



(12) 发明专利

(10) 授权公告号 CN 103019928 B

(45) 授权公告日 2016. 06. 29

(21) 申请号 201110296140. 4

US 5606661 A, 1997. 02. 25,

(22) 申请日 2011. 09. 27

审查员 陈沙沙

(73) 专利权人 北京新媒传信科技有限公司

地址 100089 北京市海淀区万泉庄路 28 号
万柳新贵大厦 A 座 5 层

(72) 发明人 刘显健 孙冬梅

(74) 专利代理机构 北京市隆安律师事务所

11323

代理人 权鲜枝

(51) Int. Cl.

G06F 11/36(2006. 01)

(56) 对比文件

CN 101187894 A, 2008. 05. 28,

CN 1851664 A, 2006. 10. 25,

CN 101266570 A, 2008. 09. 17,

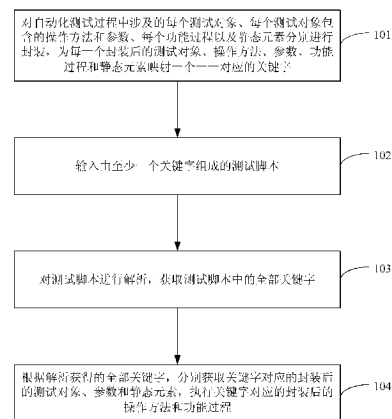
权利要求书2页 说明书6页 附图2页

(54) 发明名称

一种自动化测试方法和系统

(57) 摘要

本发明公开了一种自动化测试方法,对自动化测试过程中涉及的每个测试对象、每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装,为每一个封装后的测试对象、操作方法、参数、功能过程和静态元素映射一个一一对应的关键字,输入由至少一个关键字组成的测试脚本,对测试脚本进行解析,获取测试脚本中的全部关键字,根据解析获得的关键字,分别获取关键字对应的封装后的测试对象、参数和静态元素,执行关键字对应的封装后的操作方法和功能过程。本发明还公开了一种自动化测试系统。采用本发明公开的自动化测试方法和系统,能够提高自动化测试的效率和通用性。



1. 一种自动化测试方法,其特征在于,包括:

对自动化测试过程中涉及的每个测试对象、所述每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装,为每一个封装后的所述测试对象、所述操作方法、所述参数、所述功能过程和所述静态元素映射一个一一对应的关键字;

输入由至少一个关键字组成的测试脚本;

对所述测试脚本进行解析,获取所述测试脚本中的全部关键字;

根据解析获得的全部所述关键字,分别获取所述关键字对应的封装后的所述测试对象、所述参数和所述静态元素,执行所述关键字对应的封装后的所述操作方法和所述功能过程;

所述输入由至少一个关键字组成的测试脚本之前还包括:创建所述测试脚本;

所述创建所述测试脚本之后还包括:存储所述测试脚本;和/或,查看所述测试脚本;和/或,导出所述测试脚本;和/或,编辑所述测试脚本;和/或,删除所述测试脚本;和/或,关联至少两个所述测试脚本;和/或,查看测试脚本的执行结果。

2. 根据权利要求1所述的方法,其特征在于,

所述测试脚本采用伪代码。

3. 一种自动化测试系统,其特征在于,包括:

封装单元,用于对自动化测试过程中涉及的每个测试对象、所述每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装,为每一个封装后的所述测试对象、所述操作方法、所述参数、所述功能过程和所述静态元素映射一个一一对应的关键字;

浏览器,用于输入由至少一个关键字组成的测试脚本;

解析模块,用于对所述测试脚本进行解析,获取所述测试脚本中的全部关键字;

执行模块,用于根据所述解析模块获取的全部所述关键字,分别获取所述关键字对应的封装后的所述测试对象、所述参数和所述静态元素,执行所述关键字对应的封装后的所述操作方法和所述功能过程;

所述系统还包括:管理模块,用于在输入由至少一个关键字组成的测试脚本之前创建所述测试脚本;在创建所述测试脚本之后存储所述测试脚本、和/或查看所述测试脚本、和/或导出所述测试脚本、和/或编辑所述测试脚本、和/或删除所述测试脚本、和/或关联至少两个所述测试脚本、和/或查看测试脚本的执行结果。

4. 根据权利要求3所述的系统,其特征在于,

所述封装单元包括:底层支撑模块和对象工厂模块;

所述底层支撑模块,用于对自动化测试过程中涉及的每个测试对象、所述每个测试对象包含的操作方法和参数、以及每个功能过程分别进行封装,为每一个封装后的所述测试对象、所述操作方法、所述参数和所述功能过程映射一个一一对应的关键字;

所述对象工厂模块,用于存储经过封装的自动化测试过程中涉及的每个静态元素和每个所述静态元素对应的关键字;

所述执行模块,具体用于根据解析模块获得的至少一个关键字,获取底层支撑模块存储的所述至少一个关键字对应的测试对象、操作方法、参数和功能过程,分别执行所述操作方法和功能过程,在执行所述操作方法和功能过程的过程中,从所述对象工厂模块调用所

述操作方法和功能过程中涉及的静态元素。

5. 根据权利要求4所述的系统,其特征在于,所述底层支撑模块包括:测试对象接口、测试帮助接口、操作系统对象接口、应用程序用户接口;

所述测试对象接口用于实现窗口操作和/或控件操作;

所述测试帮助接口用于实现辅助功能;

所述操作系统对象接口用于提供系统对象;

所述应用程序用户接口用于实现应用程序的用户操作和进程操作。

6. 根据权利要求3所述的系统,其特征在于,所述管理模块包括管理接口;

所述管理接口用于所述管理模块与所述解析模块交互,以检查所述测试脚本的格式;

和/或,所述管理接口用于所述管理模块与所述执行模块交互,以触发所述执行模块启动执行测试脚本;

和/或,所述管理接口用于所述管理模块与所述浏览器交互,以创建所述测试脚本、和/或编辑所述测试脚本、和/或存储所述测试脚本、和/或查看所述测试脚本、和/或导出所述测试脚本、和/或删除所述测试脚本、和/或关联至少两个所述测试脚本。

7. 根据权利要求3所述的系统,其特征在于,所述解析模块包括解析接口;

所述解析接口连接所述执行模块,所述执行模块调用所述解析接口向所述解析模块传入所述测试脚本并从所述解析模块获取解析结果;

和/或,所述解析接口连接所述管理模块,所述管理模块调用所述解析接口向所述解析模块传入所述测试脚本并从所述解析模块获取格式检查结果和/或解析结果;

和/或,所述解析接口连接所述浏览器,所述浏览器通过所述解析接口向所述解析模块传入所述测试脚本。

8. 根据权利要求3所述的系统,其特征在于,

所述测试脚本采用伪代码。

一种自动化测试方法和系统

技术领域

[0001] 本发明涉及计算机技术,特别是涉及一种自动化测试方法和系统。

背景技术

[0002] 目前,在进行软件测试时,常采用自动化测试的方法。自动化测试是把以人为驱动的测试行为转化为机器执行的一种过程。如果采用非自动化测试的方法,则在设计了测试用例并通过评审之后,由测试人员根据测试用例中描述的规程一步步执行测试,得到实际结果与期望结果的比较。随着软件测试技术的发展,为了节省人力、时间、硬件资源,提高测试效率,提出了自动化测试方法。自动化测试与软件开发过程本质上是一致的,利用自动化测试工具,经过对测试需求的分析,设计出自动化测试用例,从而搭建自动化测试的框架,设计与编写自动化测试脚本,通过测试该脚本的正确性,从而完成该套测试脚本。

[0003] 目前的自动化测试方法中,预先编写测试框架,在该框架中为测试需要的基础操作提供接口以供调用,在测试时,测试人员根据用例操作需求,人工编写调用上述接口的自动化测试脚本。采用此方法,要求参与测试的脚本开发人员以及后期运行维护人员都必须具有很高的代码水平,自动化脚本的开发和运行维护的成本大。并且,采用此方法,在测试时需要人工编写脚本,从而导致测试过程所需时间长,测试效率低,并且,采用此方法,对测试人员的技术要求高,从而导致测试方法的通用性差。

发明内容

[0004] 本发明提供了一种自动化测试方法,采用该方法能够提高自动化测试的效率和通用性。

[0005] 本发明还提供了一种自动化测试系统,采用该系统能够提高自动化测试的效率和通用性。

[0006] 为达到上述目的,本发明的技术方案是这样实现的:

[0007] 本发明提供一种自动化测试方法,包括:

[0008] 对自动化测试过程中涉及的每个测试对象、所述每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装,为每一个封装后的所述测试对象、所述操作方法、所述参数、所述功能过程和所述静态元素映射一个一一对应的关键字;

[0009] 输入由至少一个关键字组成的测试脚本;

[0010] 对所述测试脚本进行解析,获取所述测试脚本中的全部关键字;

[0011] 根据解析获得的全部所述关键字,分别获取所述关键字对应的封装后的所述测试对象、所述参数和所述静态元素,执行所述关键字对应的封装后的所述操作方法和所述功能过程。

[0012] 所述输入由至少一个关键字组成的测试脚本之前还包括:创建所述测试脚本;

[0013] 所述创建所述测试脚本之后还包括:存储所述测试脚本;和/或,查看所述测试脚本;和/或,导出所述测试脚本;和/或,编辑所述测试脚本;和/或,删除所述测试脚本;和/

或,关联至少两个所述测试脚本;和/或,查看测试脚本的执行结果。

[0014] 所述测试脚本采用伪代码。

[0015] 本发明还提供一种自动化测试系统,包括:

[0016] 封装单元,用于对自动化测试过程中涉及的每个测试对象、所述每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装,为每一个封装后的所述测试对象、所述操作方法、所述参数、所述功能过程和所述静态元素映射一个一一对应的关键字;

[0017] 浏览器,用于输入由至少一个关键字组成的测试脚本;

[0018] 解析模块,用于对所述测试脚本进行解析,获取所述测试脚本中的全部关键字;

[0019] 执行模块,用于根据所述解析模块获取的全部所述关键字,分别获取所述关键字对应的封装后的所述测试对象、所述参数和所述静态元素,执行所述关键字对应的封装后的所述操作方法和所述功能过程。

[0020] 所述封装单元包括:底层支撑模块和对象工厂模块;

[0021] 所述底层支撑模块,用于对自动化测试过程中涉及的每个测试对象、所述每个测试对象包含的操作方法和参数、以及每个功能过程分别进行封装,为每一个封装后的所述测试对象、所述操作方法、所述参数和所述功能过程映射一个一一对应的关键字;

[0022] 所述对象工厂模块,用于存储经过封装的自动化测试过程中涉及的每个静态元素和每个所述静态元素对应的关键字;

[0023] 所述执行模块,具体用于根据解析模块获得的至少一个关键字,获取底层支撑模块存储的所述至少一个关键字对应的测试对象、操作方法、参数和功能过程,分别执行所述操作方法和功能过程,在执行所述操作方法和功能过程的过程中,从所述对象工厂模块调用所述操作方法和功能过程中涉及的静态元素。

[0024] 所述底层支撑模块包括:测试对象接口、测试帮助接口、操作系统对象接口、应用程序用户接口;

[0025] 所述测试对象接口用于实现窗口操作和/或控件操作;

[0026] 所述测试帮助接口用于实现辅助功能;

[0027] 所述操作系统对象接口用于提供系统对象;

[0028] 所述应用程序用户接口用于实现应用程序的用户操作和进程操作。

[0029] 所述装置还包括:

[0030] 管理模块,用于创建所述测试脚本、和/或存储所述测试脚本、和/或查看所述测试脚本、和/或导出所述测试脚本、和/或编辑所述测试脚本、和/或删除所述测试脚本、和/或关联至少两个所述测试脚本、和/或查看测试脚本的执行结果。

[0031] 所述管理模块包括管理接口;

[0032] 所述管理接口用于所述管理模块与所述解析模块交互,以检查所述测试脚本的格式;

[0033] 和/或,所述管理接口用于所述管理模块与所述执行模块交互,以触发所述执行模块启动执行测试脚本;

[0034] 和/或,所述管理接口用于所述管理模块与所述浏览器交互,以创建所述测试脚本、和/或编辑所述测试脚本、和/或存储所述测试脚本、和/或查看所述测试脚本、和/或导

出所述测试脚本、和/或删除所述测试脚本、和/或关联至少两个所述测试脚本。

[0035] 所述解析模块包括解析接口；

[0036] 所述解析接口连接所述执行模块，所述执行模块调用所述解析接口向所述解析模块传入所述测试脚本并从所述解析模块获取解析结果；

[0037] 和/或，所述解析接口连接所述管理模块，所述管理模块调用所述解析接口向所述解析模块传入所述测试脚本并从所述解析模块获取格式检查结果和/或解析结果；

[0038] 和/或，所述解析接口连接所述浏览器，所述浏览器通过所述解析接口向所述解析模块传入所述测试脚本。

[0039] 所述测试脚本采用伪代码。

[0040] 由上述发明内容可见，通过将自动化测试过程中涉及的每个测试对象、每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装，为每一个封装后的测试对象、操作方法、参数、功能过程和静态元素映射一个一一对应的关键字，测试人员输入所需测试的功能对应的关键字的组合，根据关键字执行该关键字对应的经过封装的上述内容，完成测试过程，从而不必在测试过程中编写针对本次测试的复杂的代码，提高了自动化测试的效率，并且，不必要求测试人员掌握现有的编程语言，提高了自动化测试方法和系统的通用性。

附图说明

[0041] 图1是本发明实施例一的自动化测试方法的流程图；

[0042] 图2是本发明实施例二的自动化测试系统的结构示意图。

具体实施方式

[0043] 为了使本发明的目的、技术方案和优点更加清楚，下面结合附图和具体实施例对本发明进行详细描述。

[0044] 本发明的基本思想是：将自动化测试过程中所需的每个功能对应的代码分别进行封装并映射一个关键字，在测试时，测试人员只需输入所需测试的功能对应的关键字或关键字的组合，即可根据关键字执行该关键字对应的经过封装的代码，从而完成测试过程。

[0045] 图1是本发明实施例一的自动化测试方法的流程图。如图1所示，该方法包括如下过程。

[0046] 步骤101：对自动化测试过程中涉及的每个测试对象、每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装，为每一个封装后的测试对象、操作方法、参数、功能过程和静态元素映射一个一一对应的关键字。

[0047] 在本步骤中，具体地，上述功能过程可以是一个或多个操作方法的组合。上述静态元素可以为窗口等静态元素，上述操作方法和功能过程中所需的静态元素可以从上述经过封装的静态元素中调去。

[0048] 步骤102：输入由至少一个关键字组成的测试脚本。

[0049] 在本步骤中，测试脚本中可以包括一个关键字，或者，也可以包括一组关键字组成的组合。该测试脚本可以由测试人员通过浏览器向自动化测试系统中输入，具体地，测试人员只需要根据当前的测试需求，采用该需求对应的关键字组成测试脚本即可。

[0050] 进一步地,在本发明实施例一中,该自动化测试方法中采用的测试脚本可以采用伪代码。具体地,该测试脚本可以不采用现有技术中的编程语言进行编写,而采用伪代码进行编写。由于现有的编程语言较复杂,通过采用更为简单的伪代码进行编写,因此不必要求测试人员掌握现有技术中的复杂的编程语言,而只需要通过伪代码对关键字进行简单组合即可。并且,采用伪代码无需对编写后的测试脚本进行编译,只需要按照预先定义的格式组装伪代码,通过测试系统对伪代码进行格式检查来确保伪代码的格式正确即可。通过采用伪代码编写测试脚本,进一步提高了测试方法的自动化测试效率和通用性。

[0051] 步骤103:对测试脚本进行解析,获取测试脚本中的全部关键字。

[0052] 在本步骤中,对测试脚本进行解析,通过解析获得步骤102中组成该测试脚本的一个或一组关键字。

[0053] 步骤104:根据解析获得的全部关键字,分别获取关键字对应的封装后的测试对象、参数和静态元素,执行关键字对应的封装后的操作方法和功能过程。

[0054] 在本步骤中,根据步骤103中的解析结果,组织执行流程,即组织执行测试脚本中的单条或多条脚本程序。具体地,在执行上述经过封装的操作或功能过程时,所需的静态元素可以从上述封装的静态元素中调取。

[0055] 在上述技术方案的基础上,进一步地,本发明实施例一的自动化测试方法还可以包括如下步骤中的一种或几种的组合。

[0056] 在步骤102之前,本发明实施例一的自动化测试方法中还可以包括:创建测试脚本。在创建测试脚本之后,本发明实施例一的自动化测试方法中还可以包括如下步骤中的一种或几种的组合:存储测试脚本、查看测试脚本、导出测试脚本、编辑测试脚本、删除测试脚本、关联至少两个测试脚本、查看测试脚本的执行结果。

[0057] 在上述技术方案的基础上,进一步地,在步骤102输入由至少一个关键字组成的测试脚本之后,在步骤103对测试脚本进行解析之前,本发明实施例一的自动化测试方法还可以包括以下步骤:检查测试脚本的格式。

[0058] 以上对本发明提出的自动化测试方法进行了说明,下面介绍一种采用上述自动化测试方法的自动化测试系统。

[0059] 图2是本发明实施例二的自动化测试系统的结构示意图。如图2所示,该系统至少包括:封装单元21、浏览器22、解析模块23和执行模块24,在该系统包括封装单元21、浏览器22、解析模块23和执行模块24的基础上,进一步地,该系统中还可以包括:管理模块25。

[0060] 其中,封装单元21对自动化测试过程中涉及的每个测试对象、每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装,为每一个封装后的测试对象、操作方法、参数、功能过程和静态元素映射一个一一对应的关键字。并且,封装单元21存储上述封装后的测试对象、操作方法、参数、功能过程和静态元素以及各自与关键字的对应关系。

[0061] 浏览器22用于为测试人员提供交互界面,测试人员通过该浏览器22向该自动化测试系统输入由至少一个关键字组成的测试脚本。具体地,该浏览器22可以采用浏览器22和服务器(Browser/Server,简称B/S)架构中的浏览器22。该浏览器22的界面可以分为两部分,其中,一部分根据系统中各个模块的需求定义控件和布局,另一部分给各个浏览器22控件添加控制逻辑。进一步地,在本发明实施例二中,该自动化测试系统中采用的测试脚本可

以采用伪代码。

[0062] 解析模块23对测试脚本进行解析,获取测试脚本中的全部关键字。进一步地,解析模块23还可以在对测试脚本进行解析之前,首先检查测试脚本的格式。解析模块23可以从浏览器22和/或执行模块24获取测试脚本,并将该测试脚本的解析结果或格式检查结果返回给浏览器22和/或执行模块24。

[0063] 执行模块24从解析模块23获取解析模块23解析获取的测试脚本中的全部关键字,根据至少一个关键字,从封装单元21分别获取关键字对应的封装后的测试对象、参数和静态元素,并且,从封装单元21获取并执行关键字对应的封装后的操作方法和功能过程。具体地,执行模块24根据解析模块23的解析结果,组织执行流程,即组织执行测试脚本中的单条或多条脚本程序。执行模块24中包括执行程序,该执行程序可以采用Java归档文件(Java Archive,简称JAR)包方式,使用启动Java浏览器(Java web start)的部署方式在浏览器22页面上调起。

[0064] 进一步地,该系统中还可以包括管理模块25。该管理模块25用于实现对脚本进行管理的功能,具体可以包括如下功能中的一项或多项的组合:创建测试脚本、存储测试脚本、查看测试脚本、导出测试脚本、编辑测试脚本、删除测试脚本、关联至少两个测试脚本、查看测试脚本的执行结果。通过管理模块25进行创建测试脚本或编辑测试脚本后,将经过管理模块25创建、编辑后的测试脚本输入到解析模块23。具体地,管理模块25从浏览器22获取上述创建、存储、查看、导出、编辑、删除、关联测试脚本和查看测试脚本的执行结果的命令,在浏览器22的后台采用JAVA服务网页技术(Java Server Pages,简称JSP)调用管理模块25,以完成上述创建、存储、查看、导出、编辑、删除、关联测试脚本并查看测试脚本的执行结果。

[0065] 在上述技术方案的基础上,进一步地,该系统中的封装单元21具体可以包括:底层支撑模块211和对象工厂模块212。

[0066] 具体地,底层支撑模块211对自动化测试过程中涉及的每个测试对象、每个测试对象包含的操作方法和参数、以及每个功能过程分别进行封装,为每一个封装后的测试对象、操作方法、参数和功能过程映射一个一一对应的关键字。

[0067] 对象工厂模块212存储经过封装的自动化测试过程中涉及的每个静态元素和每个静态元素对应的关键字。对象工厂模块212中的封装后的静态元素用于在底层支撑模块211中的操作和/或功能过程执行的过程中被调用。

[0068] 在封装单元21包括底层支撑模块211和对象工厂模块212的情况下,相应地,执行模块24根据从解析模块23获得的解析模块23解析获取的全部关键字,获取底层支撑模块211存储的至少一个关键字对应的测试对象、操作方法、参数和功能过程,按照测试脚本中的关键字顺序,分别执行关键字对应的操作方法和功能过程,在执行操作方法和功能过程的过程中,所需的静态元素从对象工厂模块212中调用。

[0069] 在上述技术方案的基础上,进一步地,自动化测试系统中的上述各个模块需要提供相应的接口,各个模块通过调用相应接口来完成自身的各种功能。例如:

[0070] 解析模块23中包括解析接口(IScriptParse接口),该解析接口用于解析测试脚本;该测试脚本可以采用伪代码编写,则该解析接口还可以进一步用于检查测试脚本的格式。该解析模块23通过该解析接口接收测试脚本。具体地,该解析接口可以应用于下述三种

场景中的任意一种或多种的组合：第一，该解析接口可以与执行模块24连接，执行模块24通过调用该解析接口向解析模块23传入测试脚本，并通过该解析接口获取解析结果，以根据解析结果执行测试脚本。第二，该解析接口可以与管理模块25连接，管理模块25通过调用该解析接口向解析模块23传入测试脚本，并通过该解析接口获取格式检查结果和/或解析结果。第三，该解析接口可以与浏览器22连接，通过该解析接口从浏览器22传入测试脚本。在实际应用中，可以根据具体应用的需要，设计该解析接口的连接对象，该解析接口可以与执行模块24、管理模块25和浏览器22中的任意一种连接，也可以与其中两者或三者连接。

[0071] 底层支撑模块211中包括：测试对象接口(ITestObject接口)、测试帮助接口(ITestAssistant接口)、操作系统对象接口(ISystemObject接口)和应用程序用户接口(IUser接口)。测试对象接口用于实现窗口操作和/或控件操作。测试帮助接口用于实现辅助功能，例如模拟鼠标键盘操作等辅助功能。操作系统对象接口用于提供系统对象，例如提供系统文件、注册表、时间、环境变量等系统对象。应用程序用户接口用于实现应用程序的用户操作和进程操作。进一步地，上述测试对象接口、操作系统对象接口和应用程序用户接口需要定义一个判断对象是否有效的静态方法，并且为每一个对象映射一个一一对应的关键字，以供伪码格式检查使用。

[0072] 管理模块25包括管理接口(IScripManagment接口)，上述管理模块25的功能均通过该管理接口或其扩展子接口实现。具体地，管理接口可以包括三部分功能。第一，管理模块25的管理接口用于与解析模块23交互，对测试脚本的格式进行检查。第二，管理模块25的管理接口用于与执行模块24交互，触发执行模块24启动执行测试脚本。第三，管理模块25通过管理接口与浏览器22交互，对测试脚本进行管理，例如，可以包括如下对测试脚本进行的管理操作：创建测试脚本、和/或编辑测试脚本、和/或存储测试脚本、和/或查看测试脚本、和/或导出测试脚本、和/或删除测试脚本、和/或关联至少两个测试脚本。在实际应用中，根据实际应用的需要，该管理接口可以包括上述三种功能中的任意一种或几种的组合。

[0073] 具体地，在浏览器22给各个网页控件或网络控件(简称Web控件)添加控制逻辑时，需要上述管理接口和解析接口的支持。

[0074] 在对象工厂模块212中，提供获取全部接口对象的静态方法。上述各个模块中，对其各自所依赖的接口均定义设置函数(简称setter)，由对象工厂在构造时传入依赖，实现依赖倒转。

[0075] 根据上述本发明具体实施方式可见，通过将自动化测试过程中所需的每个测试对象、所述每个测试对象包含的操作方法和参数、每个功能过程以及静态元素分别进行封装，为每一个封装后的测试对象、操作方法、参数、功能过程和静态元素映射一个一一对应的关键字，在测试时，测试人员只需输入所需测试的功能对应的关键字或关键字的组合，即可根据关键字执行该关键字对应的经过封装的代码，从而完成测试过程。采用该自动化测试方法和系统，不必在测试过程中编写针对本次测试的复杂的代码，从而提高了自动化测试的效率，并且，可以支持普通测试人员完成测试过程，而不必要求测试人员掌握现有的编程语言，从而提高了测试方法和系统的通用性。

[0076] 以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明保护的范围之内。

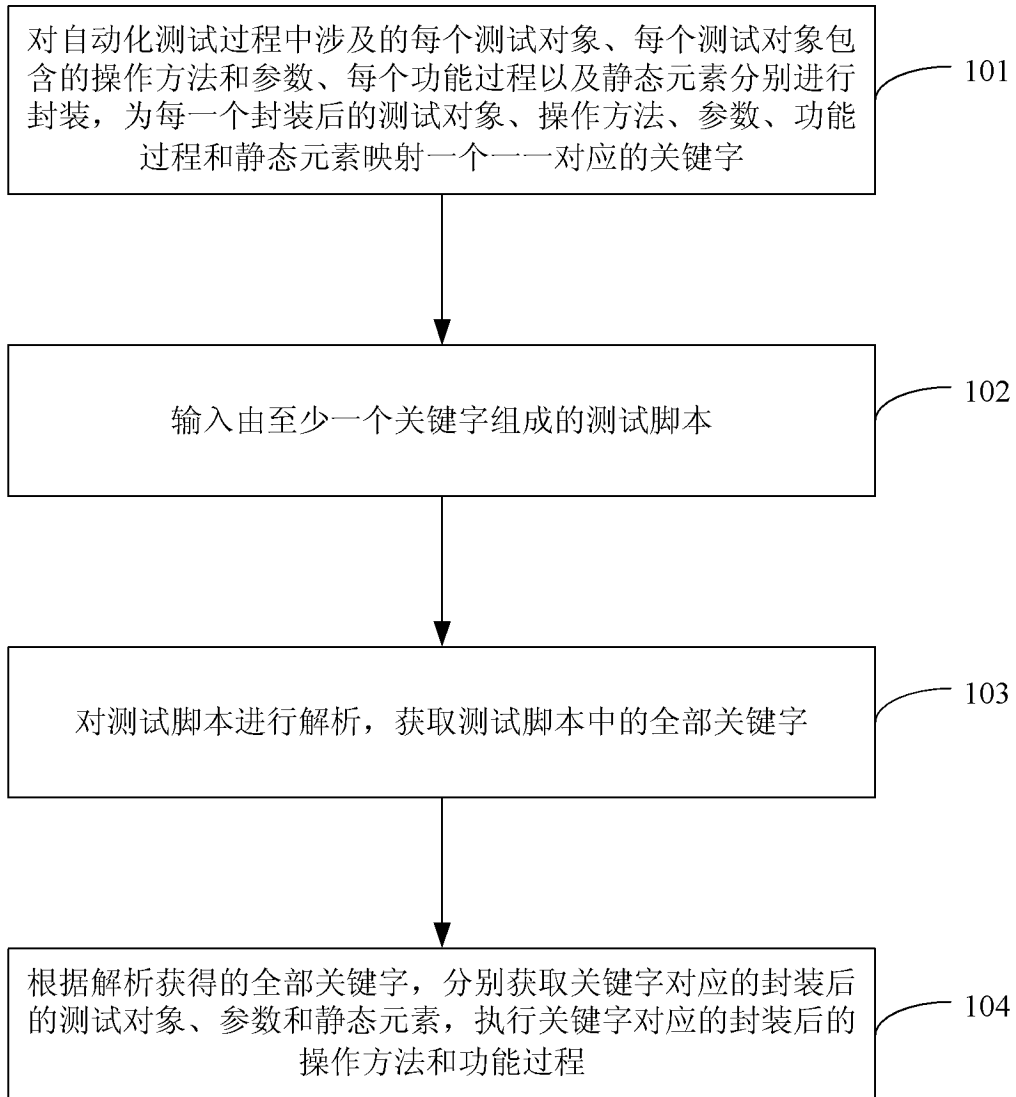


图1

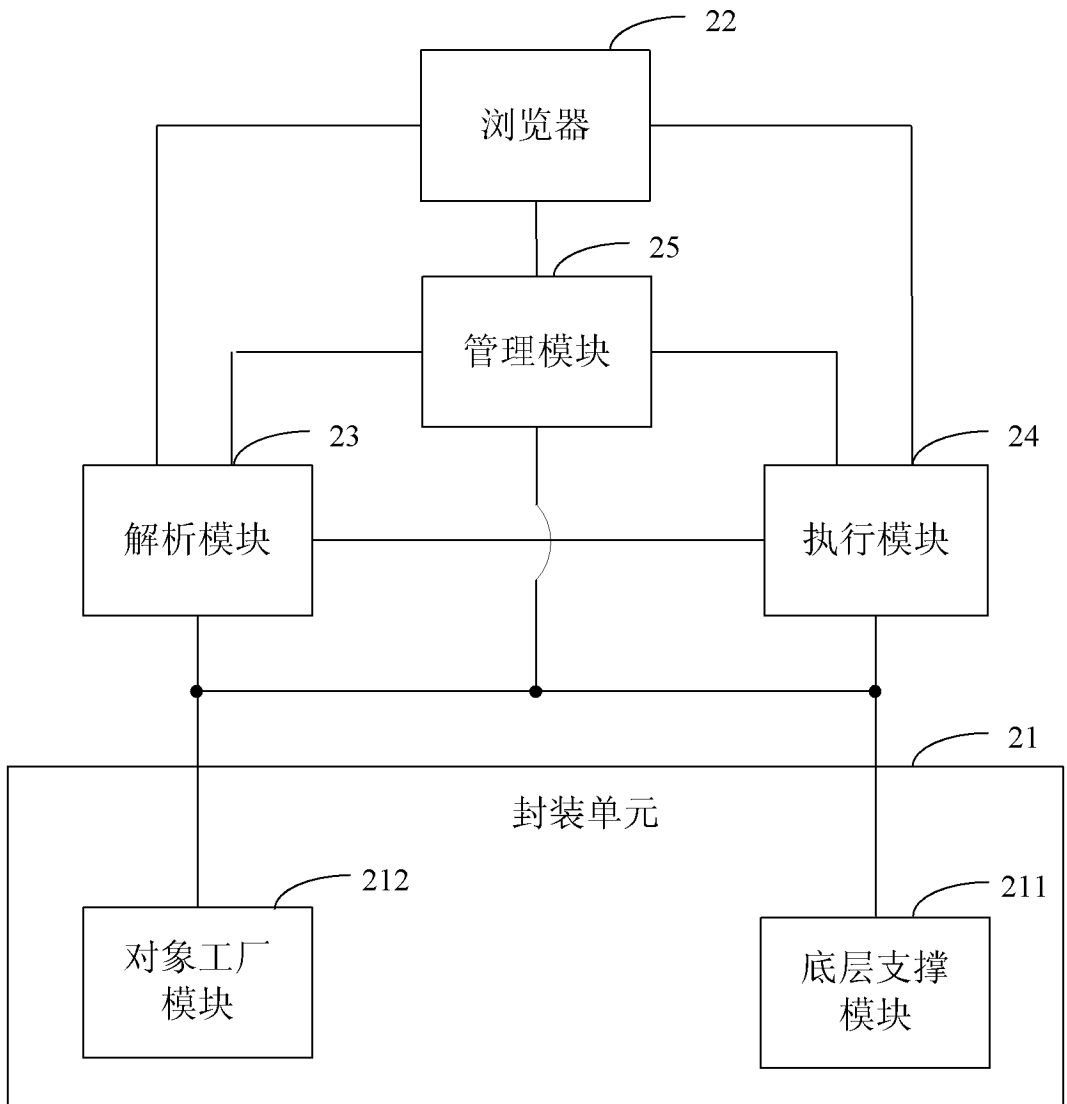


图2