



# (12)发明专利申请

(10)申请公布号 CN 110413524 A

(43)申请公布日 2019.11.05

(21)申请号 201910687877.5

(22)申请日 2019.07.26

(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 林科镨 俞子波

(74)专利代理机构 中科专利商标代理有限责任  
公司 11021

代理人 王中苇

(51)Int.Cl.  
G06F 11/36(2006.01)

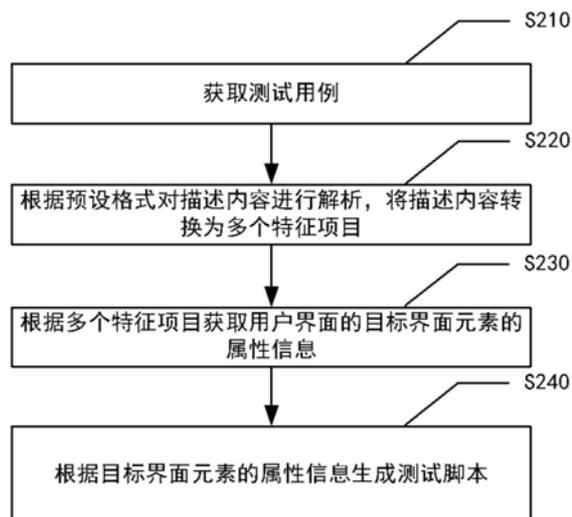
权利要求书2页 说明书11页 附图5页

## (54)发明名称

用于生成测试脚本的方法和装置、自动化测试方法

## (57)摘要

本公开提供了一种用于生成测试脚本的方法、一种用于生成测试脚本的装置、一种自动化测试方法、一种电子设备以及一种计算机可读存储介质。该用于生成测试脚本的方法包括：获取测试用例，其中，测试用例包括满足预设格式的描述内容，描述内容描述针对待测软件的用户界面的测试操作；根据预设格式对描述内容进行解析，将描述内容转换为多个特征项目；根据多个特征项目获取用户界面的目标界面元素的属性信息；以及根据目标界面元素的属性信息生成测试脚本。



1. 一种用于生成测试脚本的方法,包括:

获取测试用例,其中,所述测试用例包括满足预设格式的描述内容,所述描述内容描述针对待测软件的用户界面的测试操作;

根据所述预设格式对所述描述内容进行解析,将所述描述内容转换为多个特征项目;

根据所述多个特征项目获取所述用户界面的目标界面元素的属性信息;以及

根据所述目标界面元素的属性信息生成测试脚本。

2. 根据权利要求1所述的方法,其中,所述根据所述多个特征项目获取所述用户界面的目标界面元素的属性信息包括:

分析所述用户界面的界面结构,得到所述用户界面的多个界面元素;

确定所述多个界面元素中与所述多个特征项目中的一个或多个特征项目对应的目标界面元素;以及

获取所述目标界面元素的属性信息。

3. 根据权利要求2所述的方法,其中,分析所述用户界面的界面结构包括:

利用浏览器插件技术,通过智能引擎分析所述待测软件的用户界面的界面结构。

4. 根据权利要求1所述的方法,其中,所述根据所述目标界面元素的属性信息生成测试脚本包括:

确定执行所述测试脚本的自动化测试工具;

根据所述自动化测试工具的特性信息和所述目标界面元素的属性信息生成初始脚本;以及

对所述初始脚本进行参数化处理,以生成与所述自动化测试工具匹配的测试脚本。

5. 根据权利要求1所述的方法,其中,

所述多个特征项目包括以下中的至少一个:

要素名称、界面顺序、界面类型、操作及参数值域。

6. 一种用于生成测试脚本的装置,包括:

测试用例获取模块,获取测试用例,其中,所述测试用例包括满足预设格式的描述内容,所述描述内容描述针对待测软件的用户界面的测试操作;

测试用例转换模块,根据所述预设格式对所述描述内容进行解析,将所述描述内容转换为多个特征项目;

界面元素属性获取模块,根据所述多个特征项目获取所述用户界面的目标界面元素的属性信息;以及

测试脚本生成模块,根据所述目标界面元素的属性信息生成测试脚本。

7. 一种自动化测试方法,包括:

使用如权利要求1至5中任一项所述的方法生成测试脚本;以及

针对多个业务数据运行所述测试脚本,以便测试所述多个业务数据。

8. 一种自动化测试设备,包括:

如权利要求6所述的装置;以及

运行模块,用于针对多个业务数据运行所述测试脚本,以便测试所述多个业务数据。

9. 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指令,

其中,当所述一个或多个指令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实现权利要求1至5中任一项所述的方法。

10.一种计算机可读存储介质,存储有可执行指令,该指令被处理器执行时使处理器实现权利要求1至5中任一项所述的方法。

## 用于生成测试脚本的方法和装置、自动化测试方法

### 技术领域

[0001] 本公开涉及计算机技术领域,更具体地,涉及应用在自动化测试领域的用于生成测试脚本的方法、用于生成测试脚本的装置、自动化测试方法、电子设备以及计算机可读存储介质。

### 背景技术

[0002] 软件在版本上线前要通过版本的验收测试,以保证质量。为了满足越来越快的版本上线需求,通常会引进自动化测试工具来提高测试覆盖率和测试质量,这要求在进行测试时编制测试脚本。在基于系统界面用户界面层面的版本验收测试环节中,进行测试设计的测试人员需要具备非常丰富的业务背景知识,从而能够站在用户的角度来编制测试用例,但业务背景专业的测试人员通常不会编制测试脚本。若安排计算机技术专业的人员从事验收测试,则可能会由于不熟悉业务背景而造成在某些功能测试设计中出现测试盲区。

[0003] 因此,引进自动化测试工具进行界面验收测试的自动化测试时,通常会采用配置两组不同专业的测试人员协同工作的模式来展开。即,一组业务背景专业的测试设计人员专门设计测试用例,另一组计算机技术专业的测试脚本编制人员专门编制测试脚本。第一阶段,由业务背景专业的测试设计人员基于业务层面的测试理解,设计编写符合待测软件业务逻辑的测试用例。第二阶段,在待测软件开发完成并部署到测试环境后,计算机技术专业的测试脚本编制人员依据测试用例描述,通过录制回放或者手工编制封装测试脚本的方式来形成测试脚本,然后将测试用例与脚本进行关联对照,实现基于测试用例执行自动化测试的目标。

[0004] 在实现本公开构思的过程中,发明人发现在以现有组织模式进行系统界面用户界面层面的版本验收自动化测试工作时,测试脚本的编制、更新维护需要投入较多的时间和精力,测试脚本内容与对应测试用例可能存在差异而容易产生测试偏差。

### 发明内容

[0005] 有鉴于此,本公开提供了一种用于生成测试脚本的方法、一种用于生成测试脚本的装置、一种自动化测试方法、一种电子设备以及一种计算机可读存储介质。

[0006] 本公开的一个方面提供了一种用于生成测试脚本的方法,包括:获取测试用例,其中,上述测试用例包括满足预设格式的描述内容,上述描述内容描述针对待测软件的用户界面的测试操作;根据上述预设格式对上述描述内容进行解析,将上述描述内容转换为多个特征项目;根据上述多个特征项目获取上述用户界面的目标界面元素的属性信息;以及根据上述目标界面元素的属性信息生成测试脚本。

[0007] 根据本公开的实施例,上述根据上述多个特征项目获取上述用户界面的目标界面元素的属性信息包括:分析上述用户界面的界面结构,得到上述用户界面的多个界面元素;确定上述多个界面元素中与上述多个特征项目中的一个或多个特征项目对应的目标界面元素;以及获取上述目标界面元素的属性信息。

[0008] 根据本公开的实施例,分析上述用户界面的界面结构包括:利用浏览器插件技术,通过智能引擎分析上述待测软件的用户界面的界面结构。

[0009] 根据本公开的实施例,上述根据上述目标界面元素的属性信息生成测试脚本包括:确定执行上述测试脚本的自动化测试工具;根据上述自动化测试工具的特性信息和上述目标界面元素的属性信息生成初始脚本;以及对上述初始脚本进行参数化处理,以生成与上述自动化测试工具匹配的测试脚本。

[0010] 根据本公开的实施例,上述多个特征项目包括以下中的至少一个:要素名称、界面顺序、界面类型、操作及参数值域。

[0011] 本公开的另一方面提供了一种用于生成测试脚本的装置,包括:测试用例获取模块,获取测试用例,其中,上述测试用例包括满足预设格式的描述内容,上述描述内容描述针对待测软件的用户界面的测试操作;测试用例转换模块,根据上述预设格式对上述描述内容进行解析,将上述描述内容转换为多个特征项目;界面元素属性获取模块,根据上述多个特征项目获取上述用户界面的目标界面元素的属性信息;以及测试脚本生成模块,根据上述目标界面元素的属性信息生成测试脚本。

[0012] 根据本公开的实施例,上述界面元素属性获取模块包括:分析单元,分析上述用户界面的界面结构,得到上述用户界面的多个界面元素;界面元素确定单元,确定上述多个界面元素中与上述多个特征项目中的一个或多个特征项目对应的目标界面元素;以及属性获取单元,获取上述目标界面元素的属性信息。

[0013] 根据本公开的实施例,上述分析单元包括:智能分析子单元,利用浏览器插件技术,通过智能引擎分析上述待测软件的用户界面的界面结构。

[0014] 根据本公开的实施例,上述测试脚本生成模块包括:工具确定单元,确定执行上述测试脚本的自动化测试工具;初始脚本生成单元,根据上述自动化测试工具的特性信息和上述目标界面元素的属性信息自动生成初始脚本;以及测试脚本生成单元,对上述初始脚本进行参数化处理,以生成与上述自动化测试工具匹配的测试脚本。

[0015] 根据本公开的实施例,上述多个特征项目包括以下中的至少一个:要素名称、界面顺序、界面类型、操作及参数值域。

[0016] 本公开的另一方面提供了一种自动化测试方法,包括:使用上述的方法生成测试脚本;以及针对多个业务数据运行上述测试脚本,以便测试上述多个业务数据。

[0017] 本公开的另一方面提供了一种自动化测试设备,包括:上述的装置;以及运行模块,用于针对多个业务数据运行上述测试脚本,以便测试上述多个业务数据。

[0018] 本公开的另一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指令,其中,当上述一个或多个指令被上述一个或多个处理器执行时,使得上述一个或多个处理器实现上述的方法。

[0019] 本公开的另一方面提供了一种计算机可读存储介质,存储有可执行指令,该指令被处理器执行时使处理器实现上述的方法。

[0020] 根据本公开的实施例,能够根据包括满足预设格式的描述内容的测试用例,通过解析、转换等来自动生成测试脚本。从而,能够实现测试用例向测试脚本的自动转换。在测试用例发生变更的情况下,直接自动生成新的测试脚本即可,无需进行测试脚本的更新维护。并且,通过上述方式形成的测试脚本能够直接反映测试用例的内容,不会产生测试用例

的描述不清引起的测试偏差。

### 附图说明

[0021] 通过以下参照附图对本公开实施例的描述,本公开的上述以及其他目的、特征和优点将更为清楚,在附图中:

[0022] 图1示意性示出了根据本公开实施例的用于生成测试脚本的方法和装置的应用场景;

[0023] 图2示意性示出了根据本公开实施例的用于生成测试脚本的方法的流程图;

[0024] 图3示出了根据本公开实施例的一个测试用例和与之对应的转换后的特征项目;

[0025] 图4示意性示出了根据本公开实施例的根据多个特征项目获取用户界面的目标界面元素的属性信息的流程图;

[0026] 图5示意性示出了根据目标界面元素的属性信息生成测试脚本的流程图;

[0027] 图6示意性示出了根据本公开实施例的用于生成测试脚本的装置的框图;

[0028] 图7示意性示出了根据本公开实施例的界面元素属性获取模块的框图;

[0029] 图8示意性示出了根据本公开实施例的测试脚本生成模块的框图;以及

[0030] 图9示意性示出了根据本公开实施例的电子设备的框图。

### 具体实施方式

[0031] 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。

[0032] 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。

[0033] 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。

[0034] 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中至少一个的装置”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的装置等)。在使用类似于“A、B或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或C中至少一个的装置”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的装置等)。

[0035] 本公开的实施例提供了一种用于生成测试脚本的方法以及能够应用该方法的装置。该方法包括以下过程。获取测试用例。其中,测试用例包括满足预设格式的描述内容,描述内容描述针对待测软件的用户界面的测试操作。根据预设格式对描述内容进行解析,将

描述内容转换为多个特征项目。根据多个特征项目获取用户界面的目标界面元素的属性信息。根据目标界面元素的属性信息生成测试脚本。

[0036] 图1示意性示出了根据本公开实施例的用于生成测试脚本的方法和装置的应用场景。需要注意的是,图1所示仅为可以应用本公开实施例的场景示例,以帮助本领域技术人员理解本公开的技术内容,并未详细示出应用本公开实施例的系统架构。

[0037] 如图1所示,测试人员编制测试用例并提供给用于生成测试脚本的方法或装置。在生成测试脚本环节,用于生成测试脚本的方法或装置基于测试用例生成测试脚本,并将该测试脚本提供给实际测试环节。在实际测试环节,运行上述测试脚本,得到测试结果。本公开主要着眼于图1中的生成测试脚本环节。

[0038] 在传统的测试组织下,尤其在基于系统界面用户界面层面的版本验收测试中,测试人员从用户角度进行测试设计,需要具备非常丰富的业务背景知识,从而能够站在用户的角度来编制测试用例。

[0039] 为了避免在某些功能测试设计时产生测试盲区,当待测软件的业务逻辑比较复杂和繁琐时,通常会安排业务背景专业的测试人员来从事验收测试,而不安排从事过软件开发的测试人员进行相关的测试工作。业务背景专业的测试人员对软件开发等IT领域相关知识掌握较少,但编制自动化测试工具的测试脚本时对软件开发有一定的技能水平要求。因此,在对业务场景逻辑比较复杂的软件进行自动化测试时,可能会由于自动化测试工具使用问题或者测试人员个体技能问题导致测试脚本编制质量低、执行效果差,从而自动化测试难以达到预想的效果,无法提高测试效率。

[0040] 在这种模式下,引进自动化测试工具进行自动化的界面验收测试时,通常会采用配置两组不同专业的测试人员协同工作的模式来展开。即,一组业务背景专业的测试设计人员专门设计测试用例,另一组计算机技术专业的测试脚本编制人员专门编制测试脚本。第一阶段,由业务背景专业的测试设计人员基于业务层面的测试理解,设计编写符合待测软件业务逻辑的测试用例。第二阶段,在待测软件开发完成并部署到测试环境后,计算机技术专业的测试脚本编制人员依据测试用例描述,通过录制回放或者手工编写封装测试脚本的方式来形成测试脚本,然后将测试用例与脚本进行关联对照,实现基于测试用例执行自动化测试的目标。

[0041] 在以现有组织模式进行系统界面用户界面层面的版本验收自动化测试工作时,主要存在以下问题。

[0042] 1、只能在软件完成研发并部署到测试环境后才能开展测试脚本的编制工作。因为不同于接口测试和集成测试的自动化,界面层面的测试脚本的开发工作必须依赖于已经开发并部署好对应功能的软件。在软件功能还未完成开发并部署到测试环境时,无法展开对应的测试脚本的开发工作。并且,该测试脚本的编制工作需要根据测试用例来人工执行,会影响整个测试工作的进度。

[0043] 2、每个测试设计人员编写的测试用例标准不统一,测试脚本编制人员编制出来的测试脚本内容与对应测试用例可能存在差异。不同的两组人员需要进行协调,因此测试脚本编制人员需要对测试设计人员编写的测试用例充分理解后才能编制出完全符合测试设计人员的测试用例中对应的测试点。每个测试设计人员编写的测试用例标准不统一,并且测试用例中的内容详尽程度参差不齐,导致测试脚本编制人员编制的脚本质量无法保证,

容易产生测试偏差。

[0044] 3、测试脚本编制更新工作量大。在系统界面层面的验收测试自动化过程中,测试脚本需要随着系统界面控件/组件的属性变化而进行更新维护。当系统界面控件属性发生改变时,尤其是前台系统界面没有发生改变只是后台系统控件属性发生改变时,测试人员往往无法在第一时间发现,导致测试脚本未能及时维护而执行失败。在执行失败后,对测试脚本维护更新也需要投入较多的时间和精力。

[0045] 4、目前业界存在部分基于关键字描述的用于测试用例编写的工具,一般是通过对具体的系统功能进行封装,用户通过对封装后的系统功能进行组装的模式来形成测试用例。但尚未出现直接依据测试用例的自然语言描述来自动生成测试脚本的先例。

[0046] 本公开实施例解决了上述的测试脚本开发工作介入晚、测试用例描述不统一、测试脚本与对应测试用例覆盖情况存在差异、测试脚本编制更新工作量大等技术问题。本公开实施例能够直接根据测试用例自动生成测试脚本。测试人员基于预设格式通过自然描述进行测试用例设计后,通过用于生成测试脚本的方法或装置自动对上述自然描述进行转换并生成对应的测试脚本,免去了测试脚本编制人员的编制工作及存量测试脚本的更新工作,提高了测试效率。

[0047] 图2示意性示出了根据本公开实施例的用于生成测试脚本的方法的流程图。

[0048] 如图2所示,该方法包括操作S210~S240。

[0049] 在操作S210,获取测试用例。其中,测试用例包括满足预设格式的描述内容,描述内容描述针对待测软件的用户界面的测试操作。该测试用例通过自然语言描述针对待测软件的具体测试操作,是由具备非常丰富的业务背景知识的测试人员站在用户的角度编制的业务测试用例。对于该测试用例的具体示例,将在下文中参照图3进行具体说明。

[0050] 在操作S220,根据预设格式对描述内容进行解析,将描述内容转换为多个特征项目。多个特征项目例如包括要素名称、界面顺序、界面类型、操作及参数值域。该多个特征项目的格式与测试用例的描述内容的格式不同,结合了用户界面的界面元素的属性信息,能够用于用户界面的目标界面元素的识别。对于该多个特征项目的具体示例,将在下文中参照图3进行具体说明。

[0051] 在操作S230,根据多个特征项目获取用户界面的目标界面元素的属性信息。用户界面上具有多个界面元素,目标界面元素是指多个界面元素中的与一个或多个特征项目对应的界面元素。目标界面元素的属性信息是指生成测试脚本所需的关键属性信息,例如包括目标界面元素的name、id、其所属界面的层次结构信息以及该目标界面元素与其他界面元素的层次关系等。

[0052] 在操作S240,根据目标界面元素的属性信息生成测试脚本。例如,可以对获取到的用户界面的目标界面元素的属性信息进行整合和拼接,并基于该整合和拼接后的目标界面元素的属性信息生成测试脚本。

[0053] 根据本公开的实施例,能够根据包括满足预设格式的描述内容的测试用例,通过解析、转换等自动生成测试脚本。从而,能够实现测试用例向测试脚本的自动转换。在测试用例发生变更的情况下,直接自动生成新的测试脚本即可,无需进行测试脚本的更新维护。并且,通过上述方式形成的测试脚本能够直接反映测试用例的内容,不会产生测试用例的描述不清引起的测试偏差。

[0054] 图3示出了根据本公开实施例的一个测试用例和与之对应的转换后的特征项目。

[0055] 图3的一部分示出了根据本公开实施例的一个示例测试用例。该测试用例包括满足预设格式的描述内容,描述内容描述针对待测软件的用户界面的测试操作。并且,该测试操作是通过非计算机专业人员也能读懂的自然语言描述的。本领域技术人员可以理解,图3所示测试用例仅为示例,本公开实施例并不局限于此。

[0056] 图3所示的该测试用例可以包括以下部分。

[0057] 第一步,打开浏览器并在浏览器的地址框中输入http://test-123.icbc/icbc/uvw/index.jsp;

[0058] 第二步,在界面上的“通行证”文本框中输入:100223500;

[0059] 第三步,在界面上的“密码”框中输入:password-123;

[0060] 第四步,点击界面上的“登录”按钮;

[0061] 第五步,点击界面上的“外汇买卖”菜单;

[0062] 第六步,点击界面上的“外汇平仓明细表”菜单;

[0063] 第七步,从界面上的“业务种类”下拉框中选择某一具体业务种类;

[0064] 第八步,从界面上的“平仓种类”下拉框中选择某一具体平仓种类;

[0065] 第九步,在界面上的“对方地区号”文本框中输入某一具体对方地区号;

[0066] 第十步,从界面上的“平仓日期”日期框中选择某一具体平仓日期;

[0067] 第十一步,从界面上的“交割日期”日期框中选择某一具体交割日期;

[0068] 第十二步,从界面上的第二个“币种”下拉框中选择某一具体币种;

[0069] 第十三步,从界面上的“起始时间”日期框中选择某一具体起始时间;

[0070] 第十四步,从界面上的“截止时间”日期框中选择某一具体截止时间;

[0071] 第十五步,点击界面上的“查询”按钮。

[0072] 在第一步的测试操作为打开浏览器并在浏览器的地址框中输入http://test-123.icbc/icbc/uvw/index.jsp这一地址的情况下,虽然有很多种语言描述可以表达出该意思,但图3所示的测试用例只采用图示的唯一的描述方式,即,步骤1:交互步骤:打开浏览器输入http://test-123.icbc/icbc/uvw/index.jsp。在第二步的测试操作为在通行证这一文本框中输入“100223500”这一串数字的情况下,虽然有很多种语言描述可以表达出该意思,但图3所示的示例测试用例只采用图示的唯一的描述方式,即,步骤2:【界面·通行证】文本框输入:100223500。关于其他步骤的描述方式也类似,在此不再赘述。当然,测试用例的描述方式并不限于图3所示的描述方式,而能够采用任意一种描述方式。但是,一旦选择一种描述方式后,测试用例中的所有描述内容以及所有的测试用例都应该采用该种描述方式描述。显而易见,图3所示的测试用例的编写依据预设的标准化的描述规则,并且非计算机专业人员也能读懂其描述语言。依据上述预设的标准化的描述规则,能够通过自然语言编写格式统一的测试用例,避免产生理解误差,从而能够将测试用例的编写成果标准化,提高测试用例设计成果质量和标准。

[0073] 图3的另一部分示出了与根据本公开实施例的一个测试用例对应的转换后的特征项目。该多个特征项目的格式与测试用例的描述内容的格式不同,但描述出相同的测试操作。例如“通行证1”对应于测试用例中的“【界面·通行证】”,表示用户界面中的第一个名称为“通行证”的要素,“币种2”对应于测试用例中的【界面·币种·2】,表示用户界面中的第

二个名称为“币种”的要素，“textbox input”对应于测试用例中的“文本框输入”。该多个特征项目由于结合了用户界面的界面元素的属性信息，因此能够用于用户界面的目标界面元素的识别。该多个特征项目例如包括：要素名称、界面顺序、界面类型、操作及参数值域。在图3的该部分中，要素名称例如是指“通行证”、“密码”、“登录”、“外汇买卖”等文字。界面顺序例如是指要素名称后面的数字“1”、“2”等，表示该次操作中要操作的界面元素顺序，例如第十二步的测试操作为从界面上的第二个“币种”下拉框中选择某一具体币种，其中的“币种2”表示此次要操作的是界面上的第二个名称为币种的下拉框。界面顺序的定义方式并不唯一，例如可以是从左到右、从上到下的顺序。界面类型例如是指与“text box”、“button”、“menu”等对应的文本框、按钮、菜单等。操作例如是指与“input”、“click”等对应的输入、点击等具体操作。参数值域例如是指“100223500”、“password-123”等。

[0074] 图4示意性示出了根据本公开实施例的根据多个特征项目获取用户界面的目标界面元素的属性信息的流程图。

[0075] 如图4所示，在一个示例中，操作S230还可以包括操作S231~S233。

[0076] 在操作S231，分析用户界面的界面结构，得到用户界面的多个界面元素。该多个界面元素例如包括图片、按钮、表格等。以浏览器中运行的网页界面为例，对浏览器中网页界面的DOM、BOM结构进行分析，得到多个界面元素。

[0077] 在操作S232，确定多个界面元素中与多个特征项目中的一个或多个特征项目对应的目标界面元素。例如，根据多个特征项目中的要素名称、界面顺序、界面类型在已得到的多个界面元素中进行查找，确定与特征项目的内容一致的目标界面元素。该目标界面元素与多个特征项目中的一个或多个特征项目对应，因此与上述测试用例中的一条或多条描述内容相对应，能够用于生成测试脚本。

[0078] 在操作S233，获取目标界面元素的属性信息。该目标界面元素的属性信息是指生成测试脚本所需的关键属性信息，例如包括目标界面元素的name、id、其所属界面的层次结构信息以及该目标界面元素与其他界面元素的层次关系等。

[0079] 根据本公开的实施例，分析用户界面的界面结构包括：利用浏览器插件技术，通过智能引擎分析待测软件的用户界面的界面结构，从而能够快速得到用户界面的多个界面元素。

[0080] 图5示意性示出了根据目标界面元素的属性信息生成测试脚本的流程图。

[0081] 如图5所示，在一个示例中，操作S240还可以包括操作S241~S243。

[0082] 在操作S241，确定执行测试脚本的自动化测试工具。该自动化测试工具可以由测试人员依据喜好来自由选择，例如可以为UFT、selenium等。从而，测试人员对自动化测试工具的可选范围变大。

[0083] 在操作S242，根据自动化测试工具的特性信息和目标界面元素的属性信息生成初始脚本。自动化测试工具的特性信息包括该自动化测试工具的语法和对脚本的要求。

[0084] 在操作S243，对初始脚本进行参数化处理，以生成与自动化测试工具匹配的测试脚本。在该参数化处理中，对脚本实体和脚本数据进行分离，对脚本进行参数化编制。例如，可以对初始脚本中的所有输入数据进行参数化，再映射输入输出参数关系。完成初始脚本内部的数据传递和对输入数据的参数化后，即能够处理一些流程较为复杂的业务测试场景，比如抓取某页面的订单号，跳转到另外一个页面并进行输入等。

[0085] 从而,能够实现测试脚本对多套测试数据执行的支持,免去测试人员在测试脚本开发过程中针对脚本数据参数化的开发及调试工作量。

[0086] 根据本公开的实施例,多个特征项目包括以下中的至少一个:要素名称、界面顺序、界面类型、操作及参数值域。

[0087] 图6示意性示出了根据本公开实施例的用于生成测试脚本的装置的框图。

[0088] 如图6所示,本公开实施例的用于生成测试脚本的装置600包括测试用例获取模块610、测试用例转换模块620、界面元素属性获取模块630以及测试脚本生成模块640。

[0089] 测试用例获取模块610获取测试用例。其中,测试用例包括满足预设格式的描述内容,描述内容描述针对待测软件的用户界面的测试操作。

[0090] 测试用例转换模块620根据预设格式对描述内容进行解析,将描述内容转换为多个特征项目。

[0091] 界面元素属性获取模块630根据多个特征项目获取用户界面的目标界面元素的属性信息。

[0092] 测试脚本生成模块640根据目标界面元素的属性信息生成测试脚本。例如,可以对由界面元素属性获取模块630获取到的用户界面的目标界面元素的属性信息进行整合和拼接,并基于该整合和拼接后的目标界面元素的属性信息生成测试脚本。

[0093] 图7示意性示出了根据本公开实施例的界面元素属性获取模块的框图。

[0094] 如图7所示,本公开实施例的界面元素属性获取模块630包括分析单元631、界面元素确定单元632以及属性获取单元633。

[0095] 分析单元631可以用于分析用户界面的界面结构,得到用户界面的多个界面元素。以浏览器中运行的网页界面为例,分析单元631对浏览器中网页界面的DOM、BOM结构进行分析,得到多个界面元素。

[0096] 界面元素确定单元632可以用于确定多个界面元素中与多个特征项目中的一个或多个特征项目对应的目标界面元素。例如,界面元素确定单元632根据多个特征项目中的要素名称、界面顺序、界面类型在由分析单元631得到的多个界面元素中进行查找,确定与特征项目的内容一致的目标界面元素。

[0097] 属性获取单元633可以用于获取目标界面元素的属性信息。

[0098] 根据本公开的实施例,分析单元631包括智能分析子单元。智能分析子单元可以用于利用浏览器插件技术,通过智能引擎分析待测软件的用户界面的界面结构。

[0099] 图8示意性示出了根据本公开实施例的测试脚本生成模块的框图。

[0100] 如图8所示,本公开实施例的测试脚本生成模块640包括工具确定单元641、初始脚本生成单元642以及测试脚本生成单元643。

[0101] 工具确定单元641可以用于确定执行测试脚本的自动化测试工具。

[0102] 初始脚本生成单元642可以用于根据自动化测试工具的特性信息和目标界面元素的属性信息自动生成初始脚本。

[0103] 测试脚本生成单元643可以用于对初始脚本进行参数化处理,以生成与自动化测试工具匹配的测试脚本。

[0104] 根据本公开的实施例,多个特征项目包括以下中的至少一个:要素名称、界面顺序、界面类型、操作及参数值域。

[0105] 本公开的另一方面提供了一种自动化测试方法。该自动化测试方法包括：使用上述的用于生成测试脚本的方法生成测试脚本；以及针对多个业务数据运行所述测试脚本，以便测试所述多个业务数据。

[0106] 在该自动化测试方法中，能够根据包括满足预设格式的描述内容的测试用例，通过解析、转换等自动生成测试脚本。因此，与现有的自动化测试方法相比，该自动化测试方法的测试效率更高。该自动化测试方法可以针对多个业务数据自动运行自动生成的测试脚本，从而能够提高测试的覆盖率，节约测试时间。运行后每个业务数据的结果都可以从日志、截图等多维度查看，从而测试脚本的执行情况得到反馈。

[0107] 本公开的另一方面提供了一种自动化测试设备。该自动化测试设备包括：上述的用于生成测试脚本的装置以及运行模块。运行模块用于针对多个业务数据运行上述测试脚本，以便测试上述多个业务数据。

[0108] 根据本公开的实施例的模块、子模块、单元、子单元中的任意多个、或其中任意多个的至少部分功能可以在一个模块中实现。根据本公开实施例的模块、子模块、单元、子单元中的任意一个或多个可以被拆分成多个模块来实现。根据本公开实施例的模块、子模块、单元、子单元中的任意一个或多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的硬件或固件来实现，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，根据本公开实施例的模块、子模块、单元、子单元中的一个或多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。

[0109] 例如，测试用例获取模块610、测试用例转换模块620、界面元素属性获取模块630以及测试脚本生成模块640中的任意多个可以合并在一个模块/单元/子单元中实现，或者其中的任意一个模块/单元/子单元可以被拆分成多个模块/单元/子单元。或者，这些模块/单元/子单元中的一个或多个模块/单元/子单元的至少部分功能可以与其他模块/单元/子单元的至少部分功能相结合，并在一个模块/单元/子单元中实现。根据本公开的实施例，测试用例获取模块610、测试用例转换模块620、界面元素属性获取模块630以及测试脚本生成模块640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，测试用例获取模块610、测试用例转换模块620、界面元素属性获取模块630以及测试脚本生成模块640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。

[0110] 需要说明的是，本公开实施例的用于生成测试脚本的装置与本公开实施例的用于生成测试脚本的方法部分是相对应的，用于生成测试脚本的装置部分的描述具体参考用于生成测试脚本的方法部分，在此不再赘述。

[0111] 本公开实施例的电子设备包括一个或多个处理器和用于存储一个或多个指令的存储器，当一个或多个指令被一个或多个处理器执行时，使得一个或多个处理器实现上文描述的方法。

[0112] 图9示意性示出了根据本公开实施例的电子设备的框图。图9示出的电子设备仅仅是一个示例,不应对本公开实施例的功能和使用范围带来任何限制。

[0113] 如图9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。

[0114] 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ROM902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。

[0115] 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。

[0116] 根据本公开的实施例,根据本公开实施例的方法流程可以被实现为计算机软件程序。例如,本公开的实施例包括一种计算机程序产品,其包括承载在计算机可读存储介质上的计算机程序,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。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。

[0117] 本公开实施例的计算机可读存储介质存储有可执行指令,该指令被处理器执行时使处理器实现上文描述的方法。

[0118] 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方法。

[0119] 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。

[0120] 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。

[0121] 根据本公开的实施例,通过标准化、规范化的测试用例描述,能够提高测试人员的测试用例编写能力,实现工作成果统一性,有利于大规模的测试组织模式下的测试资产的积累和维护。

[0122] 根据本公开的实施例,测试人员只需要在测试用例设计阶段依据规则编写标准化的测试用例,在软件完成研发并部署到测试环境后即可直接生成对应的测试脚本并执行测试,免去了原有的测试脚本开发人员的工作环节及工作量,减少了人力资源的投入,加快了版本测试进度。同时,也解决了原先由于分工编写,相互沟通工作量大且测试脚本与业务用例的对照关系程度低等问题。

[0123] 根据本公开的实施例,基于测试脚本自动生成的模式,在待测软件界面控件发生变动时,直接自动生成新的测试脚本即可。从而在系统界面层面验收测试的自动化测试模式下,免去了需要对不同类型自动化测试工具的测试脚本更新投入的维护工作量。

[0124] 根据本公开的实施例,由于省去了测试脚本开发环节,极大的缩短了测试时间,提升了测试效率,测试人员只需要专注于业务流程及测试用例的编写,无需关心具体的测试脚本,同时也降低了对测试人员的技术要求。

[0125] 附图中的流程图和框图,图示了按照本公开各种实施例的方法、装置、电子设备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。也要注意的,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。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。

[0126] 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。

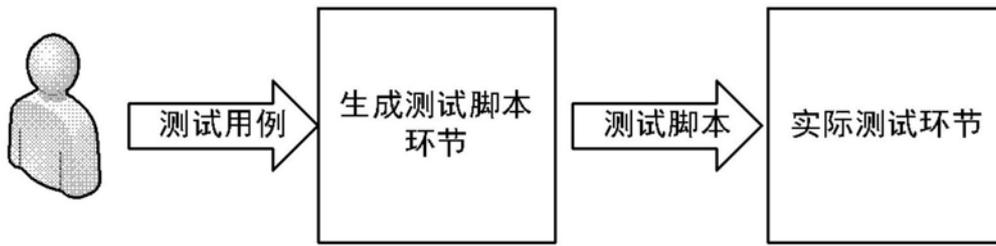


图1

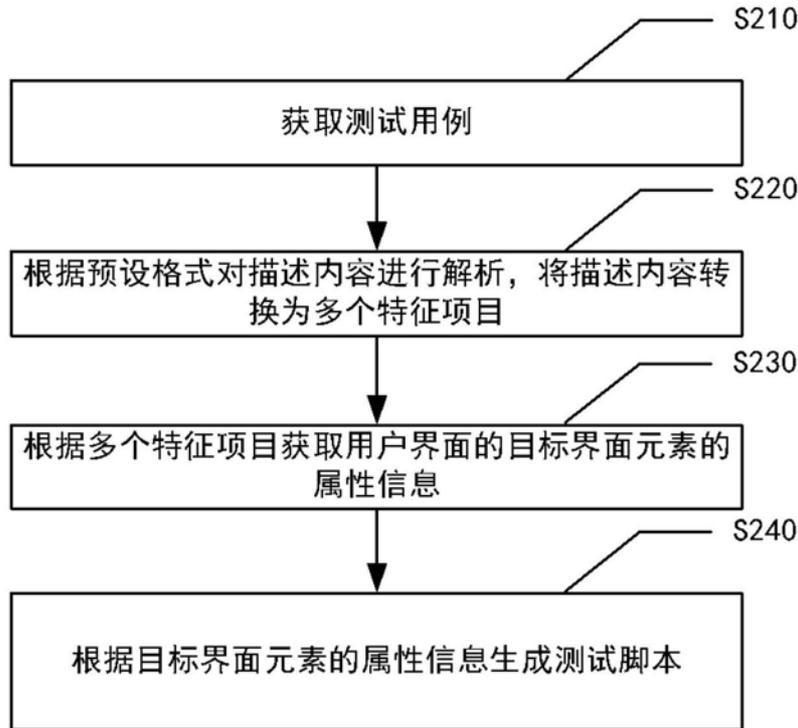
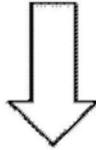


图2

## 测试用例

步骤1:交互步骤:打开浏览器输入http:// test-123.icbc/icbc/uvw/index.jsp  
 步骤2:【界面·通行证】文本框输入:100223500  
 步骤3:【界面·密码】密码框输入:password-123  
 步骤4:点击【界面·登录】按钮  
 步骤5:点击【界面·外汇买卖】菜单  
 步骤6:点击【界面·外汇平仓明细表】菜单  
 步骤7:【界面·业务种类】下拉框选择:\$\$业务种类\$\$  
 步骤8:【界面·平仓种类】下拉框选择:\$\$平仓种类\$\$  
 步骤9:【界面·对方地区号】文本框输入:\$\$对方地区号\$\$  
 步骤10:【界面·平仓日期】日期框选择:\$\$平仓日期\$\$  
 步骤11:【界面·交割日期】日期框选择:\$\$交割日期\$\$  
 步骤12:【界面·币种·2】下拉框选择:\$\$币种\$\$  
 步骤13:【界面·起始时间】日期框选择:\$\$起始时间\$\$  
 步骤14:【界面·截止时间】日期框选择:\$\$截止时间\$\$  
 步骤15:点击【界面·查询】按钮



## 转换后的特征项目

```

http:// test-123.icbc/icbc/uvw/index.jsp;
通行证 1 textbox input 100223500;
密码 1 object input password-123;
登录 1 button click;
外汇买卖 1 menu click;
外汇平仓明细表 1 menu click;
业务种类 1 combox select 1, 即期;
平仓种类 1 combox select 3, 平仓申请;
对方地区号 1 textbox input 0;
平仓日期 1 datetime select 01/06/2018;
交割日期 1 datetime select 01/06/2018;
币种 2 combox select 156, CNY;
起始时间 1 datetime select 00:00:10;
截止时间 1 datetime select 23:59:10;
查询 1 button click;
  
```

图3

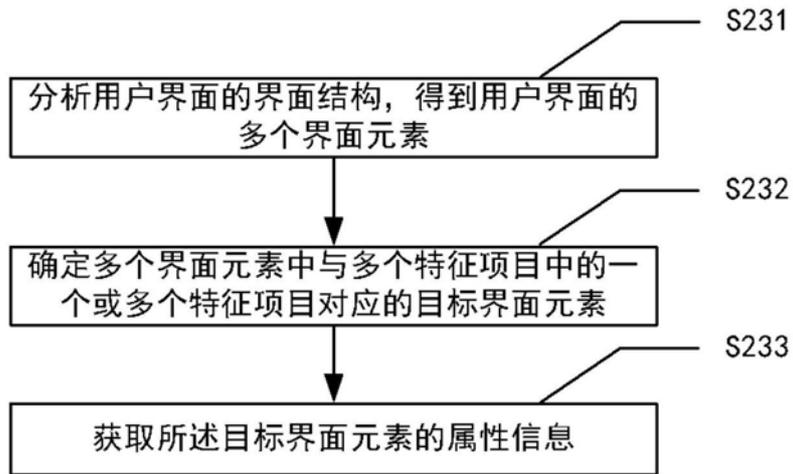


图4

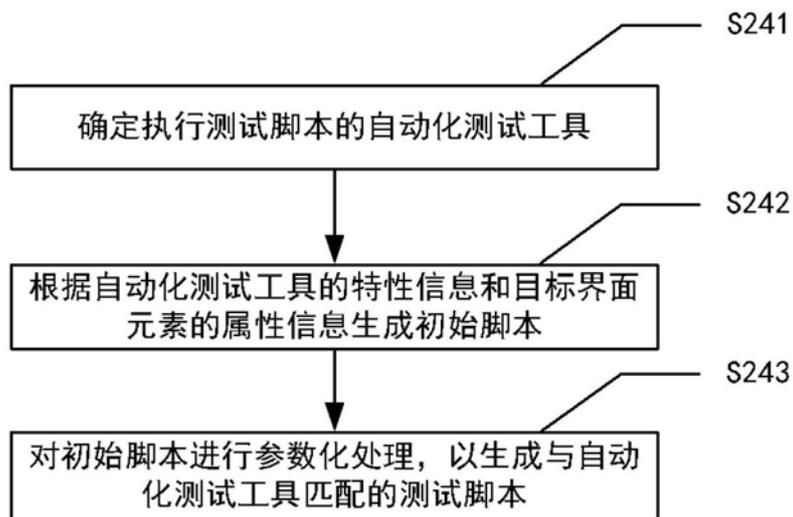


图5

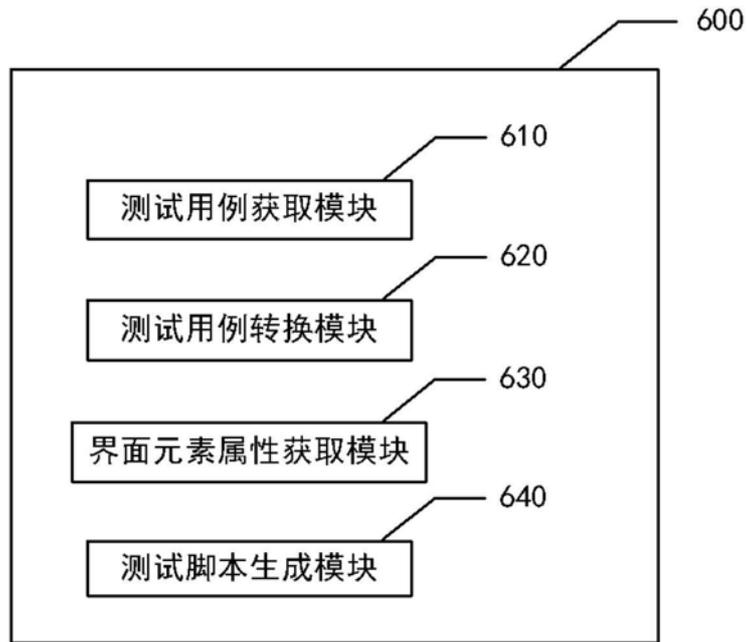


图6

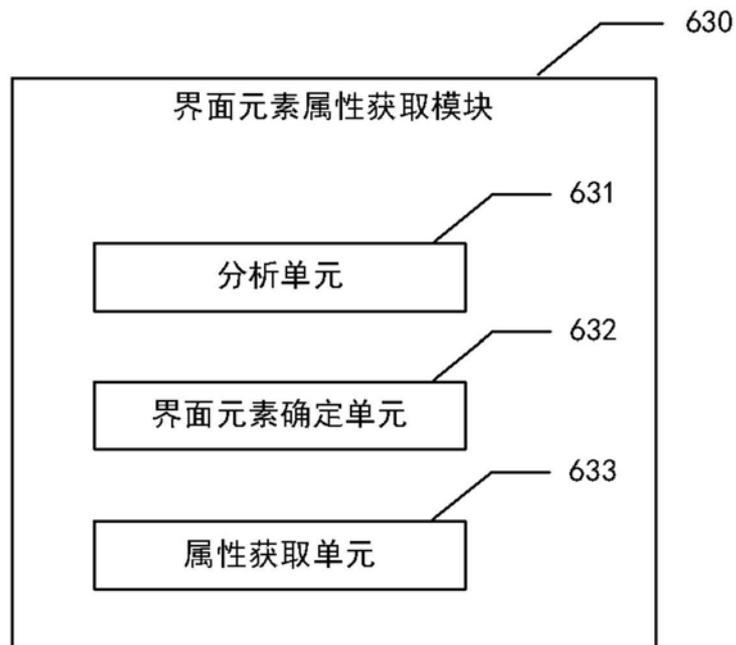


图7

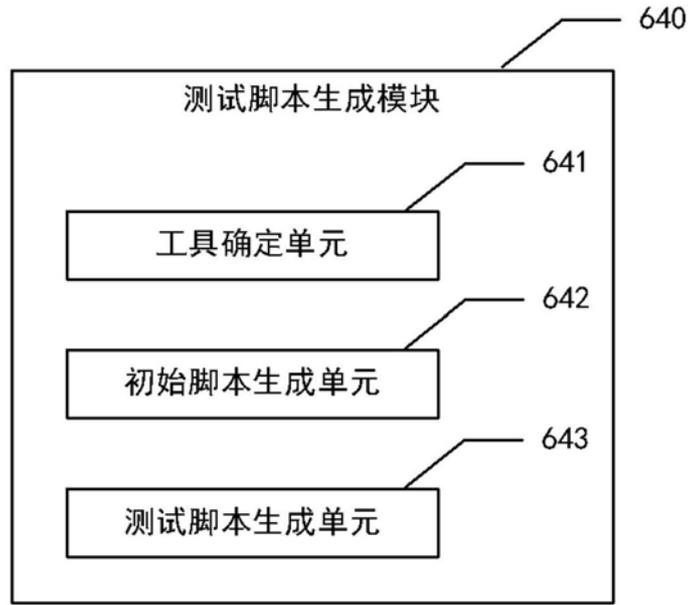


图8

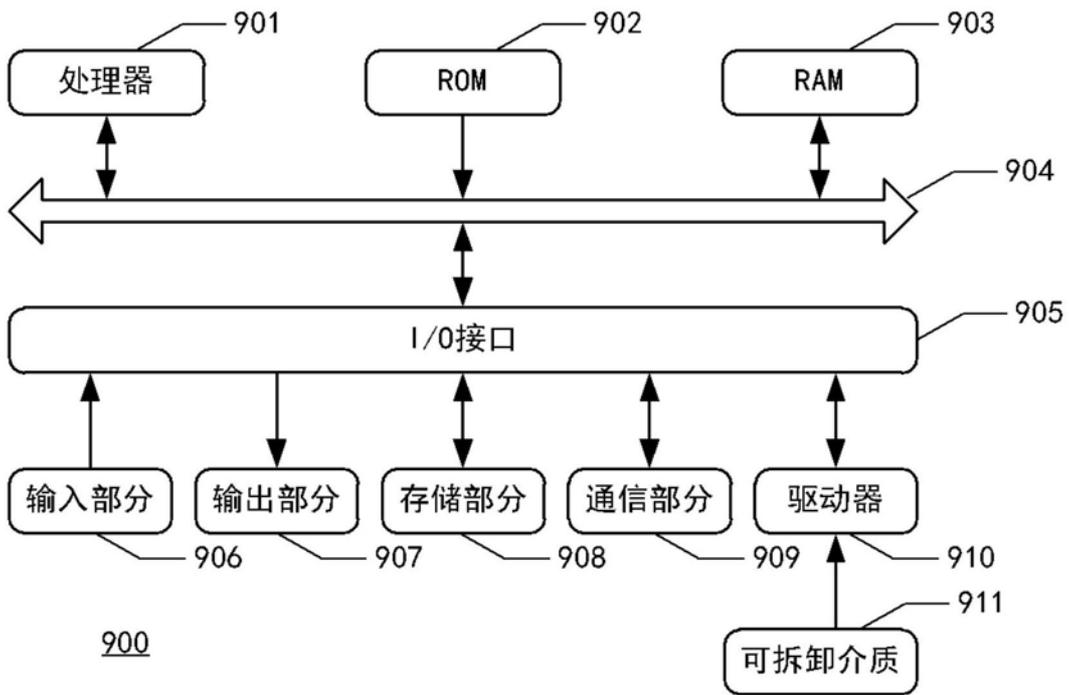


图9