



(12)发明专利申请

(10)申请公布号 CN 108491319 A

(43)申请公布日 2018.09.04

(21)申请号 201810143133.2

(22)申请日 2018.02.11

(71)申请人 广州小百合信息技术有限公司
地址 510663 广东省广州市萝岗区东荟二街81号320房

(72)发明人 吴庆庆 颜专 谢凯凯

(74)专利代理机构 广州广典知识产权代理事务
所(普通合伙) 44365
代理人 谢伟

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

权利要求书1页 说明书4页 附图3页

(54)发明名称

Web自动化测试方法及基于该方法的系统

(57)摘要

本发明公开了一种Web自动化测试方法及基于该方法的系统,Web自动化测试方法,包括搭建Selenium的自动化测试框架环境,基于该环境将测试代码划分为数据层、对象层、业务层的形式进行自动化测试,构建数据层:获取和/或存储测试用例的断言标准数据、测试结果数据、输入数据,被配置为供业务层调用;构建对象层:建立页面元素名称与页面元素的对应关系表,将Selenium的查找元素的方法进行二次封装,并返回给页面元素对象;构建业务层:利用对象层建立的页面元素名称与页面元素的对应关系表构建页面元素名称操作的对应表,建立和/或解析页面元素名称执行项的顺序表。本发明通过分层控制,使数据、对象、业务层相互独立,分布清楚,便于维护和修改,灵活性强。

构建数据层:
获取和/或存储测试用例的断言标准数据、测试结果数据、输入数据,被配置为供业务层调用

构建业务层:
利用对象层建立的页面元素名称与页面元素的对应关系表构建页面元素名称操作的对应表,建立和/或解析页面元素名称执行项的顺序表

构建对象层:
建立页面元素名称与页面元素的对应关系表,将Selenium的查找元素的方法进行二次封装,并返回给页面元素对象

1. Web自动化测试方法,其特征在于,包括搭建Selenium的自动化测试框架环境,将测试代码划分为数据层、对象层、业务层的形式进行自动化测试,

构建数据层:获取和/或存储测试用例的断言标准数据、测试结果数据、输入数据,被配置为供业务层调用;

构建对象层:建立页面元素名称与页面元素的对应关系表,将Selenium的查找元素的方法进行二次封装,并返回给页面元素对象;

构建业务层:利用对象层建立的页面元素名称与页面元素的对应关系表构建页面元素名称操作的对应表,建立和/或解析页面元素名称执行项的顺序表。

2. 如权利要求1所述的Web自动化测试方法,其特征在于,所述构建对象层的步骤中,Selenium的查找元素的方法至少包括通过ID、Class、Css Selector、Xpath的查找方法中的一种。

3. 如权利要求1所述的Web自动化测试方法,其特征在于,所述构建业务层的步骤中,页面解析页面元素名称执行项的顺序表具体为:解析其中的每一个执行项,实现该执行项对应的每一个页面元素及页面元素的操作,待执行项完成之后获取呈现的运行结果,将获取呈现的运行结果与数据层的标准数据进行对比,输出报告。

4. 如权利要求1-3任一项所述的Web自动化测试方法,其特征在于,所述页面元素名称操作的对应表可自定义。

5. 如权利要求4所述的Web自动化测试方法,所述页面元素名称操作的对应表可根据用户需求和/或功能自定义。

6. 如权利要求1-3任一项所述的Web自动化测试方法,其特征在于,所述Selenium的自动化测试框架的编译环境至少为Java、Python、C或C++中的一种。

7. 基于Web自动化测试方法的系统,其特征在于,包括数据层、对象层、业务层,

所述数据层获取测试用例的断言标准数据、测试结果数据、输入数据,被配置为供业务层调用;

所述对象层建立页面元素名称与页面元素的对应关系表,将Selenium的查找元素的方法进行二次封装,并返回给页面元素对象;

所述业务层利用对象层建立的页面元素名称与页面元素的对应关系表构建页面元素名称操作的对应表,建立和/或解析页面元素名称执行项的顺序表。

8. 如权利要求7所述的基于Web自动化测试方法的系统,其特征在于,所述基于Web自动化测试方法的系统基于BS架构或者CS架构。

Web自动化测试方法及基于该方法的系统

技术领域

[0001] 本发明涉及Web自动化测试领域,尤其涉及基于Selenium的Web自动化测试方法及基于该方法的系统。

背景技术

[0002] 传统Web测试采用录制测试用例-回放工具进行自动化测试,效率低。例如,CN 103678105 A提供了一种页面代码的测试方法和装置,其中,方法包括:S1、构造测试用例,其中,所述测试用例包括:构造或录制的后台模块响应数据、录制的页面测试操作和页面区域基准图像;S2、回放所述录制的页面测试操作,拦截被测页面向后台模块发送的请求并根据该请求向被测页面返回所述构造或录制的后台模块响应数据;S3、截取页面区域测试图像,并将所述页面区域测试图像与所述页面区域基准图像对比检验测试结果。该测试方法当系统功能发生变更时,需要修改大量的录制脚本,适应性差。

[0003] 当前,Web自动化测试框架主要有以下两种:1、基于QTP(QuickTest Professional)和QC(Quality Center)的自动化测试框架;2、基于Selenium的未分层的自动化测试框架。基于Selenium的未分层的自动化测试框架,未将测试数据、页面对象和方法、测试场景三者分层,主要有以下不足:测试场景和页面对象、页面方法相互包含,不方便进行参数化的数据驱动测试;在较大项目进行实际应用会很困难,维护成本非常高;测试人员必须非常了解业务系统才能编写脚本;该自动化测试框架一般用于回归,不能用于新功能检测。

[0004] 此外,采用Selenium IDE在执行单个的测试用例(testCase)都是过程化的,过程中有数据,有页面对象,有断言标准,较为混乱。若测试对象对于数据规模有一定的需求,当测试系统改变时,数据维护很难统一。

发明内容

[0005] 本发明的目的在于提供一种Web自动化测试方法,以解决现有Web自动化测试技术的效率低,测试数据、页面对象和方法、测试场景数据维护难,适应性差的问题。

[0006] 其技术方案如下:

[0007] Web自动化测试方法,包括搭建Selenium的自动化测试框架环境,基于该环境将测试代码划分为数据层、对象层、业务层的形式进行自动化测试,

[0008] 构建数据层:获取和/或存储测试用例的断言标准数据、测试结果数据、输入数据,被配置为供业务层调用;

[0009] 构建对象层:建立页面元素名称与页面元素的对应关系表,将Selenium的查找元素的方法进行二次封装,并返回给页面元素对象;

[0010] 构建业务层:利用对象层建立的页面元素名称与页面元素的对应关系表构建页面元素名称操作的对应表,建立和/或解析页面元素名称执行项的顺序表。

[0011] 可选的,所述构建对象层的步骤中Selenium的查找元素的方法至少包括通过ID、

Class、Css Selector、Xpath的查找方法中的一种。

[0012] 在本发明的一个实施例中,所述构建业务层的步骤中,页面解析页面元素名称执行项的顺序表具体为:解析其中的每一个执行项,实现该执行项对应的每一个页面元素及页面元素的操作,待执行项完成之后获取呈现的运行结果,将获取呈现的运行结果与数据层的标准数据进行对比,输出报告。

[0013] 在本发明的一个实施例中,所述页面元素名称操作的对应表可自定义。

[0014] 进一步的,所述页面元素名称操作的对应表可根据用户需求和/或功能自定义。

[0015] 在本发明的一个实施例中,所述Selenium的自动化测试框架的编译环境至少为Java、Python、C或C++中的一种。

[0016] 基于Web自动化测试方法的系统,包括数据层、对象层、业务层,

[0017] 所述数据层获取测试用例的断言标准数据、测试结果数据、输入数据,被配置为供业务层调用;

[0018] 所述对象层建立页面元素名称与页面元素的对应关系表,将Selenium的查找元素的方法进行二次封装,并返回给页面元素对象;

[0019] 所述业务层利用对象层建立的页面元素名称与页面元素的对应关系表构建页面元素名称操作的对应表,建立和/或解析页面元素名称执行项的顺序表。

[0020] 优选的,所述基于Web自动化测试方法的系统基于BS或者CS架构。

[0021] 下面对本发明的优点或原理进行说明:

[0022] 1、Web自动化测试方法,将数据、测试对象、业务层(测试逻辑)分层控制,使数据和业务逻辑分布更清晰,便于测试需求变化时维护;各层相互独立,互不干扰,若需求发生变化,即可根据修改对应的层级的部分,而无需对整个测试框架进行修改。

[0023] 2、由于页面元素名称与页面元素的对应关系表、页面元素名称操作的对应表可自定义,页面元素和页面元素的操作可扩展,从而可以增加更多的能测试的Web页面元素控件;测试不再限制在冒烟测试与回归测试,可进行新功能验证。

[0024] 3、基于Web自动化测试方法的系统的分层结构支持人工定义执行步骤,定义输入数据,定义测试结果的标准(文字或图片),以驱动自动化测试。

[0025] 4、在Web自动化测试方法及基于该方法的系统中,采用不同的Selenium的自动化测试框架结合来提高测试方法或系统的灵活性和适应性。

附图说明

[0026] 图1是本发明实施例Web自动化测试方法的基本流程图;

[0027] 图2是本发明实施例Web自动化测试方法的业务层基本流程图;

[0028] 图3是本发明实施例Web自动化测试方法中将Selenium查找元素进行二次封装的部分代码示意。

具体实施方式

[0029] 下面对本发明的实施例进行详细说明。

[0030] 如图1所示,Web自动化测试方法,包括搭建Selenium的自动化测试框架环境,基于该环境将测试代码划分为数据层、对象层、业务层的形式进行自动化测试,常见的自动化测

试环境还包括由java或python、C、C++等其他编译语言搭建的测试环境，

[0031] 构建数据层：获取和/或存储测试用例的断言标准数据、测试结果数据、输入数据，其中测试用例的断言标准数据包括文字对比标准、图片对比标准，数据层构建完成后被封装为方法，供业务层调用；执行项的自然语言（项名称），自然语言（页面元素名称），以及自然语言（页面元素名称）动作的对应表，该表可以为数据库表，excel表，csv表等有一定对应关系数据存储形式；

[0032] 构建对象层：建立页面元素名称与页面元素的对应关系表，将Selenium的查找元素的方法进行二次封装（参考图3），并返回给页面元素对象；需要说明的是，封装方法并不限于图3中示出的部分；

[0033] 构建业务层：利用对象层建立的页面元素名称与页面元素的对应关系表构建页面元素名称操作的对应表，建立和/或解析页面元素名称执行项的顺序表。

[0034] 可选的，所述构建对象层的步骤中Selenium的查找元素的方法至少包括通过ID、Class、Css Selector、Xpath的查找方法中的一种。

[0035] 在本发明的一个实施例中，构建对象层具体为：建立页面元素名称（中文或其他语言）与页面元素的对应关系表，通过页面元素名称作为key可以在该表中查找页面元素，页面元素应有两个属性，一、查找方式id，class，xpath，cssselector等，二、查找的值。通过封装的selenium查找元素的方法，返回浏览器中页面元素的对象，提供给业务层调用；

[0036] 在本发明的一个实施例中，构建业务层具体为：建立执行测试项表，该表中包含所需测试项的名称，页面元素，以及元素操作。建立元素动作表，表中包含元素操作名称（中文或其他语言）和元素操作，通过封装Selenium中操作页面元素的方法，可以在获取到对象后，执行元素操作，达到操作Web的目的。

[0037] 需要说明的是，本发明中所指的Web包括web页面，webUI。

[0038] 如图2所示，在本发明的一个实施例中，所述构建业务层的步骤中，页面解析页面元素名称执行项的顺序表具体为：1、建立执行项的自然语言（项名称），自然语言（页面元素名称），以及自然语言（页面元素名称）操作的对应表，页面元素的操作通常是指鼠标的点击或键盘的输入，或者包括输入装置与页面进行数据交互的一系列动作；2、建立自然语言的执行项的顺序表，该表的内容为执行项的自然语言（项名称）；执行项顺序表进入循环器（循环模块），由执行模块判断是否运行该执行项；3、解析其中的每一个执行项，实现该执行项对应的每一个页面元素及其的操作，待执行项完成之后获取呈现的运行结果，将运行结果与数据层的标准数据进行对比，输出结果，其中标准数据包括文字，预先准备的UI截图；若未执行完毕则返回给循环器，由循环器判断执行项是否执行。循环执行完成执行项的顺序表中所有定义的动作和顺序，循环完毕，输出报告，呈现执行结果和异常信息至界面，并且将执行报告以邮件的方式发送给指定邮箱。

[0039] 在本发明的一个实施例中，所述页面元素名称操作的对应表可自定义。在已实现的步骤下，根据人工自定义执行步骤、输入数据（包括输入框、上传的文档）、测试结果的标准（文字或图片）；页面元素和页面元素的操作可以进行扩展，可以增加更多能测试的Web页面元素控件。进一步的，可以根据自动义的页面元素名称操作的对应表来验证新功能。

[0040] 在本发明的一个实施例中，为了提高自动化测试框架灵活性和适应性，基于Python的自动化测试框架至少为doctest、unittest、nose、pytest或tempest其中的一种。

例如unittest框架,通常包括testcase,testsuite,testloader,unittestrunner模块。1、testcase也就是测试用例,一般一个unittest的实例就是一个testcase,可以理解为测试的最小单元。2、testsuite即测试套,就是多个testcase的集合,多个testcase加载后,然后testsuite编排执行顺序。3、testloader即测试加载器,通过各种匹配原则来进行加载,创建测试用例的实例,然后返回。4、unittestrunner,用于执行测试用例的,执行之前testloader加载的那些实例。之后,通过HtmlTestRunner等报告生成框架生成和记录报告并通知测试人员测试结果,然后在WEB界面给出本地储存或者下载报告的链接,或以邮件的形式将报告发送至预定的邮箱中。

[0041] 一般地,在nosetests测试框架中,调用nosetests xxx,其中xxx可以是目录、包、文件等,nose可以识别这些目录、包、文件中的测试用例并执行。如果单独执行一个类中的测试方法,可以使用nosetests xxxx.py:TestClass.testmethod,其中xxxx.py是文件名,TestClass是类名,testmethod是要执行的方法。Doctest、pytest、tempest为现有Python的自动化测试框架,在此不再赘述。

[0042] 基于Web自动化测试方法的系统,包括数据层、对象层、业务层,

[0043] 所述数据层获取测试用例的断言标准数据、测试结果数据、输入数据,被配置为供业务层调用;

[0044] 所述对象层建立页面元素名称与页面元素的对应关系表,将Selenium的查找元素的方法进行二次封装,并返回给页面元素对象;

[0045] 所述业务层利用对象层建立的页面元素名称与页面元素的对应关系表构建页面元素名称操作的对应表,建立和/或解析页面元素名称执行项的顺序表。

[0046] 优选的,所述基于Web自动化测试方法的系统基于BS架构或者CS架构。例如,通过Python的socket方法或像JAWEB、Django、Tornado、Flask、Twiste等网络框架对数据层、对象层、业务层分别进行封装,从而实现BS或者CS架构。

[0047] 以上仅为本发明的具体实施例,并不以此限定本发明的保护范围;在不违反本发明构思的基础上所作的任何替换与改进,均属本发明的保护范围。

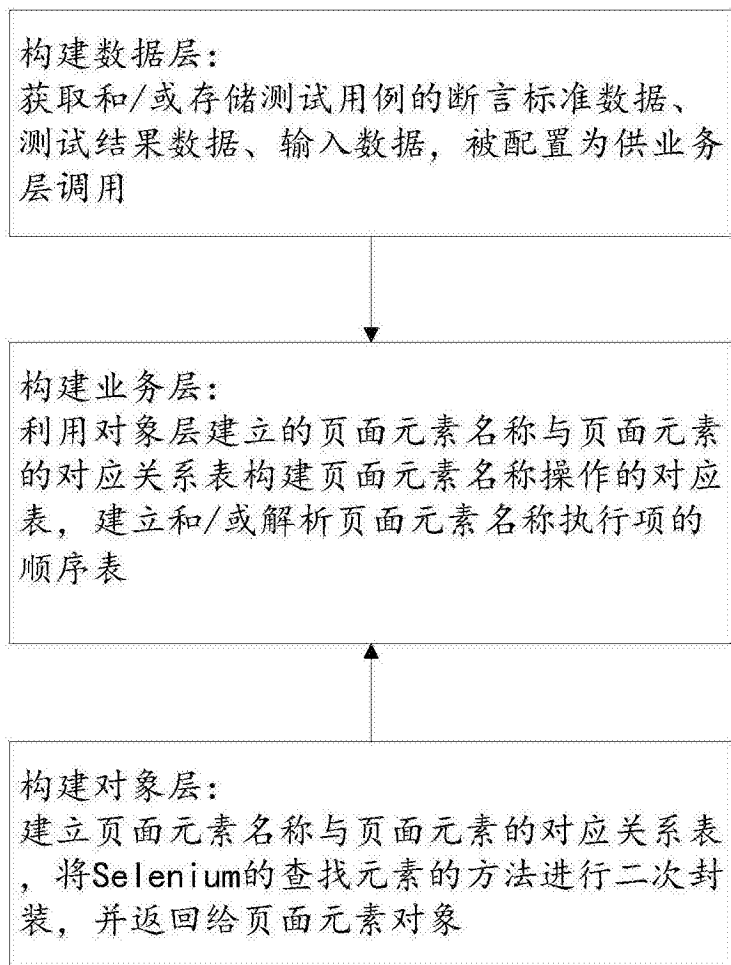


图1

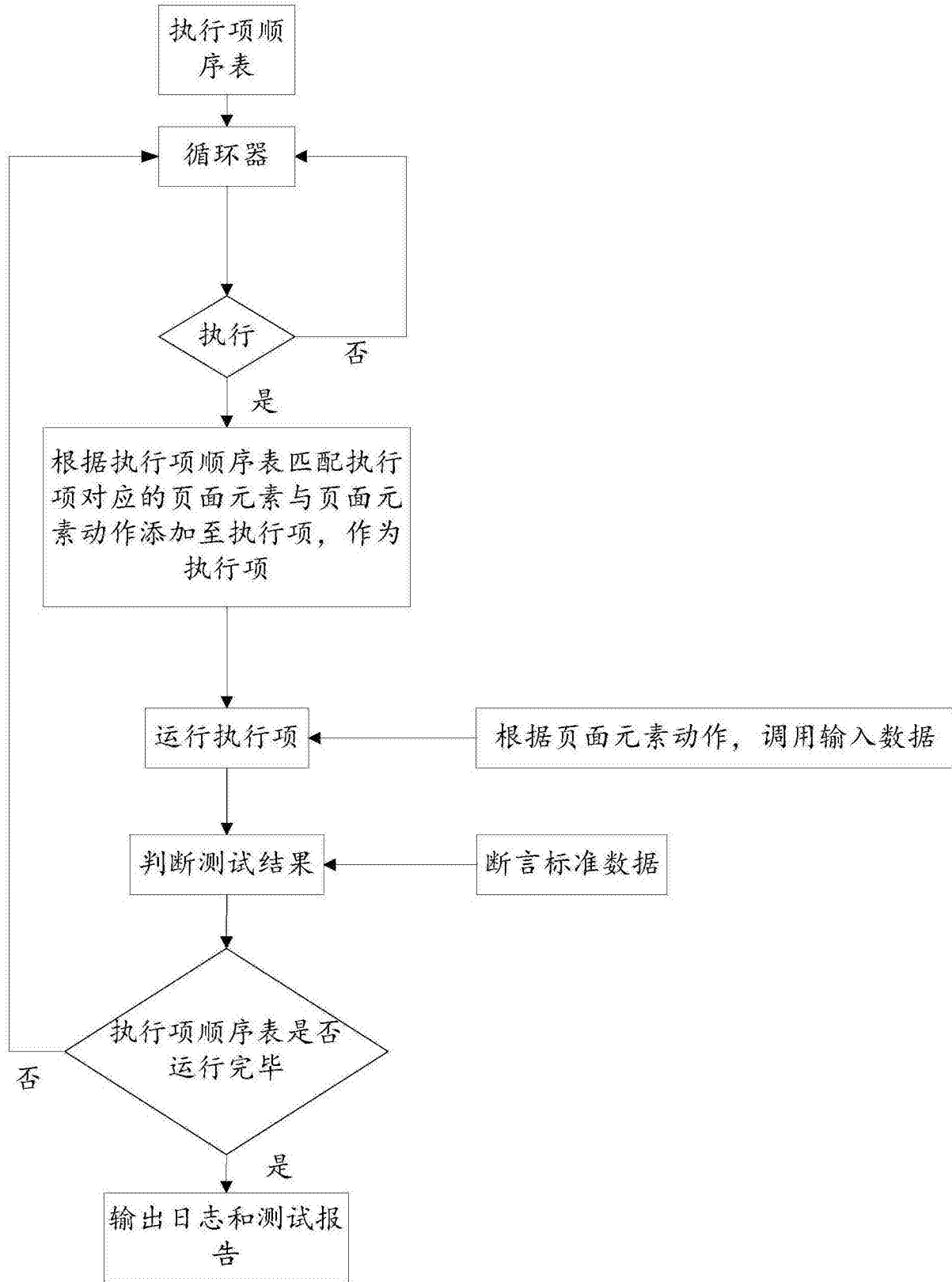


图2


```
def getElement(self,by,value):  
    """  
  
    :param by:      查找元素的方式  
    :param value:  文本值  
    :return:       查找到的元素  
    """  
  
    if by=="id":  
        return self.driver.find_element_by_id(value)  
    elif by=="class":  
        return self.driver.find_element_by_class_name(value)  
    elif by=="name":  
        return self.driver.find_element_by_name(value)  
    elif by=="css":  
        return self.driver.find_element_by_css_selector(value)  
    elif by=="linktext":  
        return self.driver.find_element_by_link_text(value)  
    elif by=="partialtext":  
        return self.driver.find_element_by_partial_link_text(value)  
    elif by=="tag":  
        return self.driver.find_element_by_tag_name(value)  
    elif by=="xpath":  
        return self.driver.find_element_by_xpath(value)  
    elif by=="js" or by=="jquery":  
        return self.ExcuteJs(value)  
    else:  
        print "请输入适合的查找元素方式。。。"
```

图3