



(12) 发明专利申请

(10) 申请公布号 CN 112699055 A

(43) 申请公布日 2021.04.23

(21) 申请号 202110068911.8

(22) 申请日 2021.01.19

(71) 申请人 航天恒星科技有限公司
地址 100086 北京市海淀区知春路82号院

(72) 发明人 邢兆飞 权怀炜 李莹

(74) 专利代理机构 中国航天科技专利中心
11009

代理人 陈鹏

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

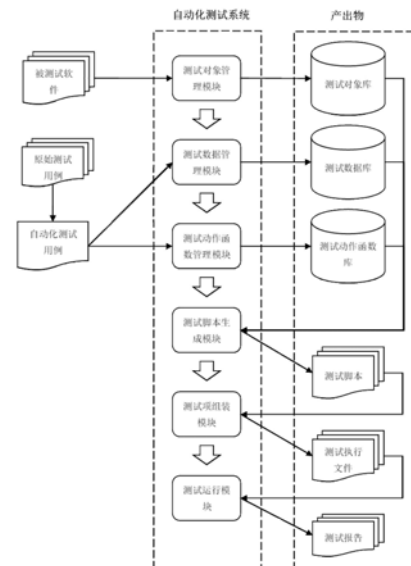
权利要求书2页 说明书7页 附图2页

(54) 发明名称

一种维护成本较低的软件自动化测试方法及系统

(57) 摘要

一种维护成本较低的软件自动化测试方法及系统,属于计算机软件技术领域。本发明包括:标识和描述测试对象;对测试用例进行归纳加工抽象出测试数据、测试动作,并使用采用标准化的语言及格式进行描述和存储;组合拼接测试对象、测试数据和动作形成测试脚本;调整组织测试脚本,执行并生成结果报告。系统包括:测试对象管理模块、测试数据管理模块、测试动作函数管理模块、测试脚本生成模块、测试项组装模块、测试运行模块。本发明降低了测试用例、测试数据、测试动作、测试对象之间的耦合度,提高了自动化测试程序代码的复用程度,提高了测试程序代码的可维护性,降低了软件自动化测试中的维护和修改成本,提高了软件自动化测试实施的效率。



1. 一种维护成本较低的软件自动化测试方法,其特征在于,包括如下步骤:

S1,标识和描述测试对象;

S2,获取测试用例,拆分出测试动作和测试数据并进行唯一标识和描述;

S3,通过组合S1、S2所述测试对象、测试数据、测试动作生成测试脚本;

S4,通过组织排列S3所述测试脚本生成测试执行文件;

S5,通过运行S4所述测试执行文件生成测试报告。

2. 根据权利要求1所述的一种维护成本较低的软件自动化测试方法,其特征在于:所述S1中,使用唯一名称及ID命名测试对象,并使用标准化的程序语言描述测试对象的位置或从属关系以唯一确定该测试对象。

3. 根据权利要求1所述的一种维护成本较低的软件自动化测试方法,其特征在于:所述S2中,从原始测试用例中筛选出的用于自动化测试的用例。

4. 根据权利要求1所述的一种维护成本较低的软件自动化测试方法,其特征在于:所述S2中,从自动化测试用例中拆分出测试数据,并使用结构化的数据格式进行存储。

5. 根据权利要求1所述的一种维护成本较低的软件自动化测试方法,其特征在于:所述S2中,从自动化测试用例中拆分出测试动作,并使用程序语言进行描述和实现。

6. 一种维护成本较低的软件自动化测试系统,其特征在于,包括:

测试对象管理模块,对测试对象进行唯一标识和描述,存入测试对象库;

测试数据管理模块,对测试用例进行抽象和拆分形成测试数据并进行唯一标识和描述,存入测试数据文件库;

测试动作函数管理模块,对测试用例进行抽象和拆分形成测试动作并进行唯一标识和描述,存入测试动作函数库;

测试脚本生成模块,根据测试需求拼装组合测试对象、测试数据、测试动作生成测试脚本文件;

测试项组装模块,组织测试脚本执行顺序,生成测试执行文件;

测试运行模块,运行测试执行文件后生成测试执行结果报告。

7. 根据权利要求6所述的一种维护成本较低的软件自动化测试系统,其特征在于:所述测试对象管理模块采用标准化的程序语言描述和唯一标识被测试的对象,存入测试对象库;

所述测试数据管理模块横向分析和归纳自动化测试用例中使用的测试数据,以键值对和列表格式将其存储于结构化的数据文件中;

所述测试动作函数管理模块分析自动化测试用例,抽象出测试动作,并使用程序语言描述实现这个动作,存入测试动作函数库。

8. 根据权利要求6所述的一种维护成本较低的软件自动化测试系统,其特征在于:所述测试脚本生成模块从测试对象库、测试数据文件库和测试动作函数库中取出测试对象、测试数据、测试动作,根据原始测试用例的具体需求进行组合,生成测试脚本文件。

9. 根据权利要求6所述的一种维护成本较低的软件自动化测试系统,其特征在于,所述测试脚本生成模块以数据驱动的方式来组织测试脚本,具体为:构建一个测试模板脚本,将测试数据以数据集的方式提供给该模板脚本,产生满足多项测试的脚本。

10. 根据权利要求6所述的一种维护成本较低的软件自动化测试系统,其特征在于:所

述测试项组装模块用于调整测试脚本执行先后顺序,使其满足测试用例场景设计,最终生成测试执行文件;所述测试运行模块,按照测试执行的参数命令运行测试执行文件后在制定位位置生成固定格式的测试报告。

一种维护成本较低的软件自动化测试方法及系统

技术领域

[0001] 本发明涉及一种维护成本较低的软件自动化测试方法及系统,属于计算机软件技术领域。

背景技术

[0002] 现今计算机软件行业飞速发展,软件的数量和规模也越来越大,逐步实现软件测试的自动化已经成为行业趋势。但是软件自动化测试的方法却依旧延续着手工测试的思维模式,将测试用例逐条编写为测试程序,之后再逐条执行。在这种方式开发出的自动化测试程序中,被测试的对象,要执行的测试动作,要输入和预期结果的数据都混杂在每一条测试用例中,而且很多测试对象、动作和数据都重复出现在多个不同的用例中。被测软件修改更新一个元素,在软件测试中针对一个元素需要从不同维度进行测试,就会产生多条测试程序,还会存在其他元素的测试中也使用到这个元素的情况。所有涉及该元素的测试程序全部都需要逐条进行修改,而且从数千条测试程序中检索出所有相关测试程序避免遗漏也是比较困难的。还存在随着测试工作的深入,需要有新的测试数据、测试动作函数的加入和旧测试数据、测试动作函数的更新等维护工作,这也需要逐条对涉及的测试程序进行修改。这样逐条更新维护的方式造成维护测试程序消耗的人力成本非常高,甚至常常超过被测软件本身的修改成本,这显然是得不偿失的,这也是造成软件自动化测试项目实施失败常见的原因之一。

发明内容

[0003] 本发明解决的技术问题是:克服现有技术的不足,提供了一种维护成本较低的软件自动化测试方法及系统,通过降低了测试用例、测试数据、测试动作、测试对象之间的耦合度,提高了自动化测试程序代码的复用程度,解决上述背景技术中提出的问题。

[0004] 本发明的技术解决方案是:一种维护成本较低的软件自动化测试方法,包括如下步骤:

[0005] S1,标识和描述测试对象;

[0006] S2,获取测试用例,拆分出测试动作和测试数据并进行唯一标识和描述;

[0007] S3,通过组合S1、S2所述测试对象、测试数据、测试动作生成测试脚本;

[0008] S4,通过组织排列S3所述测试脚本生成测试执行文件;

[0009] S5,通过运行S4所述测试执行文件生成测试报告。

[0010] 进一步地,所述S1中,使用唯一名称及ID命名测试对象,并使用标准化的程序语言描述测试对象的位置或从属关系以唯一确定该测试对象。

[0011] 进一步地,所述S2中,从原始测试用例中筛选出的用于自动化测试的用例。

[0012] 进一步地,所述S2中,从自动化测试用例中拆分出测试数据,并使用结构化的数据格式进行存储。

[0013] 进一步地,所述S2中,从自动化测试用例中拆分出测试动作,并使用程序语言进行

描述和实现。

[0014] 一种维护成本较低的软件自动化测试系统,包括:

[0015] 测试对象管理模块,对测试对象进行唯一标识和描述,存入测试对象库;

[0016] 测试数据管理模块,对测试用例进行抽象和拆分形成测试数据并进行唯一标识和描述,存入测试数据文件库;

[0017] 测试动作函数管理模块,对测试用例进行抽象和拆分形成测试动作并进行唯一标识和描述,存入测试动作函数库;

[0018] 测试脚本生成模块,根据测试需求拼装组合测试对象、测试数据、测试动作生成测试脚本文件;

[0019] 测试项组装模块,组织测试脚本执行顺序,生成测试执行文件;

[0020] 测试运行模块,运行测试执行文件后生成测试执行结果报告。

[0021] 进一步地,所述测试对象管理模块采用标准化的程序语言描述和唯一标识被测试的对象,存入测试对象库;

[0022] 所述测试数据管理模块横向分析和归纳自动化测试用例中使用的测试数据,以键值对和列表格式将其存储于结构化的数据文件中;

[0023] 所述测试动作函数管理模块分析自动化测试用例,抽象出测试动作,并使用程序语言描述实现这个动作,存入测试动作函数库。

[0024] 进一步地,所述测试脚本生成模块从测试对象库、测试数据文件库和测试动作函数库中取出测试对象、测试数据、测试动作,根据原始测试用例的具体需求进行组合,生成测试脚本文件。

[0025] 进一步地,所述测试脚本生成模块以数据驱动的方式来组织测试脚本,具体为:构建一个测试模板脚本,将测试数据以数据集的方式提供给该模板脚本,产生满足多项测试的脚本。

[0026] 进一步地,所述测试项组装模块用于调整测试脚本执行先后顺序,使其满足测试用例场景设计,最终生成测试执行文件;所述测试运行模块,按照测试执行的参数命令运行测试执行文件后在制定位置生成固定格式的测试报告。

[0027] 本发明与现有技术相比的优点在于:

[0028] 本发明通过使用所述一种维护成本较低的软件自动化方法及系统,解除了普遍存在于以往自动化测试中测试对象、测试数据、测试动作以及测试用例之间紧密耦合的现象,实现了测试代码和测试数据的高复用。测试开发人员完成测试系统的建立和基础测试对象、数据、动作的添加后,遇到被测试软件进行版本升级更新或者新的测试需求要加入自动化测试,仅需在所述系统相应的模块去更新管理相关的内容,无需逐条修改测试脚本。极大的减少了自动化测试维护的工作量,显著的降低了自动化测试维护所需的人力成本。

附图说明

[0029] 图1是本发明的测试系统总体实现框图;

[0030] 图2是本发明实施例中的一种运行架构图。

具体实施方式

[0031] 为了更好的理解上述技术方案,下面通过附图以及具体实施例对本申请技术方案做详细的说明,应当理解本申请实施例以及实施例中的具体特征是对本申请技术方案的详细的说明,而不是对本申请技术方案的限定,在不冲突的情况下,本申请实施例以及实施例中的技术特征可以相互组合。

[0032] 以下结合说明书附图对本申请实施例所提供的一种维护成本较低的软件自动化测试方法及系统做进一步详细的说明,具体实现方式可以包括(如图1~2所示):

[0033] S1,标识和描述测试对象;

[0034] S2,获取测试用例,拆分出测试动作和测试数据并进行唯一标识和描述;

[0035] S3,通过组合S1、S2所述测试对象、测试数据、测试动作生成测试脚本;

[0036] S4,通过组织排列S3所述测试脚本生成测试执行文件;

[0037] S5,通过运行S4所述测试执行文件生成测试报告。

[0038] 所述S1使用唯一名称及ID命名测试对象,并使用标准化的程序语言描述测试对象的位置或从属关系等内容以唯一确定这个测试对象。

[0039] 所述S2从原始测试用例中筛选出的用于自动化测试的用例。

[0040] 所述S2通过分析和抽象的方法从自动化测试用例中拆分出测试数据,并使用结构化的数据格式进行存储。

[0041] 所述S2通过分析和抽象的方法从自动化测试用例中拆分出测试动作,并使用程序语言进行描述和实现。

[0042] 所述S3使用S1及S2中生成的测试对象、测试动作和测试数据组合成测试脚本。

[0043] 所述S4使用S3中生成的测试脚本,根据测试需求排列测试脚本生成测试执行文件。

[0044] 所述S5使用S4中生成的测试执行文件运行测试并生成测试报告。

[0045] 一种维护成本较低的软件自动化测试系统,用于实现软件自动化测试,包括:

[0046] 测试对象管理模块,对测试对象进行唯一标识和描述,存入测试对象库;

[0047] 测试数据管理模块,对测试用例进行抽象和拆分形成测试数据并进行唯一标识和描述,存入测试数据文件库;

[0048] 测试动作函数管理模块,对测试用例进行抽象和拆分形成测试动作并进行唯一标识和描述,存入测试动作函数库;

[0049] 测试脚本生成模块,根据测试需求拼装组合测试对象、测试数据、测试动作生成测试脚本文件;

[0050] 测试项组装模块,组织测试脚本执行顺序,生成测试执行文件;

[0051] 测试运行模块,运行测试执行文件后生成测试执行结果报告。

[0052] 所述测试对象管理模块采用标准化的程序语言描述和唯一标识被测试的对象,存入测试对象库。

[0053] 所述测试数据管理模块横向分析和归纳自动化测试用例中使用的测试数据,以键值对(Key-Value)和列表等格式将其存储于结构化的数据文件中。

[0054] 所述测试动作函数管理模块分析自动化测试用例,抽象出测试动作,并使用程序语言描述实现这个动作,存入测试动作函数库。

[0055] 所述测试脚本生成模块从测试对象库、测试数据文件库和测试动作函数库中取出测试对象、测试数据、测试动作,根据原始测试用例的具体需求进行组合,生成测试脚本文件。

[0056] 所述测试脚本生成模块支持以数据驱动的方式来组织测试脚本,即构建一个测试模板脚本,将测试数据以数据集的方式提供给这个模板脚本,从而一次性的产生满足多项测试的脚本。

[0057] 所述测试项组装模块用于调整测试脚本执行先后顺序,使其满足测试用例场景设计,最终生成测试执行文件。

[0058] 所述测试运行模块,按照测试执行的参数命令运行测试执行文件后在制定位置生成固定格式的测试报告。

[0059] 具体的,在本申请实施例提供的技术方案中,被测试软件作为系统的输入,进入测试对象管理模块中,分析被测试软件的组件和元素,从中挑选出测试中需要用到的对象,将这些对象存储到测试对象库中,测试对象库通常是指按照被测试软件的界面结构组织划分的一系列包含测试对象标识和描述的数据文件。文件的具体格式是,对象名称作为一列,对象描述作为一列,两者是一一对应的。对象名称是对象的唯一标识,对象描述是描述对象的特征,用于识别和确认这个对象,特征包括对象的相对或绝对位置、样式、属性等等。

[0060] 例如:用户登录界面,包含的测试对象有UserNameTextInput、UserPasswordTextInput、LoginButton、CancelButton等,使用xpath描述对象的相对或绝对位置。

	//测试对象标识	测试对象特征描述
	<code>#{UserNameTextInput}</code>	<code>xpath=//*[@id=login]/div/from/div/div</code>
	<code>/input</code>	
[0061]	<code>#{UserPasswordTextInput}</code>	<code>xpath=//*[@id=login]/div/from/div[2]/</code>
	<code>div/input</code>	
	<code>#{LoginButton}</code>	<code>xpath=//*[@id=login]/div/from/div[3]/div /button[1]</code>
	<code>#{CancelButton}</code>	<code>xpath=//*[@id=login]/div/from/div[3]/div</code>
	<code>/button[2]</code>	

[0062] 如图1所示,测试对象管理模块可以使用名称、ID、Key等唯一标识对测试对象库中的存储的对象进行检索,还可以对相关内容进行增加、编辑、删除等操作。

[0063] 如图1所示,从原始测试用例中根据自动化测试的需求选取必要的部分作为自动化测试用例。以自动化测试用例作为系统输入,通过对自动化测试用例的归纳和分析,采集出测试中用到的数据,在测试数据管理模块中进行格式化的处理并存储到测试数据库中。其中通常情况下测试数据库是指用于存储测试数据的格式化的数据文件,在测试数据的结构关系复杂或者数据量较大时也使用数据库服务器作为测试数据库。在使用格式化数据文件来存储测试数据时,通常使用数组、列表或、字典的形式对数据进行组织。

[0064] 例如:用户信息相关的测试数据通常使用如下格式进行存储。

//用户信息字典

&{user} adminAvailabe=user1

... adminWrongPassword=user2

//用户信息数据

[0065] @{user1} \${adminName} \${adminPassword}

 @{user2} \${adminName} password000

//数据变量

 \${adminName} admin

 \${adminPassword} password123

[0066] 如图1所示,测试数据管理模块可以使用名称、ID、Key等唯一标识对测试数据库中的存储的数据进行检索,还可以对相关内容进行增加、编辑、删除等操作。

[0067] 如图1所示,以自动化测试用例作为系统输入,通过对自动化测试用例中的测试动作行为进行抽象和分析,在测试动作管理模块中使用唯一标识命名这个测试动作,并且使用程序函数实现该测试的一系列操作动作,并且抽象出需要用到的对象、测试数据作为函数的参数,存储到测试动作库中,测试动作库通常是指按照被测试软件的界面结构组织划分的一系列包含测试动作函数的数据文件。

[0068] 例如:用户登录的测试用例中,我们可以抽象出一个Login函数,测试数据是用户信息,函数的具体操作是将用户信息输入软件的

Login

[0069] [Argument] \${username} \${password}

[Documentation] 登录函数, 参数为: 用户名, 密码

Open Browser \${IP_ADDRESS} gc

Maxinize Browser Window

[0070] Input Text \${UserNameTextinput} \${username}

Input Text \${UserPasswordTextinput} \${password}

Click Button \${LoginButton}

[0071] 根据实际需求和可以在函数内增加成功失败的判断等功能,或编写其他函数调用此函数进行进一步操作。

[0072] 如图1所示,测试动作函数模块可以使用名称、ID、Key等唯一标识对测试动作函数库中的存储的动作函数进行检索,还可以对相关内容进行增加、编辑、删除等操作。

[0073] 如图1所示,以测试对象库、测试数据库、测试函数库中的对象、数据、函数作为输入,在测试脚本生成模块中根据自动化测试用例进行组合,生成一个个测试脚本,每个测试脚本可以执行一系列测试,具体的复杂程度以自动化测试用例的要求为准,通常采用一个测试脚本对应一个测试用例或者一个测试脚本对应一系列紧密相关的测试用例这两种方式。在测试脚本的实现中还可采用函数模板的形式进行脚本内容的组织,以便实现数据驱动测试模式。

[0074] 例如:

//普通测试用例

```
testcase1_app_login
```

```
Login    ${user1}
```

```
testcase2_app_login
```

[0075]

```
Login    ${user2}
```

//数据驱动, 使用函数模板

```
testcase_app_login_temp
```

```
[Setup]
```

```
[Template]    Login
```

[0076]

```
@${ user. adminAvailabe}}
```

```
@${ user. adminWrongPassword}}
```

[0077] 如图1所示,以测试脚本为输入,在测试项组装模块中,根据测试场景设计调整测试脚本的顺序,将相互关联或者有先后逻辑关系的测试脚本按照合理的顺序组合在一起,并且设置每个脚本执行的参数信息,形成一个或多个测试执行文件。

[0078] 如图1所示,以测试执行文件为输入,测试执行模块按照测试执行文件中设置的参数,调用测试工具包、测试框架和底层程序运行容器,编译运行测试程序并在指定位置生成测试报告。

[0079] 例如:以Python作为测试程序底层语言,RobotFramework作为测试框架,Selenium作为界面测试工具包,具体实现架构如图2所示。

[0080] 本申请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指令,当所述计算机指令在计算机上运行时,使得计算机执行图1所述的方法。

[0081] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实

施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器和光学存储器等)上实施的计算机程序产品的形式。

[0082] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0083] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0084] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0085] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

[0086] 本发明说明书中未作详细描述的内容属本领域技术人员的公知技术。

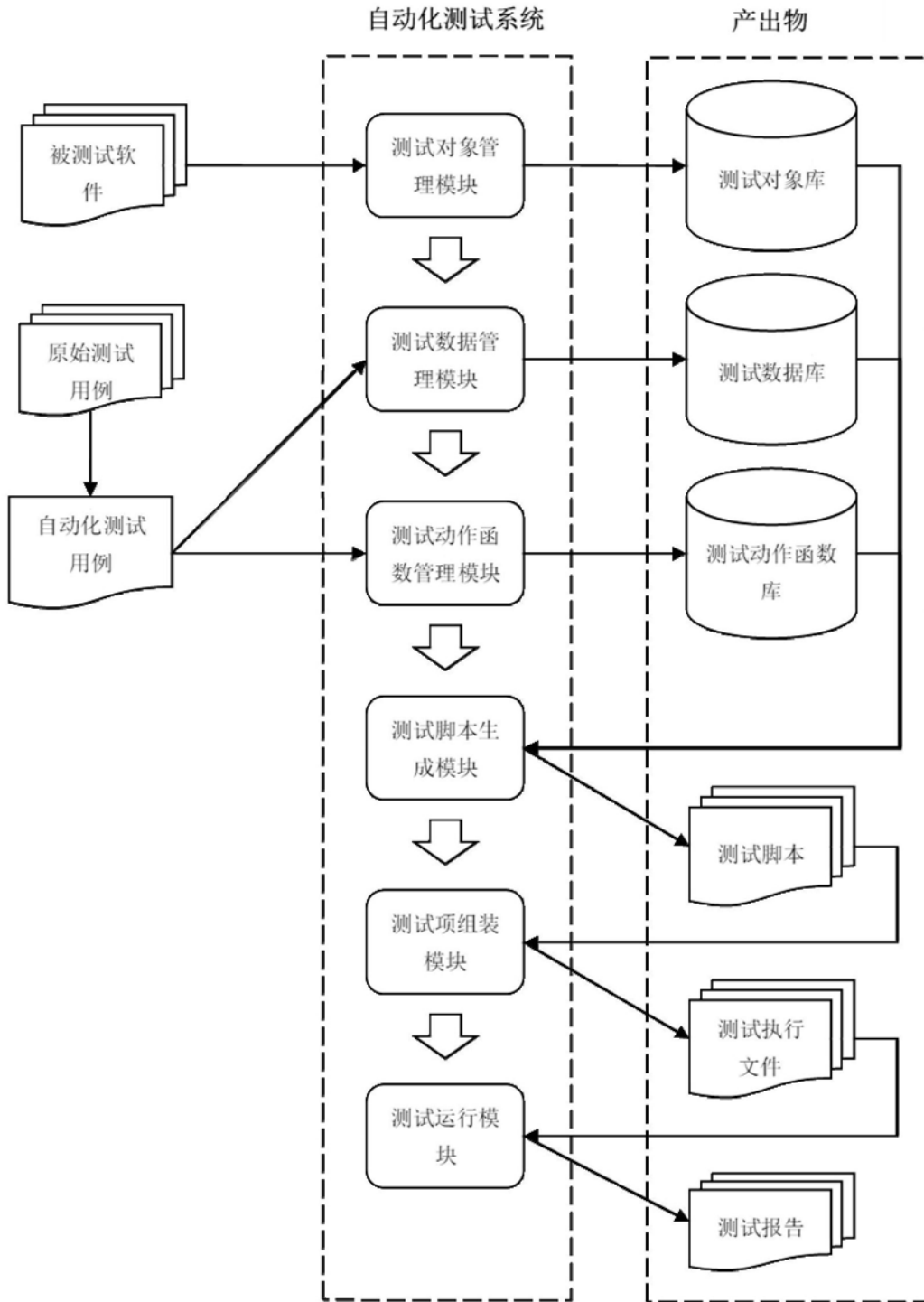


图1

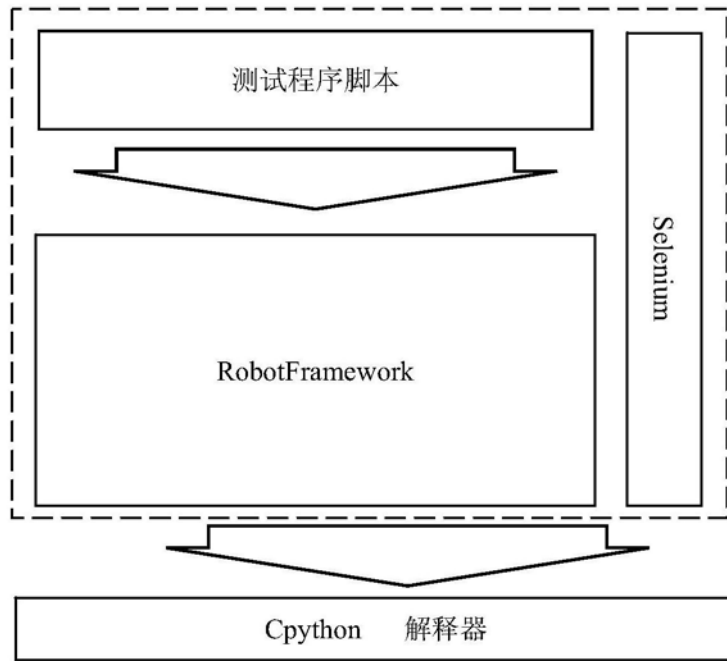


图2