目标测试脚本的执行成功率。

[0184] 在操作S313,当确定所述执行成功率大于等于预设阈值,则确定所述目标公共代码块有效。

[0185] 在操作S314,当确定所述执行成功率小于预设阈值,则确定所述目标公共代码块无效。

[0186] 一个示例中,公共代码块有效性验证规则如下:

[0187] (1) 若80%的自动化测试脚本执行通过,证明封装成功,将公共代码块定义模块中对应的公共代码状态置为验证通过,并且将已执行通过的自动化测试脚本对应内容进行模块化封装后的版本更新到自动化测试脚本资产库中。

[0188] (2) 若80%执行不通过,重新调整代码公共片段,并且将不通过的代码片段放入不通过代码特征库中,后续比对过程中出现相应代码块时,不再进行二次封装处理。同时将脚本回退为上一个版本。

[0189] 在操作S320,根据有效性验证结果生成公共代码块管理清单。

[0190] 一个示例中,为了避免重复抽取验证,提高公共代码块的生成效率,将上述验证结果生成公共代码块管理清单,详见表2公共代码块管理清单。记录公共代码块的状态,用来存放已封装好的公共代码块,以及公共代码块及引用到该代码块的自动化测试脚本对应清单对照关系,公共代码块定义有两个来源:本装置自动化智能封装的公共模块及测试人员



手工封装的公共模块。测试人员手工封装的公共模块主要用于已知的、不适合按以上模型进行封装的部分,对于这些不能进行封装的公共代码,可以直接将对应信息完善到公共代码块定义模块中,避免重复进行筛选,导致资源浪费。

[0191] 表2公共代码块管理清单

[0192]	公共代码块	引用脚本	来源	状态
	登录公共代码	纸票电票记账信息查询 商票提示付款 票据买入查询	自动化智能封装	可用
	理财股买公共代码	企网对公理财-QXQY1901-单盾-购买 企网对公理财-19GS2204-多级-购买 企网对公理财-QXQY1901-二级-计划1-购买 企网对公理财-XTL1901-二级-计划2-赎回	自动化智能封装	可用
[0193]	调整对公客户信息公共代码	4199-通过客编查询对进行作废 4103-单位客户证件维护	自动化智能封装	待验证
	招投标报文发报公共代码	银企互联-招投标-行外中标 银企互联-招投标-行内未中标 银企互联-招投标-行内中标	手工封装	可用
	对公协议签署公共代码	受理对公支付担保支付(账户支付) 受理对公支付担保支付(余额支付)	自动化智能封装	不可用

[0194] 通过本公开实施例提供的测试脚本代码的封装方法,通过智能化地将原有自动化

测试脚本中的代码块调整为公共模块引用方式,简化自动化测试脚本结构,并且智能化形成公共代码块资产;同时能够实现对脚本中存在的误操作点击或者是重复点击等无效操作进行智能化分析,自动剔除无效误操作,在封装的代码片段中删除掉原有无效操作脚本语句。然后自动进行封装,大大简化了自动化测试脚本结构,降低无效冗余代码量,提供自动化脚本公共模块封装质量和执行效率。

[0195] 基于上述测试脚本代码的封装方法,本公开还提供了一种测试脚本代码的封装装置。以下将结合图7对该装置进行详细描述。

[0196] 图7示意性示出了根据本公开实施例的一种测试脚本代码的封装装置的结构框图。

[0197] 如图7所示,该实施例的测试脚本代码的封装装置700包括初始公共代码段确定模块710、无效代码确定模块720、目标公共代码段确定模块730和目标公共代码块生成模块740。

[0198] 初始公共代码段确定模块710用于响应于测试脚本代码的封装指令,对测试脚本源代码进行第一相似度分析,以确定初始公共代码段。在一实施例中,初始公共代码段确定模块710可以用于执行前文描述的操作S210,在此不再赘述。

[0199] 无效代码确定模块720用于基于所述初始公共代码段对所述测试脚本进行无效代码分析,以确定无效代码。在一实施例中,无效代码确定模块720可以用于执行前文描述的操作S220,在此不再赘述。

[0200] 目标公共代码段确定模块730用于对去除所述无效代码段后的测试脚本代码进行第二相似度分析,以确定目标公共代码段。在一实施例中,目标公共代码段确定模块730可以用于执行前文描述的操作S230,在此不再赘述。

[0201] 目标公共代码块生成模块740用于将目标公共代码段按照预设封装规则进行封装,以生成目标公共代码块。在一实施例中,目标公共代码块生成模块740可以用于执行前文描述的操作S230,在此不再赘述。

[0202] 根据本公开的实施例,无效代码确定模块包括:测试脚本组生成子模块和无效代码分析子模块。

[0203] 测试脚本组生成子模块,用于根据初始公共代码段的执行顺序对所述测试脚本进行相似性分析,以生成测试脚本组。在一实施例中,测试脚本组生成子模块可以用于执行前文描述的操作S221,在此不再赘述。

[0204] 无效代码分析子模块,用于对所述测试脚本组中的测试脚本进行无效代码规则分析,以确定无效代码段。在一实施例中,无效代码分析子模块可以用于执行前文描述的操作S222,在此不再赘述。

[0205] 根据本公开的实施例,测试脚本组生成子模块包括:第一获取单元和分类单元。

[0206] 第一获取单元,用于获取测试脚本中初始公共代码段的数量和执行顺序。在一实施例中,第一获取单元可以用于执行前文描述的操作S2211,在此不再赘述。

[0207] 分类单元,用于将初始公共代码段相同数量大于第一预设阈值且所述初始公共代码段的执行顺序相同的测试脚本归类为同一测试脚本组。在一实施例中,分类单元可以用于执行前文描述的操作S2212,在此不再赘述。

[0208] 根据本公开的实施例,无效代码分析子模块包括:第二获取单元和确定单元。

[0209] 第二获取单元,用于获取测试脚本组中相邻公共代码段之间的代码片段。在一实施例中,第二获取单元可以用于执行前文描述的操作S2221,在此不再赘述。

[0210] 确定单元,用于确定所述代码片段中无效操作对应的无效代码段,其中,所述无效操作包括重复点击操作、重复输入操作和无效拖拽操作。在一实施例中,确定单元可以用于执行前文描述的操作S2222,在此不再赘述。

[0211] 根据本公开的实施例,目标公共代码块生成模块包括:第一确定子模块和目标公共代码块生成子模块。

[0212] 第一确定子模块,用于根据所述目标公共代码段中的操作对象类型确定目标封装规则。在一实施例中,第一确定子模块可以用于执行前文描述的操作S241,在此不再赘述。

[0213] 目标公共代码块生成子模块,用于根据所述目标封装规则对所述目标公共代码段进行封装,以生成目标公共代码块。在一实施例中,目标公共代码块生成子模块可以用于执行前文描述的操作S242,在此不再赘述。

[0214] 根据本公开的实施例,第一确定子模块包括第一确定单元、第二确定单元和第三确定单元。

[0215] 第一确定单元,用于若确定所述操作对象类型为获取浏览器连接操作,则确定所述目标封装规则为在获取浏览器连接操作后增加等待和窗口最大化操作。在一实施例中,第一确定单元可以用于执行前文描述的操作S2411,在此不再赘述。

[0216] 第二确定单元,用于若确定所述操作对象类型为点击操作,则确定所述目标封装规则为在点击操作后增加等待时间。在一实施例中,第二确定单元可以用于执行前文描述的操作S2412,在此不再赘述。

[0217] 第三确定单元,用于若确定所述操作对象类型为输入操作,则确定所述目标封装规则为在输入操作前增加同元素的点击操作。在一实施例中,第三确定单元可以用于执行前文描述的操作S2413,在此不再赘述。

[0218] 根据本公开的实施例,还包括:验证模块和公共代码块管理清单生成模块。

[0219] 验证模块,用于对所述目标公共代码块的有效性进行验证。在一实施例中,验证模块可以用于执行前文描述的操作S310,在此不再赘述。

[0220] 公共代码块管理清单生成模块,用于根据有效性验证结果生成公共代码块管理清单。在一实施例中,公共代码块管理清单生成模块可以用于执行前文描述的操作S320,在此不再赘述。

[0221] 根据本公开的实施例,公共代码块管理清单生成模块包括:目标测试脚本生成子模块、获取子模块、第二确定子模块和第三确定子模块。

[0222] 目标测试脚本生成子模块,用于将所述目标公共代码块替换测试脚本中对应公共代码片段,以生成目标测试脚本。在一实施例中,公共代码块管理清单生成模块可以用于执行前文描述的操作S320,在此不再赘述。

[0223] 获取子模块,用于执行所述目标测试脚本,以获取所述目标测试脚本的执行成功率。在一实施例中,获取子模块可以用于执行前文描述的操作S321,在此不再赘述。

[0224] 第二确定子模块,用于当确定所述执行成功率大于等于预设阈值,则确定所述目标公共代码块有效。在一实施例中,第二确定子模块可以用于执行前文描述的操作S322,在此不再赘述。

[0225] 第三确定子模块,用于当确定所述执行成功率小于预设阈值,则确定所述目标公共代码块无效。在一实施例中,第三确定子模块可以用于执行前文描述的操作S323,在此不再赘述。

[0226] 根据本公开的实施例,初始公共代码段确定模块包括:预处理子模块、词频分析子模块和第四确定子模块。

[0227] 预处理子模块,用于对测试脚本源代码进行预处理,以生成目标测试脚本代码信息。在一实施例中,预处理子模块可以用于执行前文描述的操作S211,在此不再赘述。

[0228] 词频分析子模块,用于使用文本关键词抽取算法对所述目标测试脚本代码信息进行词频分析,以生成词频分析结果。在一实施例中,词频分析子模块可以用于执行前文描述的操作S212,在此不再赘述。

[0229] 第四确定子模块,用于确定所述目标测试脚本代码信息中词频大于预设阈值的代码为初始公共代码段。在一实施例中,第四确定子模块可以用于执行前文描述的操作S213,在此不再赘述。

[0230] 根据本公开的实施例,预处理子模块包括:扫描单元、第四确定单元、哈希值计算单元和目标测试脚本代码信息生成单元

[0231] 扫描单元,用于对测试脚本源代码进行扫描,以识别所述测试脚本源代码中的测试页面操作对象。在一实施例中,扫描单元可以用于执行前文描述的操作S2111,在此不再赘述。

[0232] 第四确定单元,用于根据所述测试页面操作对象确定所述测试页面操作对象的操作动作信息和处理方法信息。在一实施例中,第四确定单元可以用于执行前文描述的操作S2112,在此不再赘述。

[0233] 哈希值计算单元,用于对所述测试页面操作对象进行哈希运算,以生成哈希值。在一实施例中,哈希值计算单元可以用于执行前文描述的操作S2113,在此不再赘述。

[0234] 目标测试脚本代码信息生成单元,用于根据所述测试页面操作对象信息、所述操作动作信息、所述处理方法信息和所述哈希值生成目标测试脚本代码信息。在一实施例中,目标测试脚本代码信息生成单元可以用于执行前文描述的操作S2114,在此不再赘述。

[0235] 根据本公开的实施例,词频分析子模块包括:第一目标代码生成子模块、计算子模块和词频分析结果生成子模块。

[0236] 第一目标代码生成子模块,用于根据预设步长按照脚本执行顺序对所述目标测试脚本代码进行抽取,以生成第一目标代码。在一实施例中,第一目标代码生成子模块可以用于执行前文描述的操作S2121,在此不再赘述。

[0237] 计算子模块,用于计算所述第一目标代码的词频信息。在一实施例中,计算子模块可以用于执行前文描述的操作S2122,在此不再赘述。

[0238] 词频分析结果生成子模块,用于根据所述词频信息生成词频分析结果。在一实施例中,词频分析结果生成子模块可以用于执行前文描述的操作S2123,在此不再赘述。

[0239] 根据本公开的实施例,初始公共代码段确定模块710、无效代码确定模块720、目标公共代码段确定模块730和目标公共代码块生成模块740中的任意多个模块可以合并在一个模块中实现,或者其中的任意一个模块可以被拆分成多个模块。或者,这些模块中的一个或多个模块的至少部分功能可以与其他模块的至少部分功能相结合,并在一个模块中实

现。根据本公开的实施例,初始公共代码段确定模块710、无效代码确定模块720、目标公共代码段确定模块730和目标公共代码块生成模块740中的至少一个可以至少被部分地实现为硬件电路,例如现场可编程门阵列(FPGA)、可编程逻辑阵列(PLA)、片上系统、基板上的系统、封装上的系统、专用集成电路(ASIC),或可以通过对电路进行集成或封装的任何其他的合理方式等硬件或固件来实现,或以软件、硬件以及固件三种实现方式中任意一种或以其中任意几种的适当组合来实现。或者,初始公共代码段确定模块710、无效代码确定模块720、目标公共代码段确定模块730和目标公共代码块生成模块740中的至少一个可以至少被部分地实现为计算机程序模块,当该计算机程序模块被运行时,可以执行相应的功能。

[0240] 图8示意性示出了根据本公开实施例的适于实现测试脚本代码的封装方法的电子设备的方框图。

[0241] 如图8所示,根据本公开实施例的电子设备900包括处理器901,其可以根据存储在只读存储器(ROM)902中的程序或者从存储部分908加载到随机访问存储器(RAM)903中的程序而执行各种适当的动作和处理。处理器901例如可以包括通用微处理器(例如CPU)、指令集处理器和/或相关芯片组和/或专用微处理器(例如,专用集成电路(ASIC))等等。处理器901还可以包括用于缓存用途的板载存储器。处理器901可以包括用于执行根据本公开实施例的方法流程的不同动作的单一处理单元或者是多个处理单元。

[0242] 在RAM 903中,存储有电子设备900操作所需的各种程序和数据。处理器901、ROM 902以及RAM 903通过总线904彼此相连。处理器901通过执行ROM 902和/或RAM 903中的程序来执行根据本公开实施例的方法流程的各种操作。需要注意,所述程序也可以存储在除ROM 902和RAM 903以外的一个或多个存储器中。处理器901也可以通过执行存储在所述一个或多个存储器中的程序来执行根据本公开实施例的方法流程的各种操作。

[0243] 根据本公开的实施例,电子设备900还可以包括输入/输出(I/O)接口905,输入/输出(I/O)接口905也连接至总线904。电子设备900还可以包括连接至I/O接口905的以下部件中的一项或多项:包括键盘、鼠标等的输入部分906;包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分907;包括硬盘等的存储部分908;以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分909。通信部分909经由诸如因特网的网络执行通信处理。驱动器910也根据需要连接至I/O接口905。可拆卸介质911,诸如磁盘、光盘、磁光盘、半导体存储器等等,根据需要安装在驱动器910上,以便于从其上读出的计算机程序根据需要被安装入存储部分908。

[0244] 本公开还提供了一种计算机可读存储介质,该计算机可读存储介质可以是上述实施例中描述的设备/装置/系统中所包含的;也可以是单独存在,而未装配入该设备/装置/系统中。上述计算机可读存储介质承载有一个或者多个程序,当上述一个或者多个程序被执行时,实现根据本公开实施例的测试脚本代码的封装方法。

[0245] 根据本公开的实施例,计算机可读存储介质可以是非易失性的计算机可读存储介质,例如可以包括但不限于:便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本公开中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。例如,根据本公开的实施例,计算机可读存储介质可以包括上文描述的

ROM 902和/或RAM 903和/或ROM 902和RAM 903以外的一个或多个存储器。

[0246] 本公开的实施例还包括一种计算机程序产品,其包括计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。当计算机程序产品在计算机系统中运行时,该程序代码用于使计算机系统实现本公开实施例所提供的测试脚本代码的封装方法。

[0247] 在该计算机程序被处理器901执行时执行本公开实施例的系统/装置中限定的上述功能。根据本公开的实施例,上文描述的系统、装置、模块、单元等可以通过计算机程序模块来实现。

[0248] 在一种实施例中,该计算机程序可以依托于光存储器件、磁存储器件等有形存储介质。在另一种实施例中,该计算机程序也可以在网络介质上以信号的形式进行传输、分发,并通过通信部分909被下载和安装,和/或从可拆卸介质911被安装。该计算机程序包含的程序代码可以用任何适当的网络介质传输,包括但不限于:无线、有线等等,或者上述的任意合适的组合。

[0249] 在这样的实施例中,该计算机程序可以通过通信部分909从网络上被下载和安装,和/或从可拆卸介质911被安装。在该计算机程序被处理器901执行时,执行本公开实施例的系统中限定的上述功能。根据本公开的实施例,上文描述的系统、设备、装置、模块、单元等可以通过计算机程序模块来实现。

[0250] 根据本公开的实施例,可以以一种或多种程序设计语言的任意组合来编写用于执行本公开实施例提供的计算机程序的程序代码,具体地,可以利用高级过程和/或面向对象的编程语言、和/或汇编/机器语言来实施这些计算程序。程序设计语言包括但不限于诸如Java,C++,python,“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算设备上执行、部分地在用户设备上执行、部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。在涉及远程计算设备的情形中,远程计算设备可以通过任意种类的网络,包括局域网(LAN)或广域网(WAN),连接到用户计算设备,或者,可以连接到外部计算设备(例如利用因特网服务提供商来通过因特网连接)。

[0251] 附图中的流程图和框图,图示了按照本公开各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图或流程图中的每个方框、以及框图或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0252] 本领域技术人员可以理解,本公开的各个实施例和/或权利要求中记载的特征可以进行多种组合和/或结合,即使这样的组合或结合没有明确记载于本公开中。特别地,在不脱离本公开精神和教导的情况下,本公开的各个实施例和/或权利要求中记载的特征可以进行多种组合和/或结合。所有这些组合和/或结合均落入本公开的范围。

[0253] 以上对本公开的实施例进行了描述。但是,这些实施例仅仅是为了说明的目的,而并非为了限制本公开的范围。尽管在以上分别描述了各实施例,但是这并不意味着各个实

施例中的措施不能有利地结合使用。本公开的范围由所附权利要求及其等同物限定。不脱离本公开的范围,本领域技术人员可以做出多种替代和修改,这些替代和修改都应落在本公开的范围之内。

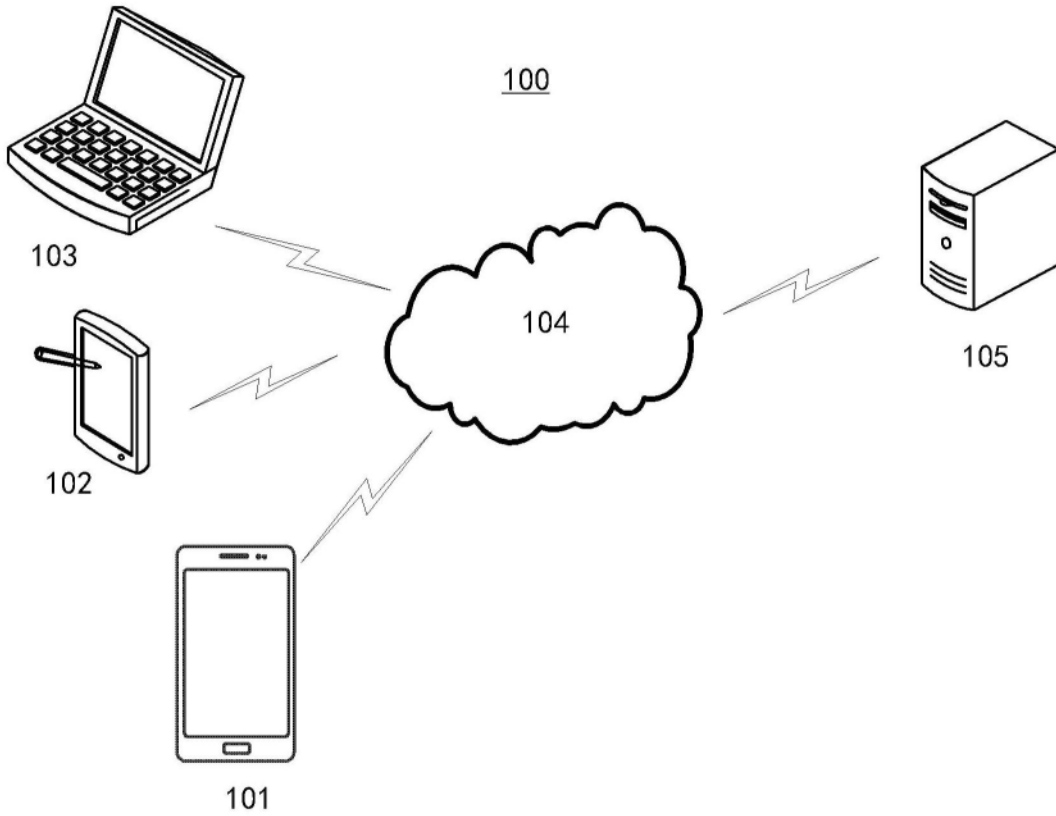


图1

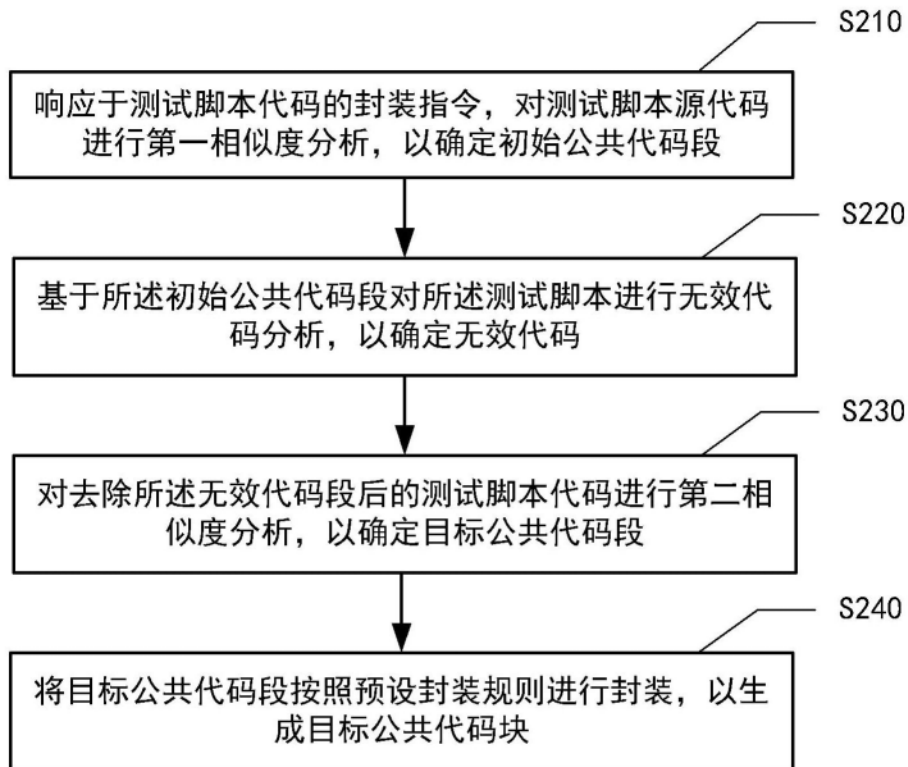


图2



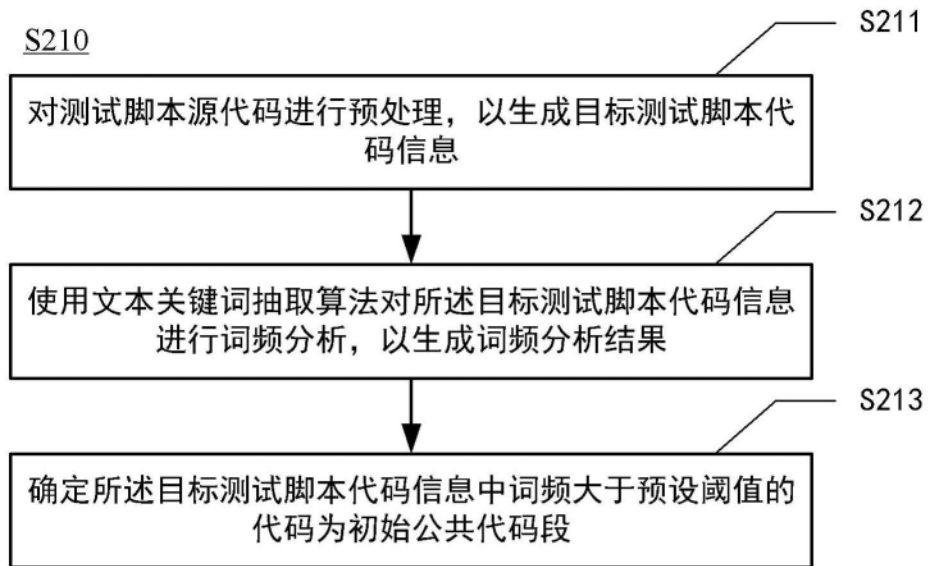


图3a

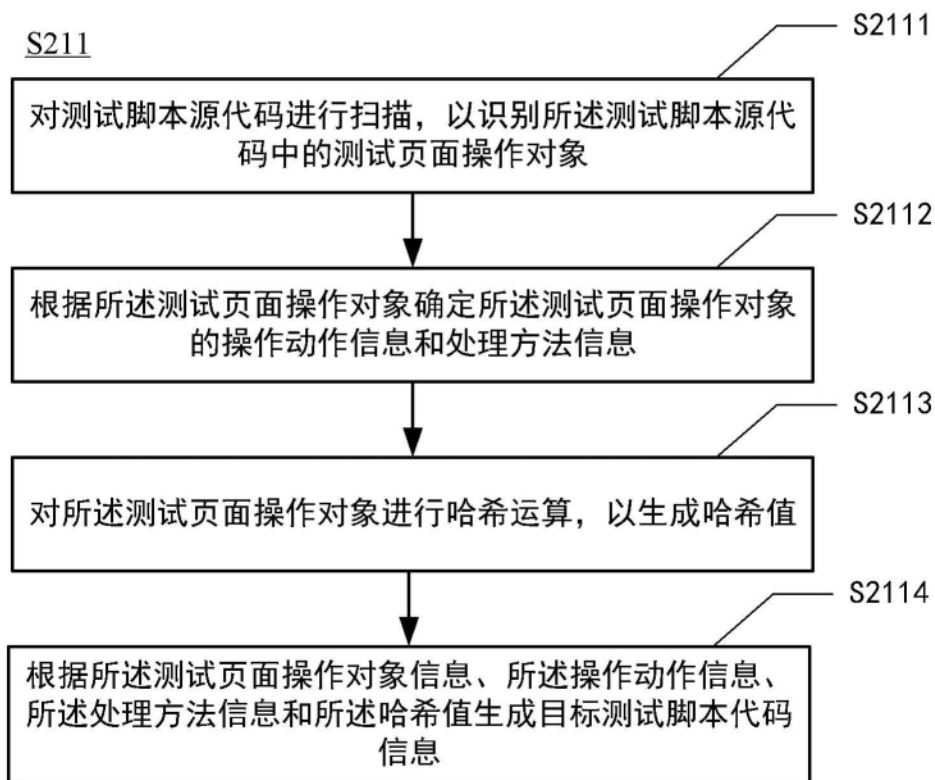


图3b

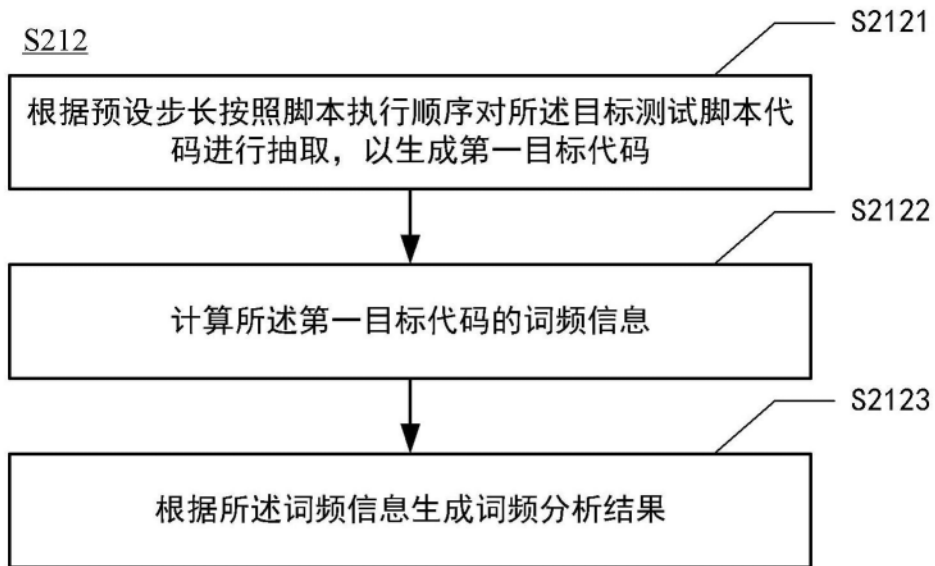


图3c

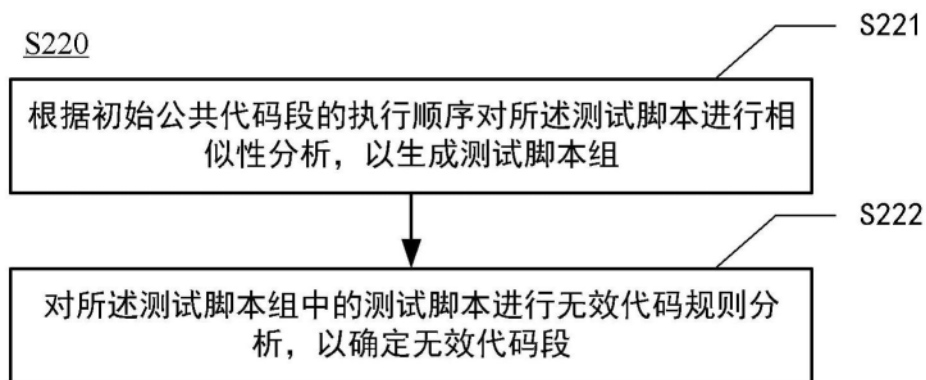


图4a

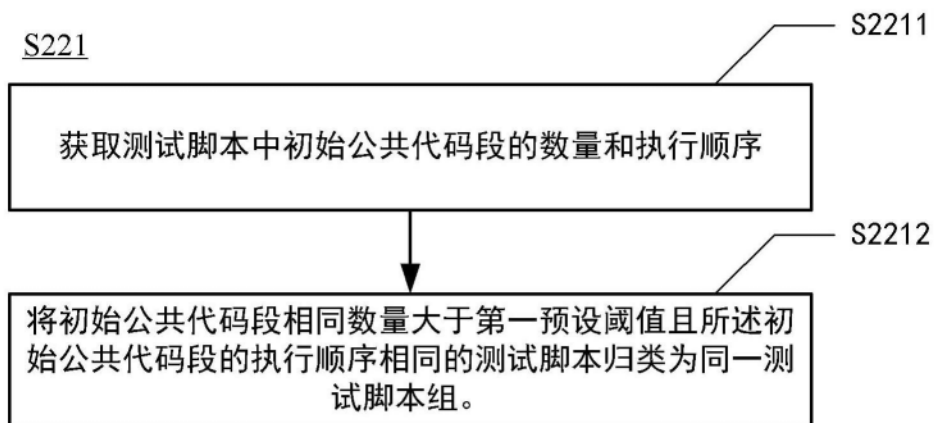


图4b

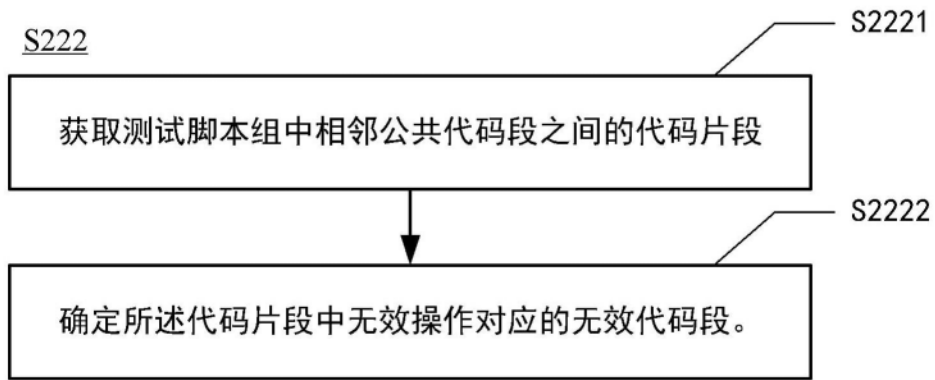


图4c

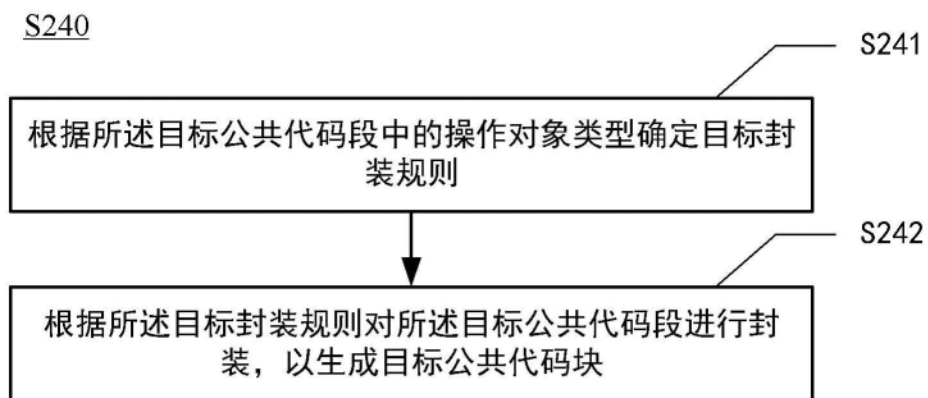


图5a

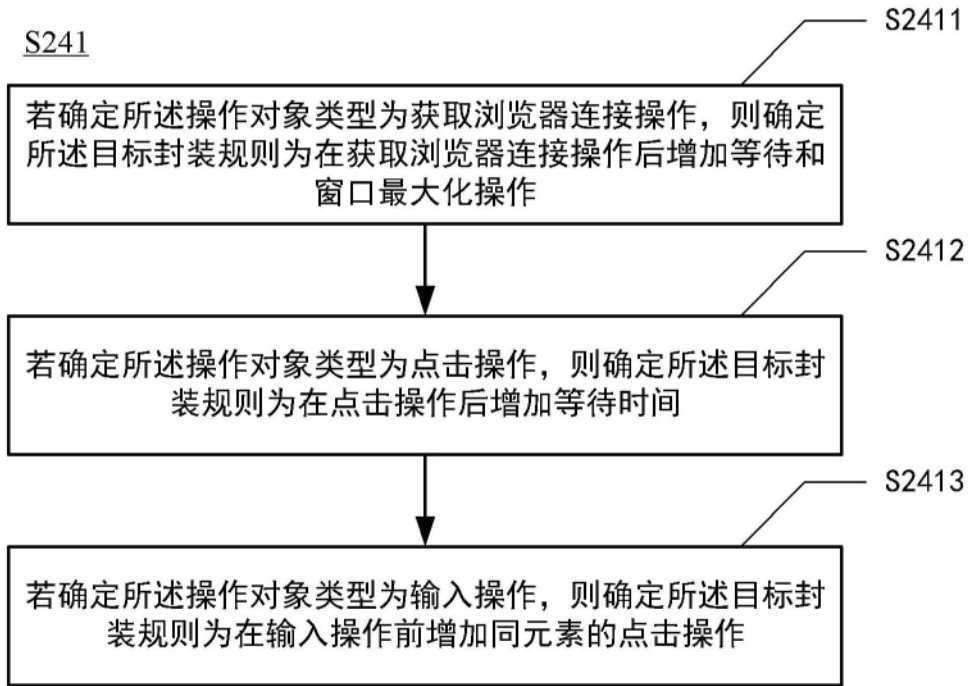


图5b

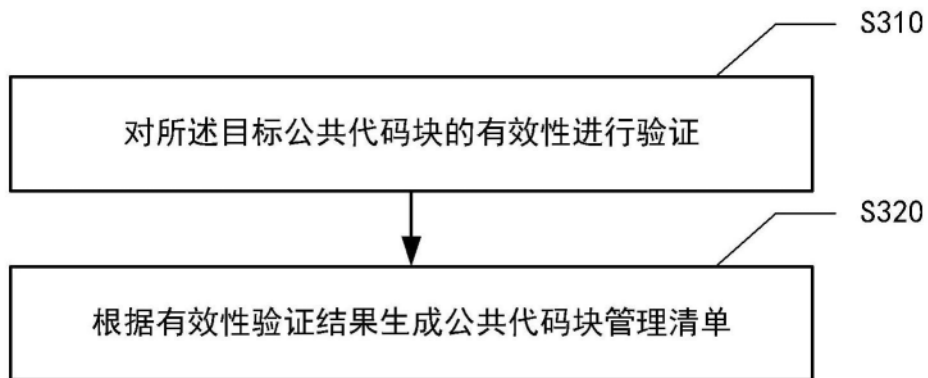


图6a

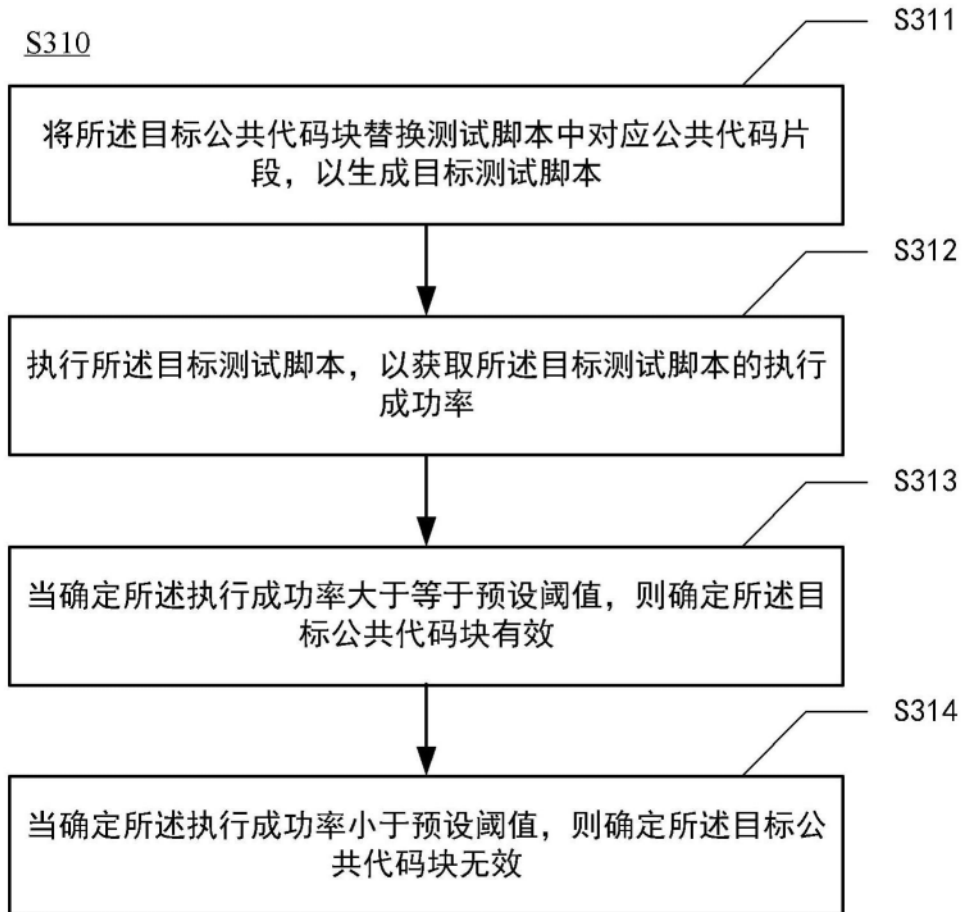


图6b

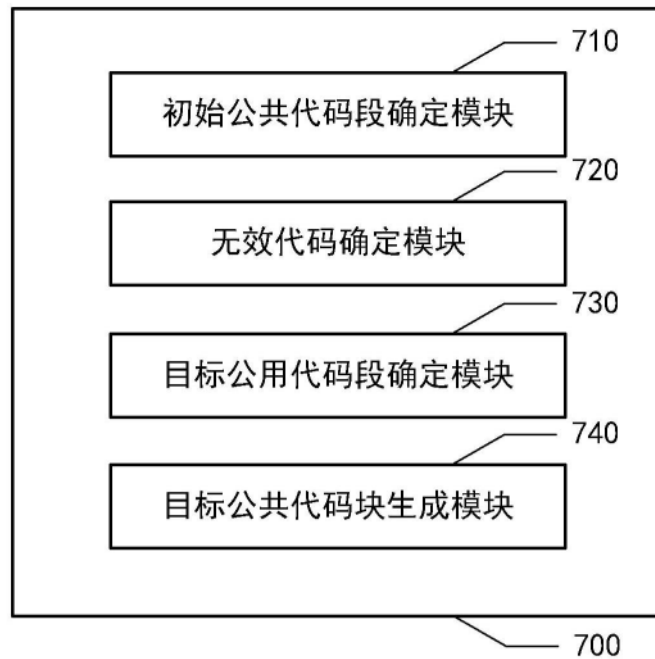


图7

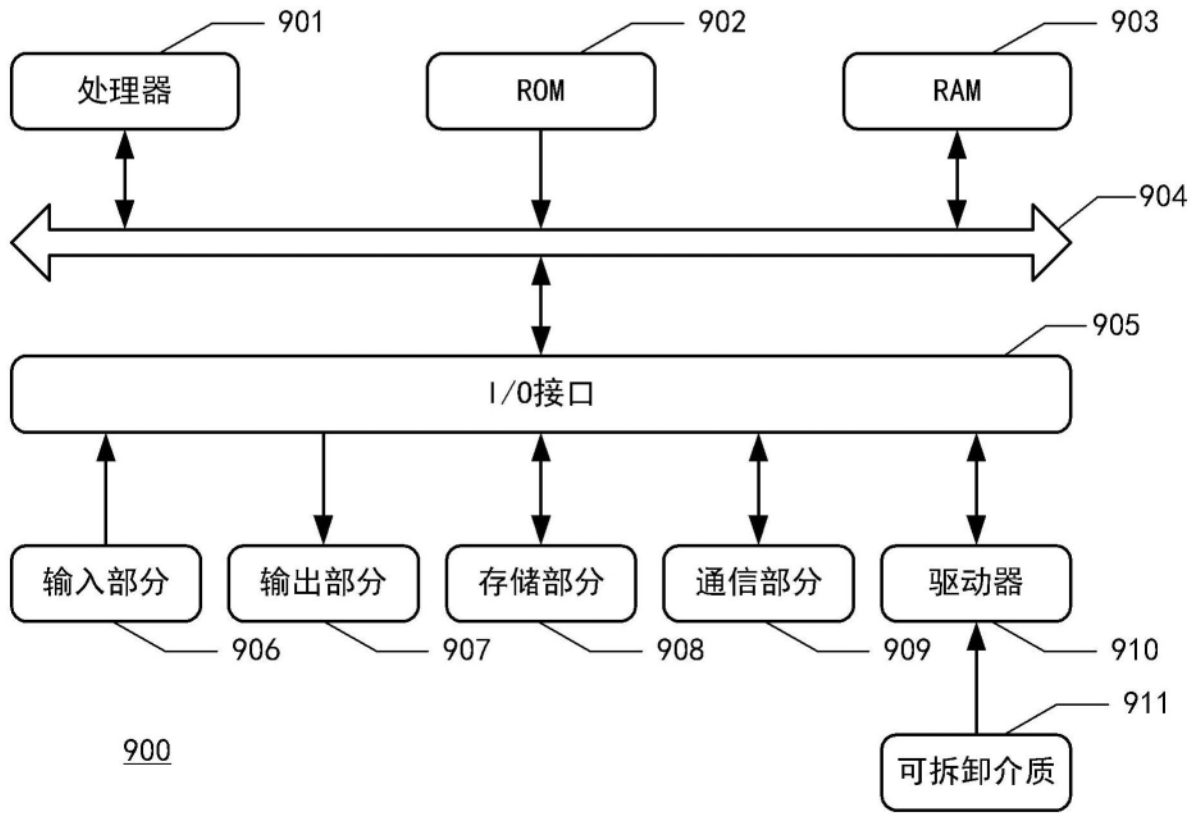


图8