



(12)发明专利申请

(10)申请公布号 CN 108459957 A

(43)申请公布日 2018.08.28

(21)申请号 201711221943.7

(22)申请日 2017.11.29

(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 伍朗 伍振亮

(74)专利代理机构 深圳众鼎专利商标代理事务
所(普通合伙) 44325

代理人 谭果林

(51)Int.Cl.

G06F 11/36(2006.01)

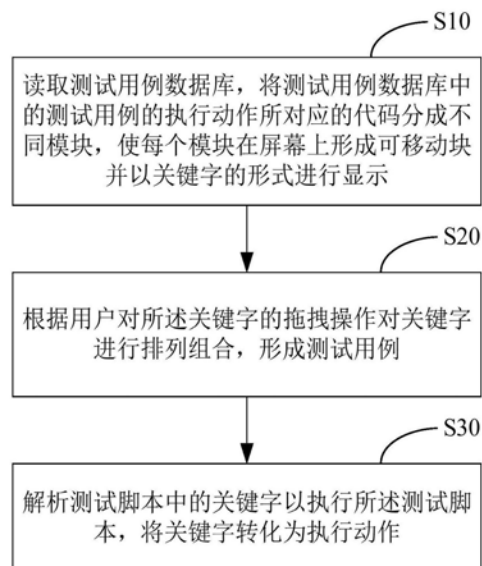
权利要求书2页 说明书9页 附图7页

(54)发明名称

基金系统测试用例的测试方法、装置、设备
及存储介质

(57)摘要

本发明涉及自动化测试技术领域,本发明提供一种基金系统测试用例的测试方法、装置、设备及存储介质,基金系统测试用例的测试方法包括:读取测试用例数据库,将测试用例数据库中的测试用例的执行动作所对应的代码分成不同模块,使每个模块在可视化界面上形成可移动块并以关键字的形式进行显示;根据用户对关键字的拖拽操作对关键字进行排列组合形成测试脚本;解析测试脚本中的关键字以执行所述测试脚本,将关键字转化为执行动作,在下层通过对这些关键字的解释,将其转化为特定的动作,让基金行业测试人员能够快速使用的脚本语言,大大降低了创建测试用例对人员编程能力的要求。



1. 一种基金系统测试用例的测试方法,其特征在于,所述基金系统测试用例的测试方法包括:

读取测试用例数据库,将所述测试用例数据库中的测试用例的执行动作所对应的代码分成不同模块,使每个模块在可视化界面上形成可移动块并以关键字的形式进行显示;

根据用户对所述关键字的拖拽操作对关键字进行排列组合形成测试脚本;

解析所述测试脚本中的关键字以执行所述测试脚本,将所述关键字转化为执行动作。

2. 如权利要求1所述的测试方法,其特征在于,根据用户对所述关键字的拖拽操作对关键字进行排列组合形成测试脚本,包括:

获取用户进行的拖拽操作,并获取所述拖拽操作的关键字对应的拖拽提示信息;

显示所述拖拽提示信息,并根据所述拖拽操作的移动轨迹移动所述拖拽提示信息,并对关键字进行排列组合形成测试脚本。

3. 如权利要求2所述的测试方法,其特征在于,根据所述拖拽操作的移动轨迹移动所述拖拽提示信息,包括:

当拖拽关键字到预设位置时,使所述拖拽提示信息显示参数提示以提示用户设置不同的参数;

当所述关键字被拖拽到的位置不符合逻辑时,发出错误拖拽提示信息进行提示。

4. 如权利要求3所述的测试方法,其特征在于,所述基金系统测试用例的测试方法还包括:

当关键字的源码文件发生修改时,将测试用例、所述源码文件以及所述源码文件中语句行对应的关键字形成映射关系;

检测源码文件中被覆盖语句行对应的关键字,将具有被覆盖语句行对应的关键字所在的测试用例确定为需要进行回归的测试用例。

5. 如权利要求1所述的测试方法,其特征在于,解析所述测试脚本中的关键字,将所述关键字转化为执行动作,包括:

获取关键字中的测试对象、执行对象以及执行参数,将所述测试对象、所述执行对象以及所述执行参数转化为执行动作。

6. 一种基金系统测试用例的测试装置,其特征在于,所述基金系统测试用例的测试装置包括:

关键字生成模块,用于读取测试用例数据库,将所述测试用例数据库中的测试用例的执行动作所对应的代码分成不同模块,使每个模块在可视化界面上形成可移动块并以关键字的形式进行显示;

测试脚本生成模块,用于根据用户对所述关键字的拖拽操作对关键字进行排列组合形成测试脚本;

测试脚本执行模块,用于解析所述测试脚本中的关键字以执行所述测试脚本,将所述关键字转化为执行动作。

7. 如权利要求6所述的测试装置,其特征在于,所述测试脚本生成模块还包括:

提示信息显示单元,用于获取用户进行的拖拽操作,并获取所述拖拽操作的关键字对应的拖拽提示信息,显示所述获取的拖拽提示信息;

提示信息移动单元,用于根据所述拖拽操作的移动轨迹移动所述拖拽提示信息;

关键字组合单元,用于对关键字进行排列组合形成测试脚本。

8.如权利要求6所述的测试装置,其特征在于,所述提示信息显示单元还用于当拖拽关键字到预设位置时,使所述拖拽提示信息显示参数提示以提示用户设置不同的参数,并当所述关键字被拖拽到的位置不符合逻辑时,发出错误拖拽提示信息进行提示。

9.一种终端设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至5任一项所述方法的步骤。

10.一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至5任一项所述方法的步骤。

基金系统测试用例的测试方法、装置、设备及存储介质

技术领域

[0001] 本发明涉及自动化测试技术领域,尤其涉及一种基金系统测试用例的测试方法、装置、设备及存储介质。

背景技术

[0002] 目前,基金系统自动化测试中使用测试脚本进行测试,需要根据测试脚本语言进行编程,尤其是测试过程中需要不断变更测试脚本,又要测试人员重新编程,而测试脚本的语言规则复杂,基金系统的业务人员和测试人员不容易理解,对测试人员的编程能力要求较高,导致工作效率较低。

发明内容

[0003] 本发明的目的在于提供一种基金系统测试用例的测试方法、装置、设备及存储介质,能够用拖拽的方式实现测试用例的创建,并且实现测试脚本的自动化测试,使创建测试用例不再受人员编程能力的影响。

[0004] 本发明是这样实现的,本发明第一方面提供一种基金系统测试用例的测试方法,所述基金系统测试用例的测试方法包括:

[0005] 读取测试用例数据库,将所述测试用例数据库中的测试用例的执行动作所对应的代码分成不同模块,使每个模块在可视化界面上形成可移动块并以关键字的形式进行显示;

[0006] 根据用户对所述关键字的拖拽操作对关键字进行排列组合形成测试脚本;

[0007] 解析所述测试脚本中的关键字以执行所述测试脚本,将所述关键字转化为执行动作。

[0008] 本发明第二方面提供一种基金系统测试用例的测试装置,所述基金系统测试用例的测试装置包括:

[0009] 关键字生成模块,用于读取测试用例数据库,将所述测试用例数据库中的测试用例的执行动作所对应的代码分成不同模块,使每个模块在可视化界面上形成可移动块并以关键字的形式进行显示;

[0010] 测试脚本生成模块,用于根据用户对所述关键字的拖拽操作对关键字进行排列组合形成测试脚本;

[0011] 测试脚本执行模块,用于解析所述测试脚本中的关键字以执行所述测试脚本,将所述关键字转化为执行动作。

[0012] 本发明第三方面提供一种终端设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现如本发明第一方面所述方法的步骤。

[0013] 本发明第四方面提供一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现如本发明第一方面所述方法的步骤。

[0014] 本发明实施例提供一种基金系统测试用例的测试方法、装置、设备及存储介质,将读取到的测试用例分解成不同的模块,每个模块对应关键字在可视化界面上显示,根据用户对关键字的拖拽操作对关键字进行排列组合形成不同的测试脚本,执行测试脚本并获取测试脚本中的关键字,解析关键字,并执行关键字中对应的执行动作,通过将代码封装成关键字,并使其可视化和可移动化,该语言将简单容易理解的最上层关键字暴露给基金行业测试人员使用,在下层通过对这些关键字的解释,将其转化为特定的动作,让基金行业测试人员能够快速使用的脚本语言,大大降低了创建测试用例对人员编程能力的要求。另外,该关键字可扩展性强,能够对新的动作进行很容易的扩充,语法简单,可扩展性强,便于后期维护

附图说明

[0015] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0016] 图1是本发明一种实施例提供的一种基金系统测试用例的测试方法的流程图;

[0017] 图2是本发明另一种实施例提供的一种基金系统测试用例的测试方法的流程图;

[0018] 图3是本发明一种实施例提供的一种基金系统测试用例的测试方法中的步骤S40的一种实施方式提供的具体流程图;

[0019] 图4是本发明一种实施例提供的一种基金系统测试用例的测试方法中的步骤S20的具体流程图;

[0020] 图5是本发明一种实施例提供的一种基金系统测试用例的测试方法中的步骤S202的具体流程图;

[0021] 图6是本发明另一种实施例提供的一种基金系统测试用例的测试方法的流程图;

[0022] 图7是本发明另一种实施例提供的一种基金系统测试用例的测试装置的结构示意图;

[0023] 图8是本发明另一种实施例提供的一种基金系统测试用例的测试装置中的测试脚本生成模块的结构示意图;

[0024] 图9是本发明另一种实施例提供的一种基金系统测试用例的测试装置的结构示意图;

[0025] 图10是本发明另一种实施例提供的一种基金系统测试用例的测试装置中的验证模块的结构示意图;

[0026] 图11是本发明另一种实施例提供的终端设备的结构示意图。

具体实施方式

[0027] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。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。

[0028] 为了说明本发明的技术方案,下面通过具体实施例来进行说明。

[0029] 本发明实施例提供一种基金系统测试用例的测试方法,如图1所示,该基金系统测试用例的测试方法包括:

[0030] 步骤S10. 读取测试用例数据库,将测试用例数据库中的测试用例的执行动作所对应的代码分成不同模块,使每个模块在可视化界面上形成可移动块并以关键字的形式进行显示。

[0031] 在步骤S10中,读取测试用例数据库中的测试用例,读取方式可以是连接测试用例管理工具来读取测试用例;对测试用例进行分块是将测试用例数据库中的测试用例的执行动作所对应的代码分成不同模块,分成的模块是实现一个功能的最小代码的单元,关键字是与测试用例中的操作命令对应的代码映射表中的名称。例如:

```
<style type="text/css">
```

```
body {scrollbar-face-color: #330033;
```

```
scrollbar-shadow-color: #FFFFFF;
```

```
scrollbar-highlight-color: #FFFFFF;
```

[0032] scrollbar-3dlight-color: #FFFFFF;

```
scrollbar-darkshadow-color: #FFFFFF;
```

```
scrollbar-track-color: #330033;
```

```
scrollbar-arrow-color: #FFFFFF;}
```

```
</style>
```

[0033] 上述代码是涉及滚动条的测试用例,其中<style type="text/css">body {scrollbar-face-color:#330033;}</style>,为实现一个功能的最小代码单元;每个模块生成可移动块并以关键字的形式进行显示,实现了代码可视化,可以使用关键字“滚动条表面颜色设定”显示可移动块,其下面隐藏的代码为<style type="text/css">body {scrollbar-face-color:#330033;}</style>。

[0034] 需要说明的是,关键字库根据读取到的测试用例数据库的更新而更新。例如,通过连接的测试用例管理工具,通过联网或者人工导入一批测试用例,则对关键字库进行更新。

[0035] 步骤S20. 根据用户对关键字的拖拽操作对关键字进行排列组合形成测试脚本。

[0036] 在步骤S20中,当检测到用户对关键字的拖拽操作时,例如,检测到用户用鼠标拉动某个关键字时,使该关键字按照用户拖动的地点进行移动,对于用户的拖拽行为,表面上是对关键字的重组,实际上是将按模块排列的代码的重组,是生成新的测试脚本的过程。例如,用户通过拖拽“开始、登录、上次记录、继续播放、时间设置、结束”等关键字,即可创建一个新的测试脚本。

[0037] 在步骤S20中,作为一种实施方式,完成关键字的排列后,建立测试脚本表格化模板,通过应用配置文件,实现将测试用例表格化模板与关键字的测试脚本语言语法进行绑定,对测试脚本中组成各步骤的关键字设置步骤数据列,测试用例编写人员通过在该模板中的逐行填写关键字对应步骤所需的步骤参数,来形成具有流程和步骤数据的测试用例表格,基于测试用例表格化模块和绑定关系配置模块实现测试脚本自动生成引擎模块,通过

该引擎进行解析测试用例表格,并读取其脚本语法绑定配置文件,逐行将各列内容绑定的脚本语言语法元素相关联,然后对单元格内容即测试参数进行基于正则表达式的模式识别,获取其语义,进而自动生成相应的脚本。

[0038] 步骤S30.解析所述测试脚本中的关键字以执行所述测试脚本,将关键字转化为执行动作。

[0039] 在步骤S30中,对关键字的解释可以通过语言解释器进行的,解释器读取测试脚本中的关键字,再根据相应的语法规则来识别需要执行的动作,然后再将这些动作传递给底层的操作模块,操作模块再到各个应用程序上进行实践的操作,比如打开页面、进行基金的申购以及查询数据库等等。解析关键字具体为获取关键字中的测试对象、执行对象以及执行参数,根据测试对象、执行对象以及执行参数执行动作。例如,对上述关键字“滚动条表面颜色设定”进行解析时,获取关键字中的测试对象为滚动条,执行对象可以为滚动条上的按钮、滚动滑槽、滚动条轨迹等,执行参数为所设定的具体颜色,解析关键字时还需要按照预定语法规则进行解析,例如,语法规则可以设置脚本定义为`defin`、逻辑判断为`if...else...`、循环测试为`do...while...`、`for...`、测试步骤为`Steps`、检验规则为`Rule`、检查点为`Check`、测试场景为`Scenario`等。

[0040] 本发明实施例提供一种基金系统测试用例的测试方法,将读取到的测试用例分解成不同的模块,每个模块对应关键字在可视化界面上显示,根据用户对关键字的拖拽操作对关键字进行排列组合形成不同的测试脚本,执行测试脚本并获取测试脚本中的关键字,解析关键字,执行关键字中对应的执行动作,通过将代码封装成关键字,并使其可视化和可移动化,该语言将简单容易理解的最上层关键字暴露给基金行业测试人员使用,在下层通过对这些关键字的解释,将其转化为特定的动作,让基金行业测试人员能够快速使用的脚本语言,大大降低了创建测试用例对人员编程能力的要求。另外,该关键字可扩展性强,能够对新的动作进行很容易的扩充,语法简单,可扩展性强,便于后期维护。

[0041] 进一步的,作为一种实施方式,如图2所示,步骤S30之后还包括步骤40。

[0042] 步骤S40.验证测试脚本的代码完整性,使测试脚本的代码符合自动化测试工具代码规范的代码。

[0043] 在步骤S40中,检查自动化生成的脚本是否完整和是否有语法错误。

[0044] 对于上述实施例中的步骤S40,作为一种实施方式,如图3所示,包括:

[0045] 步骤S401.添加生成的测试用例信息。

[0046] 步骤S402.在自动化测试脚本的外层添加异常处理代码,异常处理代码为`try-catch`代码。

[0047] 步骤S403.根据`try-catch`代码检查自动化测试脚本是否有语法错误,若无,则执行步骤S405,直接生成最终版本;若有,则执行步骤S404。

[0048] 具体的,如果`try`正确就会直接完成测试,如果`try`错误就会走`catch`代码,进而检测出语法是否错误。

[0049] 步骤S404.发出错误提示,使测试人员根据错误提示进行修改。

[0050] 步骤S405.直接生成最终版本。

[0051] 具体的,测试过程可以采用如下方式:

[0052] `try`

[0053] {
[0054] 有可能出现错误的自动化测试脚本代码
[0055] }
[0056] catch
[0057] {
[0058] 出错后的处理
[0059] }
[0060] 如果try中的代码没有出错,则程序正常运行try中的内容后,不会执行catch中的内容,
[0061] 如果try中的代码一旦出错,程序立即跳入catch中去执行代码,那么try中出错代码后的所有代码就不再执行了,遇到异常抛出时,将按照如下算法进行异常处理,算法的过程如下:
[0062] Begin
[0063] (1) if检查异常抛出
[0064] (2) 将线程中表示异常发生的标志位置上
[0065] (3) for取出当前函数Method的栈帧do
[0066] (4) 扫描函数中的所有的异常处理代码,找出和当前异常相匹配的异常处理块
[0067] (5) if找到do
[0068] (6) 跳出循环
[0069] (7) else
[0070] (8) 根据当前栈帧的类型进行不同策略的栈回退,找到上一个函数的栈帧 (9) if取出栈帧为中断栈帧do
[0071] (10) 将线程异常位依旧置上,抛给上层进行处理
[0072] (11) end if
[0073] (12) end for
[0074] (13) 找到了匹配的异常处理块,进行提示,并根据当前栈帧的类型,采取不同的策略更新PC值,跳转至Catch块的首指令处
[0075] (14) end if
[0076] End。
[0077] 本发明实施例可以验证自动化测试脚本的代码完整性,使自动化测试脚本的代码符合自动化测试工具代码规范的代码。
[0078] 对于上述实施例中的步骤S20,作为一种实施方式,如图4所示,根据用户对关键字的拖拽操作对关键字进行排列组合,形成不同的测试用例,包括:
[0079] 步骤S201. 获取用户进行的拖拽操作,并获取拖拽操作的关键字对应的拖拽提示信息。
[0080] 在步骤S201中,当用户对关键字进行拖动时,此时检测关键字对应的提示信息,该提示信息可以是关键字自身带有的提示信息,例如关键字的规则等等,也可以是用户在拖动关键字前自行定义的一些提示,例如关键字所属的逻辑位置等等,其中,逻辑位置是指关键字满足逻辑执行顺序的位置,逻辑顺序之间原则上是由上至下的依赖关系,例如关键字

为“登录”、“查询”以及“退出”，其逻辑位置为前端、中端及后端。

[0081] 步骤S202. 显示所述拖拽提示信息，并根据所述拖拽操作的移动轨迹移动所述拖拽提示信息，以使用户在进行所述拽操作的过程中能够查看到所述拖拽提示信息。

[0082] 在步骤S202中，当拖拽着关键字移动时，提示信息随着拖拽轨迹而移动。作为一种实施例，提示信息中文字的颜色不同于关键字的颜色，并且提示信息中文字的字体大小不同于关键字的字体大小，例如，关键字颜色和字体为黑色四号时，提示信息中的文字颜色和字体为红色小五。本实施例有利于用户时刻看到提示信息，且不与关键字相混淆。

[0083] 对于上述实施例中的步骤S202，作为一种实施方式，如图5所示，包括：

[0084] 步骤S2021. 当拖拽关键字到某些位置时，所述拖拽提示信息显示参数提示以提示用户设置不同的参数。

[0085] 在步骤S2021中，当拖拽关键字到某些位置时，为了适应该位置，需要对关键字内部代码中的某些参数进行修改，此时可以发出修改参数提示，用户可以根据需求自行设置参数。

[0086] 步骤S2022. 当关键字被拖拽到的位置不符合逻辑时，发出错误拖拽提示信息进行提示。

[0087] 在步骤S2022中，出现错误提示的主要原因是关键字下隐藏的代码逻辑按照拖拽的顺序行不通，提示用户更换位置或者更换关键字。

[0088] 本发明实施例在用户拖拽关键字时，显示关键字的提示信息，使用户根据提示信息更准确的对关键字进行组成形成测试用例，提供了用户创建测试用例的效率。

[0089] 作为另一种实施方式，当对关键字的代码进行修改时，可以对包括该关键字的测试用例进行回归测试，回归测试是指修改了关键字的旧代码后，重新对关键字的测试用例进行测试以确认修改没有引入新的错误或导致其他代码产生错误。如图6所示，该基金系统测试用例的测试方法包括：还包括：

[0090] 步骤S301. 当关键字的源码文件发生修改时，将测试用例、源码文件以及源码文件中语句行对应的关键字形成映射关系。

[0091] 步骤S302. 检测源码文件中被覆盖语句行对应的关键字，将具有被覆盖语句行对应的关键字所在的测试用例确定为需要进行回归的测试用例。

[0092] 基于步骤301和步骤S302，源码文件可以对应多个关键字，例如，查询功能的源代码根据其执行动作可以分为登录、查询以及退出关键字，其中，源码文件1的第1至10行对应登录关键字，源码文件1的第11至20行对应查询关键字，源码文件1的第21至30行对应退出关键字，测试用例1至测试用例5与不同关键字的源码文件相关，其中，针对每一测试用例得到以下映射关系为：

[0093] 1、测试用例1使用了登录关键字，即测试用例1，源码文件1、测试用例1覆盖源码文件1的第1至10行之间的映射关系；

[0094] 2、测试用例2使用了查询关键字，测试用例2、源码文件1、测试用例2覆盖源码文件1的第11至20行之间的映射关系；

[0095] 3、测试用例3使用了退出关键字，测试用例3、源码文件1、测试用例3覆盖源码文件1的第21至30行之间的映射关系；

[0096] 4、测试用例4使用了登录关键字、查询关键字以及退出关键字，测试用例4、源码文

件4、测试用例4覆盖源码文件1的第1至30行之间的映射关系。

[0097] 当修改源码文件1的第1行至10行中的代码,可以以源码文件1中第1行至10行对应的关键字查找包含该关键字的映射关系,即可得到测试用例1以及测试用例4对应的映射关系中包含该关键字,将该查找到的映射关系中的测试用例1和4确定为需要回归的用例,而测试用例2和测试用例3则不需要回归。

[0098] 在上述步骤中,对测试用例中的关键字的源码文件进行修改或者替换时,则源码文件的语句行发生修改,使测试用例与源码文件中的语句行对应的关键字一一对应,可以根据被覆盖的语句行对应的关键字找到需要回归的测试用例,相比于现有技术,能够快速、准确定位出需要回归的用例。

[0099] 本发明另一种实施例提供一种测试用例创建装置50,如图7所示,测试用例创建装置50包括:

[0100] 关键字生成模块501,用于读取测试用例数据库,将测试用例数据库中的测试用例的执行动作所对应的代码分成不同模块,每个模块在可视化界面上形成可移动块并以关键字的形式进行显示;

[0101] 测试脚本生成模块502,用于根据用户对所述关键字的拖拽操作对关键字进行排列组合形成测试脚本;

[0102] 测试脚本执行模块503,用于执行所述测试脚本并解析所述测试脚本中的关键字,将所述关键字转化为执行动作。

[0103] 进一步的,如图8所示,测试脚本生成模块502还包括:

[0104] 提示信息显示单元511,用于获取用户进行的拖拽操作,并获取所述拖拽操作的关键字对应的拖拽提示信息,显示所述获取的拖拽提示信息;

[0105] 提示信息移动单元512,用于根据所述拖拽操作的移动轨迹移动所述拖拽提示信息;

[0106] 关键字组合单元512,用于对关键字进行排列组合形成测试脚本。

[0107] 进一步的,所述提示信息显示单元511还用于当拖拽关键字到预设位置时,使所述拖拽提示信息显示参数提示以提示用户设置不同的参数,并当所述关键字被拖拽到的位置不符合逻辑时,发出错误拖拽提示信息进行提示。

[0108] 进一步的,如图9所示,测试用例创建装置50还包括验证模块504,所述验证模块504用于检查自动化生成的脚本是否完整和是否有语法错误。

[0109] 进一步的,如图10所示,验证模块504包括:

[0110] 测试用例信息添加单元514,用于添加测试用例信息,所述测试用例信息包括:测试用例名称和ID;

[0111] 异常处理代码添加单元515,用于在所述自动化测试脚本的外层添加异常处理代码,所述异常处理代码为try-catch代码;

[0112] 错误验证单元516,用于根据所述try-catch代码验证所述自动化测试脚本是否有语法和编写错误,若验证不通过,则发出错误提示,若通过,则生成最终的自动化测试脚本。

[0113] 上述终端设备中模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0114] 本发明另一种实施例提供一计算机可读存储介质,该计算机可读存储介质上存储

有计算机程序,该计算机程序被处理器执行时实现上述实施例中的基金系统测试用例的测试方法,为避免重复,这里不再赘述。或者,该计算机程序被处理器执行时实现上述实施例中基金系统测试用例的测试装置中各模块/单元的功能,为避免重复,这里不再赘述。

[0115] 图11是本实施例中终端设备的示意图。如图11所示,终端设备6包括处理器60、存储器61以及存储在存储器61中并可在处理器60上运行的计算机程序62。处理器60执行计算机程序62时实现上述实施例中测试用例创建方法的各个步骤,例如图1所示的步骤S10、S20和S30。或者,处理器60执行计算机程序62时实现上述实施例中基金系统测试用例的测试装置各模块/单元的功能。

[0116] 示例性的,计算机程序62可以被分割成一个或多个模块/单元,一个或者多个模块/单元被存储在存储器61中,并由处理器60执行,以完成本发明。一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述计算机程序62在终端设备6中的执行过程。例如,计算机程序62可以被分割成关键字生成模块501、测试用例生成模块502以及测试用例脚本生成模块503。

[0117] 该终端设备6可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。终端设备可包括,但不仅限于,处理器60、存储器61。本领域技术人员可以理解,图11仅仅是终端设备6的示例,并不构成对终端设备6的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件,例如终端设备还可以包括输入输出设备、网络接入设备、总线等。

[0118] 所称处理器60可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0119] 存储器61可以是终端设备6的内部存储单元,例如终端设备6的硬盘或内存。存储器61也可以是终端设备6的外部存储设备,例如终端设备6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61还可以既包括终端设备6的内部存储单元也包括外部存储设备。存储器61用于存储计算机程序以及终端设备6所需的其他程序和数据。存储器61还可以用于暂时地存储已经输出或者将要输出的数据。

[0120] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,仅以上述各功能单元、模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能单元、模块完成,即将所述装置的内部结构划分成不同的功能单元或模块,以完成以上描述的全部或者部分功能。

[0121] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0122] 所述集成的模块/单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明实现上述实施

例方法中的全部或部分流程,也可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一计算机可读存储介质中,该计算机程序在被处理器执行时,可实现上述各个方法实施例的步骤。其中,所述计算机程序包括计算机程序代码,所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括:能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、电载波信号、电信信号以及软件分发介质等。需要说明的是,所述计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减,例如在某些司法管辖区,根据立法和专利实践,计算机可读介质不包括是电载波信号和电信信号。

[0123] 以上所述实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围,均应包含在本发明的保护范围之内。

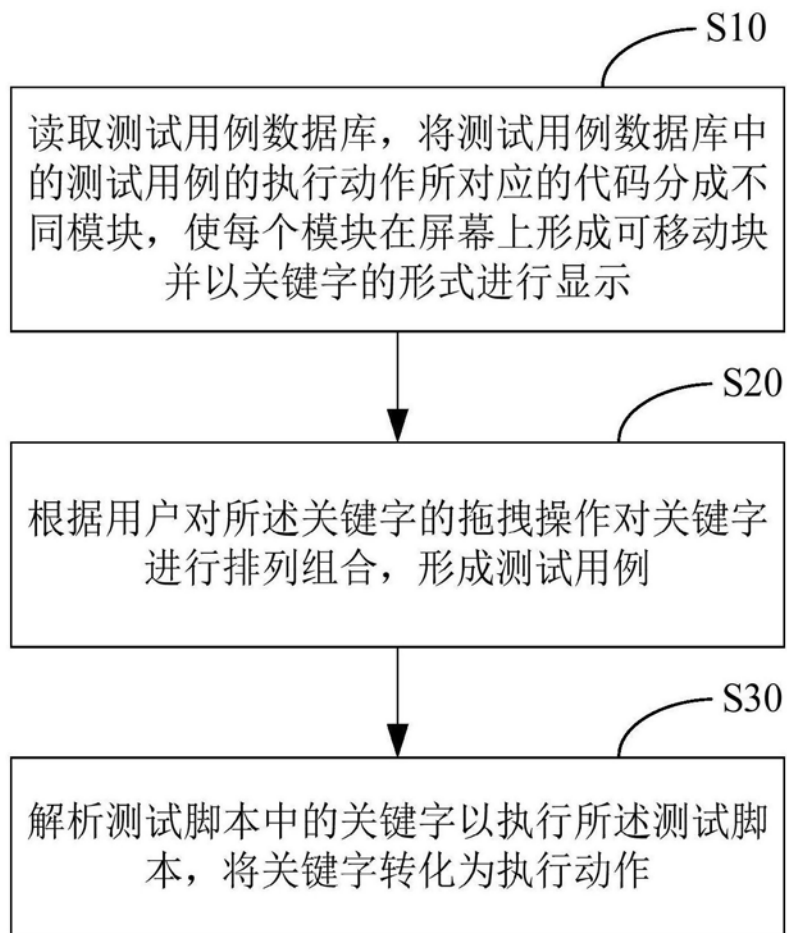


图1

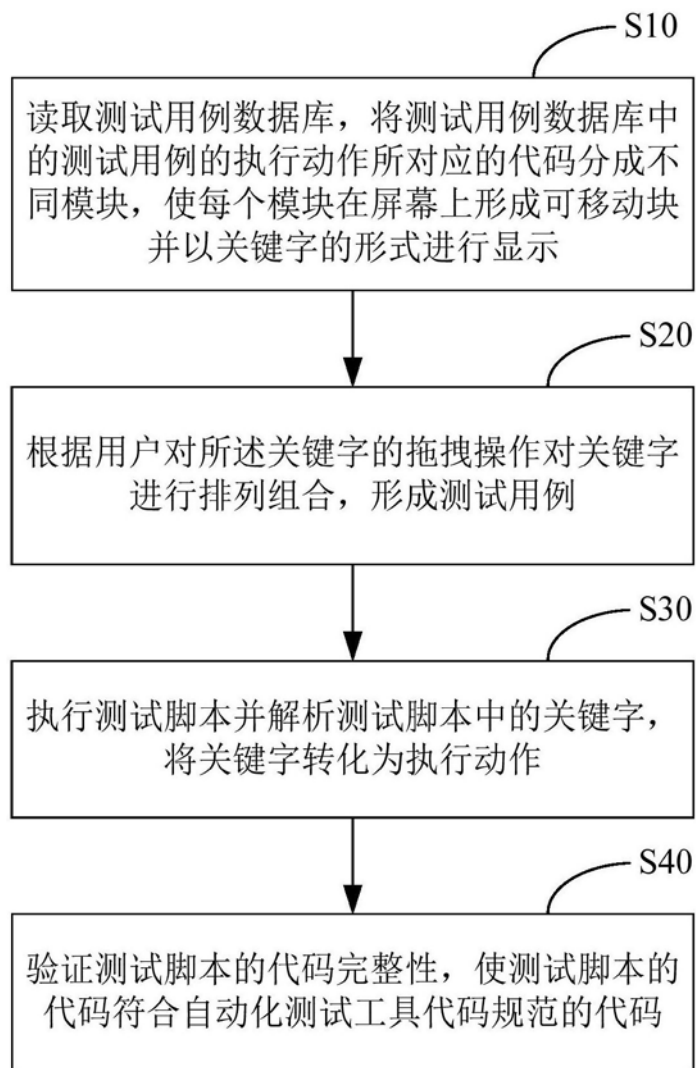


图2

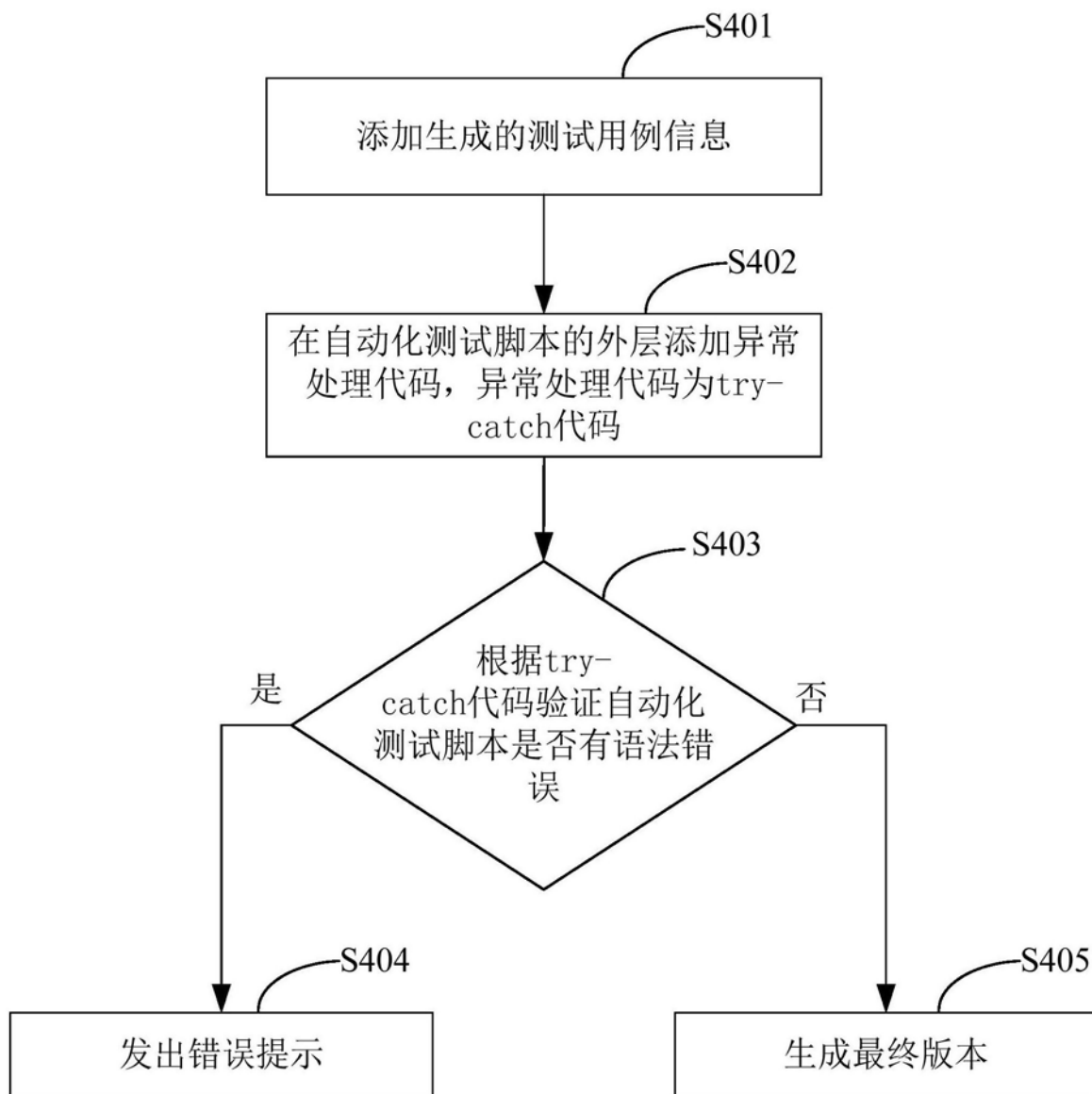


图3

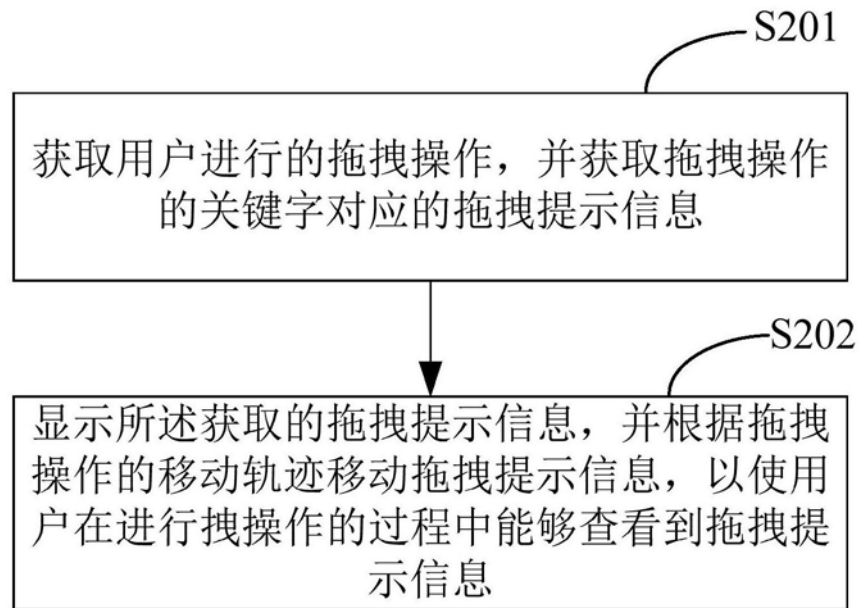


图4

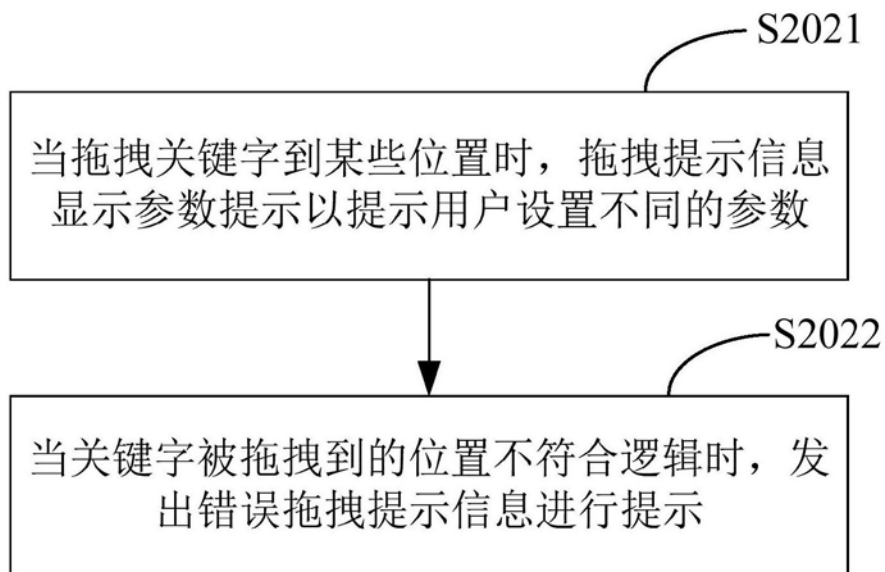


图5

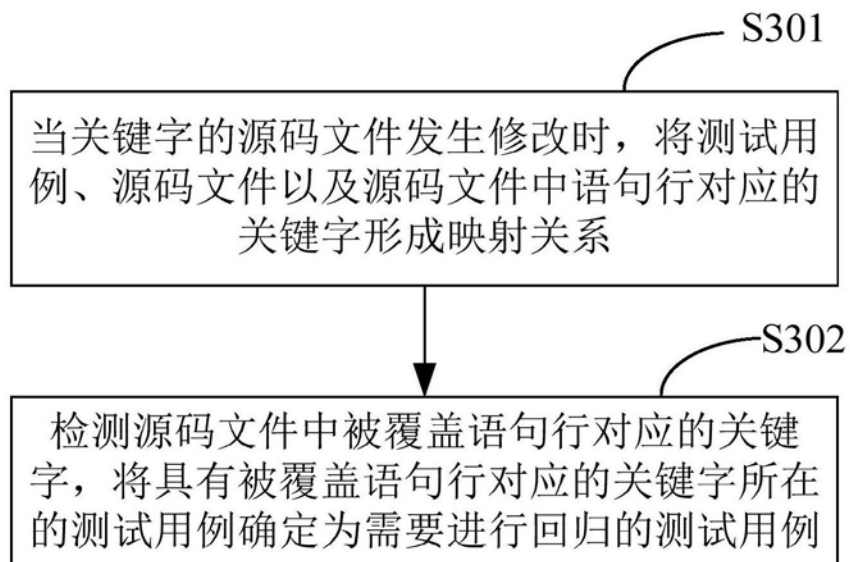


图6

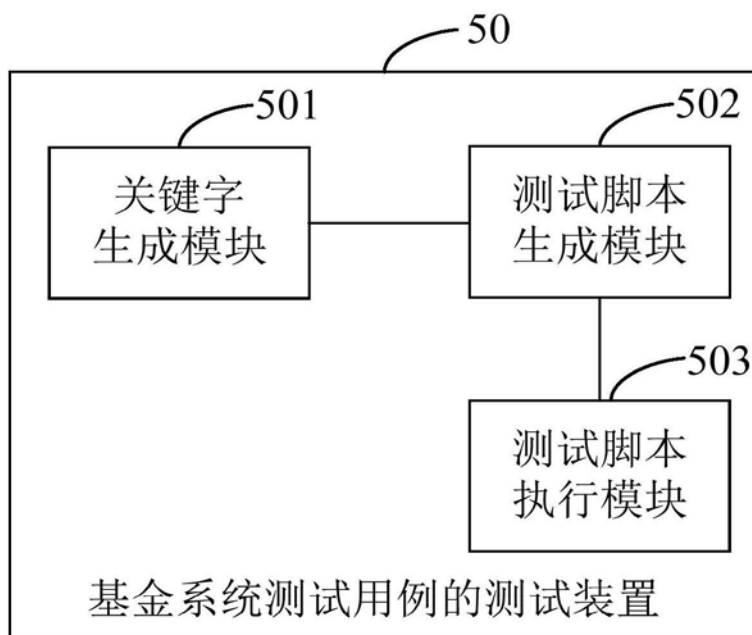


图7

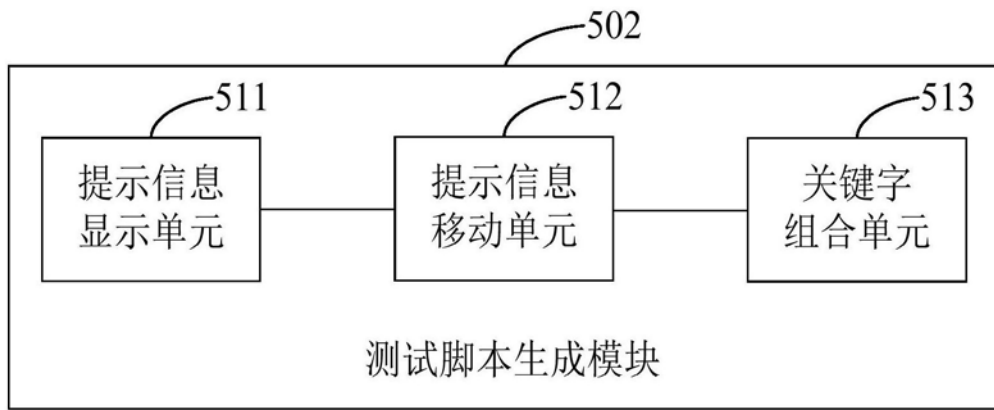


图8

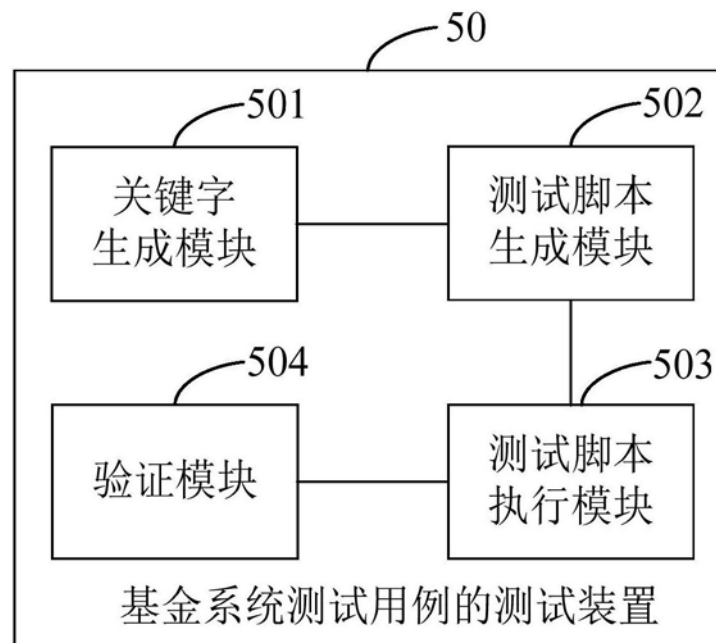


图9

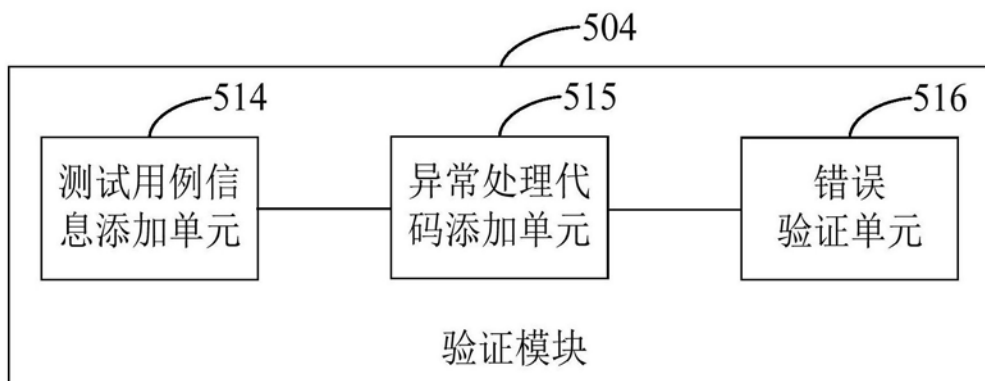


图10

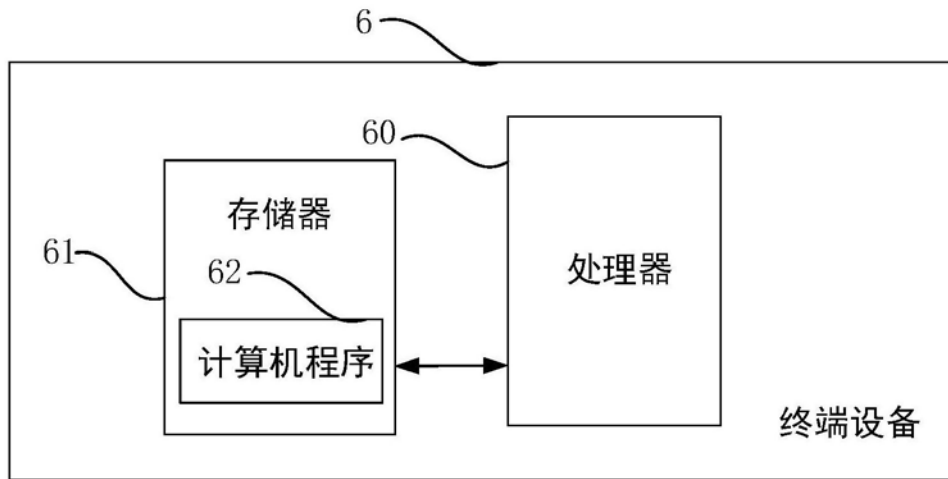


图11