



(12) 发明专利申请

(10) 申请公布号 CN 104220993 A

(43) 申请公布日 2014. 12. 17

(21) 申请号 201280072501. X

(22) 申请日 2012. 04. 20

(85) PCT国际申请进入国家阶段日

2014. 10. 20

(86) PCT国际申请的申请数据

PCT/US2012/034395 2012. 04. 20

(87) PCT国际申请的公布数据

W02013/158112 EN 2013. 10. 24

(71) 申请人 惠普发展公司, 有限责任合伙企业

地址 美国德克萨斯州

(72) 发明人 D. 利瓦伊

(74) 专利代理机构 中国专利代理(香港)有限公司 72001

代理人 谢攀 徐红燕

(51) Int. Cl.

G06F 11/36 (2006. 01)

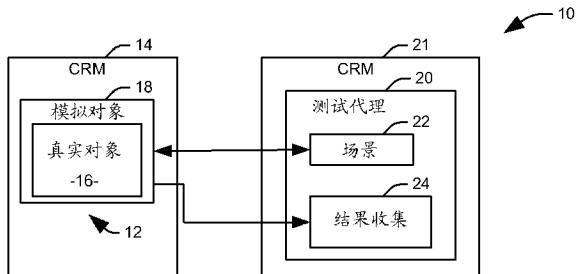
权利要求书2页 说明书6页 附图5页

(54) 发明名称

用于集成软件系统的测试系统

(57) 摘要

一种用于测试集成软件系统的系统和方法。模拟对象(18)包括加入到集成软件系统内的对应对象(16)的方面。模拟对象(18)拦截与对应对象(16)相关联的方法呼叫，并且将该方法呼叫路由到与模拟对象相关联的编程行为和对应对象之一。测试代理(20)指令该方面将方法呼叫路由到与对应对象(16)相关联的方法和与模拟对象(18)相关联的编程行为之一。



1. 一种用于集成软件系统的测试系统,包括:

被实现为第一非临时计算机可读介质上的机器可执行指令的模拟对象,所述模拟对象包括加入到集成软件系统内的对应对象的方面,以拦截与所述对应对象相关联的方法呼叫并且将所述方法呼叫路由到与所述模拟对象相关联的编程行为和所述对应对象之一;以及

测试代理,被实现为第一非临时计算机可读介质和第二非临时计算机可读介质之一上的机器可执行指令,以指令所述方法将所述方法呼叫路由到与所述模拟对象相关联的编程行为和所述对应对象之一。

2. 如权利要求 1 所述的测试系统,所述测试代理包括场景以存储针对与所述对应对象相关联的多种方法的表示与所述模拟对象相关联的编程行为的配置数据。

3. 如权利要求 2 所述的测试系统,其中所述模拟对象是多个模拟对象之一,并且所述场景包括存储针对多个模拟对象中的每个的配置数据的分层数据结构。

4. 如权利要求 1 所述的测试系统,所述测试代理还包括结果收集组件以收集来自模拟对象和对应对象之一的输入数据和输出。

5. 如权利要求 4 所述的测试系统,所述测试代理还包括场景以存储数据收集规则,所述结果收集组件使用数据收集规则来选择性地收集来自所述输入数据的输入参数和来自所述输出的输出参数值、返回值、以及出现的异常之一。

6. 如权利要求 4 所述的测试系统,所述测试代理包括多个测试代理中的一个测试代理,并且所述测试系统还包括测试服务以接收来自多个测试代理的每个中的结果收集组件的所收集的输入数据和输出,所述测试服务包括数据检查组件以查证来自模拟对象和对应对象之一的所收集的输入数据和输出具有相应的预期值。

7. 如权利要求 6 所述的测试系统,所述模拟对象包括所述测试系统中的多个模拟对象中的一个模拟对象,并且所述多个测试代理中的每个包括场景以存储表示与多个模拟对象相关联的编程行为的配置数据,所述测试服务将所述场景分布到多个测试代理中的每个。

8. 如权利要求 1 所述的测试系统,所述模拟对象是多个模拟对象之一,并且所述测试系统还包括数据绑定模型,其中多个模拟对象中的每个被链接到存储来自所述模拟对象的数据的属性包。

9. 一种存储机器可读指令的非临时计算机可读介质,所述机器可读指令当被处理器执行时,执行一种方法,包括:

生成作为分层数据对象的场景,其中表示与集成软件系统中的真实对象相关联的多种方法的定义针对模拟对象的编程行为的配置参数与相关联的方法签名相关,所述配置参数包括指令以供模拟对象将到与真实对象相关联的多种方法中的给定方法的呼叫路由到针对模拟对象的编程行为和给定方法之一;以及

将被实现为环绕针对多种方法的真实对象的方面的模拟对象注入到集成软件系统中。

10. 如权利要求 9 所述的介质,其中所述方法还包括:

接收到所述给定方法的呼叫;

执行针对所述模拟对象的编程行为和所述给定方法之一;以及

收集来自编程行为和给定方法之一的输出,所收集的输出包括由调用的模拟对象所提供的输出参数值、返回值、以及出现的异常之一,并且将所述输出提供给测试代理。

11. 如权利要求 10 所述的介质,其中所述方法还包括针对预期输出而查证所收集的输

出。

12. 如权利要求 10 所述的介质,其中执行针对模拟对象的编程行为和给定方法之一涉及具有注册订户的事件,所述方法还包括记录与所述事件相关联的发布者和订户之一。

13. 如权利要求 9 所述的介质,其中所述方法还包括生成具有与模拟对象相关联的新一组配置参数的新场景,以使得模拟对象被指令成将到与真实对象相关联的多种方法中的对应方法的呼叫路由到针对模拟对象的编程行为和所述方法中的另一个。

14. 如权利要求 9 所述的介质,其中所述方法还包括响应于与输入数据和模拟对象的调用之一相关联的事件而执行与模拟对象相关联的触发。

15. 一种用于集成软件系统的测试系统,包括:

被实现为第一非临时计算机可读介质上的机器可执行指令的模拟对象,所述模拟对象包括加入到集成软件系统内的对应对象的方面,以拦截与对应对象相关联的方法呼叫并且将所述方法呼叫路由到与模拟对象相关联的编程行为和所述对应对象之一;以及

测试代理,被实现为第一非临时计算机可读介质和第二非临时计算机可读介质之一上的机器可执行指令,以指令所述方面将所述方法呼叫路由到与模拟对象相关联的编程行为和对应对象之一,所述测试代理包括:

结果收集组件,以收集来自模拟对象和对应对象之一的输入数据和输出并将所收集的输入数据和输出提供给测试服务,所述测试服务包括数据检查组件以查证来自模拟对象和对应对象之一的所收集的输入数据和输出具有相应的预期值;以及

场景,以存储针对与模拟对象相关联的方法的集合的配置数据,所述配置数据包括针对模拟对象的数据收集规则,所述结果收集组件使用数据收集规则来选择性地收集来自所述输入数据的输入参数和来自所述输出的输出参数值、返回值、和出现的异常之一。

用于集成软件系统的测试系统

技术领域

[0001] 本发明涉及软件测试，并且更特别地涉及用于集成测试的测试系统。

背景技术

[0002] 软件测试在计算机软件的开发中起作用，并且被用来确认软件程序的质量或性能是否符合在软件的开发之前所提出的一些要求。软件测试是在软件被付诸实践之前对软件要求分析、设计规范描述和编码的检查，并且是用于保证软件质量的关键一步。软件测试是为了发现错误而执行程序的过程。软件测试可以被划分成单元测试和集成测试，其中单元测试是对软件设计的最小单元的测试，而集成测试是对整个软件系统的测试。在根据设计要求而将已经通过单元测试的相应模块组装在一起之后，执行集成测试以便找到各种接口相关的错误。

附图说明

[0003] 图 1 图示用于集成软件系统的测试系统；

图 2 图示集成软件系统中的复合应用测试系统的一个示例；

图 3 图示用于捕获针对模拟对象的期望行为以生成测试场景的记录器系统；

图 4 图示用于测试集成软件系统的一种方法；

图 5 图示用于调用复合对象的方法的一个示例；以及

图 6 是图示硬件组件的示例性系统的示意性框图。

具体实施方式

[0004] 图 1 图示了用于集成软件系统的测试系统 10。该系统包括被实现为第一非临时计算机可读介质(CRM)14 上的机器可执行指令的复合对象 12。复合对象 12 包括在本文中被称为“真实对象”的集成软件系统内的实现对象 16 和被实现为环绕(wrap around)真实对象 16 的方面的模拟对象 18。具体而言，模拟对象 18 是可以拦截所有的方法呼叫并可以将它们路由到方法的真实实现或模拟实现的动态代理器，其基于当前配置而被实现为编程行为。模拟对象 18 可以例如使用面向方面编程(AOP)框架、中间语言或字节代码编排、或代码插装而被注入到集成软件系统中。

[0005] 测试代理 20 管理复合对象 12 的上下文，其中上下文包括模拟对象 18 的虚拟状态、使用针对给定方法的模拟实现或真实实现的指令、以及针对复合对象的所收集的输入和输出数据。在图示的实现中，测试代理 20 被实现为第二非临时计算机可读介质 21 上的机器可执行指令，但将理解的是，测试代理也可以被实现在第一非临时计算机可读介质 14 上。测试代理 20 包括配置数据以指令模拟对象 18 将传入的方法呼叫路由到相应的真實实现或模拟实现。给定的模拟对象 18 可以被指令成总是使用编程行为，总是调用真实对象 16，或仅调用针对与该对象相关联的特定方法的编程行为。任何过程可以被配置为包含用于远程控制的测试代理 20。

[0006] 测试代理 20 包括场景 22 以存储表示与真实对象 16 相关联的方法的针对模拟对象 18 的配置数据。例如，场景 22 可以包括针对与模拟的真实对象相关联的每个唯一方法签名的编程的步骤集合，以响应于模拟对象的调用而对其行为进行建模。编程行为可以包括返回值、输出和参考参数值、异常(exception)抛出、事件出现、回叫执行、以及类似行为。在执行期间，模拟对象 16 参考场景 22 来确定其在相关联的方法被调用时应该如何继续进行。如果真实对象被确定为适于给定方法，则它以正常的方式被执行以产生输出。如果模拟实现是所期望的，则来自场景 22 的编程行为被执行以提供针对该方法的输出。在一个实现中，模拟对象 12 是多个模拟对象之一，并且场景 22 包括存储针对多个模拟对象中的每个的配置数据的分层数据结构。

[0007] 测试代理还可以包括结果收集组件 24 以收集被提供给复合对象 16 的输入数据和由模拟对象 16 或真实对象 18 所生成的输出。以预定的间隔，所收集的值被发送到中央测试服务(未示出)，其中它们可以针对预期值而被查证。在一个实现中，结果收集组件 24 选择性地收集输入数据和输出，以使得收集少于所有的输入数据和输出。通过选择性地收集输入和输出数据，可以提高测试系统 10 的效率。

[0008] 所图示的系统提供了许多优点。例如，开发人员能够使用他们的选择的执行框架和测试编写针对规则接口进行测试。任何接口可以被完全虚拟化，如果其尚未实现的话。可以使得编程行为完全一致。可以针对接口本地地完成测试配置，而在执行期间将远程地创建模拟对象。同样，可能的是在忙碌中切换远程对象的配置，包括在不重新启动系统的情况下从模拟到真实实现的切换。从远程模拟和真实实现执行自动地聚集的数据可以被微调以收集仅所期望且可传递的信息。最后，该系统非常适于处理负面场景，诸如绕过机制、仿真资源中断、超时、以及非预期的异常抛出。

[0009] 图 2 图示集成软件系统 50 中的复合应用测试系统的一个示例。该系统包括多个应用 52-54，每个包含对象 56-58。将理解的是，给定的应用 52-54 可以包括多个对象，但本文出于解释的简单性，以单个对象来图示每个应用 52-54。对象 56-58 可以包括模拟对象 56 以及复合对象 57 和 58。模拟对象 56 是可以被注入到系统中或在执行时间被创建的无状态代理器。模拟对象 56 表示当执行集成测试时未完成或不期望包括的系统组件。复合对象 57 和 58 包括被包裹(wrap)在相应的模拟对象方面 66 和 67 中的实现的应用组件 62 和 63。

[0010] 给定的模拟或复合对象可以包括用于对从其他系统组件被提供到模拟对象的输入数据进行接收和记录的输入数据收集器以及用于对响应于接收到的输入而被提供的输出数据进行记录的输出数据收集器。在一个实现中，每个模拟对象 56、66 和 67 可以包括许多执行前和执行后触发，以提供响应于事件而被执行的针对模拟对象的定制行为。例如，可以响应于被提供到模拟对象的输入数据、由模拟对象所生成的输出、或者与模拟对象相关联的方法的调用来执行触发。

[0011] 给定的应用 52-54 还可以包括测试代理 72-74，以虑及对与应用相关联的过程的远程控制和监视。例如，每个测试代理 72-74 可以包括存储在被称为场景的便携式数据模型中的用于模拟对象的行为的配置数据。每个场景被实现为存储表示模拟对象或多个模拟对象的行为的配置数据的其表示的分层数据结构。例如，对于每个模拟对象而言，相关联的多种方法可以利用相关联的配置数据来表示为方法步骤的集合。方法步骤的每个集合可以

与模拟类型和唯一的方法签名相关联。

[0012] 场景还包含针对其相关联的应用(例如,52)中的每个复合对象(例如,57)的配置数据,以指令复合对象执行与其真实对象相关联的行为、存储在该场景处的方法步骤、或者基于所调用的特定方法来在这两者之间进行选择。例如,给定的复合对象可以具有场景中表示的其相关联的方法中的一些但不是所有,并且场景处的编程行为可以被用于它已经针对其被定义的那些方法。对在调用每种方法时所收集的特定输入和输出数据进行支配的特定于每种方法的数据收集规则也可以被存储在该场景处。每个测试代理 72-74 还包括结果收集组件,以根据存储在该场景处的数据收集规则来收集来自模拟对象或真实对象的输入数据和输出。在一个实现中,结果收集系统被实现为其中多个对象 56-58 中的每个被链接到存储来自模拟对象的数据的属性包的数据绑定模型。因此,来自模拟对象的每个实例的数据被提供给相关联的属性包的对应实例。

[0013] 当期望时,对数据绑定模型的使用虑及对数据访问层的虚拟化,以创建仍然在其包含的数据中具有一些变化的相同类型的大量对象。作为配置大数目的模拟对象的代替,数据绑定模型将模拟对象链接到包含数据的属性包。具体而言,每个模拟对象实例的唯一模拟对象实例 id 与属性包 id 相链接。使用对象中的字段名和属性包中的属性名之间的一些命名惯例,从属性包 ID 可能的是链接数据源中的属性包并从属性包取得所有的模拟对象字段数据。将指出,几个模拟对象实例可能指向相同属性包。

[0014] 系统 50 与测试具 76 进行交互,其为相关联的用户提供了接口并且通常管理测试环境的上下文。测试具 76 可以包括由用户所选择的用于执行集成测试的任何测试框架。在所图示的系统中,测试具 76 可以远程地控制本地编程的接口,并且然后远程地对聚集的数据进行断言。测试具 76 可以操作地连接到与系统 50 相关联的测试服务客户端 77,以允许与测试服务 78 的通信。例如,测试服务客户端 77 可以提供应用程序接口,以适应所选择的测试具。用户可以通过测试具 76 向测试服务 78 提供所生成的场景,以提供用于模拟和复合对象 56-58 的行为配置。在图示的系统中,测试服务 78 保持代理 72-74 的列表,并在所有代理确认它们切换到新场景之后控制被返回到测试具 76 以供测试执行。由于各种模拟对象 56、66 和 67 是无状态代理器,所以可以在任何时间提供新场景,以完全更改模拟对象的行为或在真实实现和模拟实现之间动态地切换,甚至在测试环境是现场(live)时。每个测试代理利用所收集的数据来周期性且异步地更新测试服务 77。在一个实现中,代理可以经由超文本传输协议(HTTP)向中央测试服务发送数据。例如,在预定义的时段之后,代理发送执行所收集的数据并且询问服务是否存在要执行的任何新命令(例如,是否存在要应用的新场景)。测试服务 77 可以包括数据检查组件(未示出),以查证所收集的输入和输出数据遵循场景中所规定的规则。例如,规则可以包括预期的输入参数值、预期的输出参数值、预期的返回值、预期的呼叫数目、预期的针对各种异常的故障、以及针对方法的成功终止的核对。

[0015] 在执行期间,当调用对象 56-58 时,其可以对如何基于存储在场景处的参数来继续进行并且相应地依赖于对象是被配置成提供真实、模拟、或还是动态行为来动作而执行来自活动场景的请求指令或针对对象的真实行为。来自对象的输出和输入数据(包括输出参数、返回值、以及出现的异常)可以在代理 72-74 处被收集,其中结果被提供给测试服务 77 以供验证。将理解的是,数据可以被选择性地收集,其中仅与给定的测试相关的数据被收

集。对象还可以支持执行前和执行后触发，其是可以被编程到模拟对象 56、66 和 67 中的定制行为。这些触发可以与输入或输出数据相关联的特定事件为条件，或被简单地配置成在每次调用对象时执行。例如，对象 56–58 可以被指令成在其被调用之后睡眠给定数目的毫秒。

[0016] 图 3 图示用于捕获针对模拟对象的期望行为(包括独立的模拟对象和与复合对象相关联的方面这二者)来生成测试场景的记录器系统 80。记录器系统 80 包括记录代理器 82，其收集表征与由模拟对象所表示的模拟的真实对象相关联的方法的数据。在图示的实现中，记录器系统 80 利用流畅的应用程序接口来捕获来自一组测试配置代码的针对模拟对象的期望行为。结果所得的命令然后在动作验证器 86 处受到验证核对，以确保所确定的命令针对编程接口而言是合法的。步骤生成器 88 创建定义与模拟对象相关联的每种方法的步骤。例如，所支持的行为可以包括返回值、输入和输出参考参数值、异常抛出、事件出现、执行回叫。它也可以建立用于在运行时收集数据以供结果分析的规则以及供模拟对象建立定制行为的触发。表示一个或多个模拟对象的步骤可以被收集到作为场景的分层数据结构中以用于给定测试。

[0017] 鉴于以上在图 1–3 中描述的前述结构和功能特征，示例性方法将参考图 4 和 5 来更好地理解。虽然出于解释的简单的目的，图 4 和 5 的方法被示出并描述为连续执行，但要理解和领会的是，本发明不由图示的次序所限制，这是因为在其他示例中一些动作可以与本文所示出和描述的以不同的次序和 / 或并发地发生。

[0018] 图 4 图示用于测试集成软件系统的一种方法 150。将理解，方法 150 可以被实现为存储在一个或多个非临时计算机可读介质上的并且由相关联的(一个或多个)处理器所执行的机器可读指令。在 152 处，场景被生成为其中针对与模拟对象相关联的多种方法中的每种的配置参数与相关联的方法签名相关的分层数据对象。测试场景对在测试集成软件系统中使用的模拟对象的行为进行建模。因此，记录组件可以被用来捕获模拟对象的期望行为，并将其存储在场景中，这是所谓的复杂数据结构，其唯一地使配置涉及与模拟对象相关联的方法签名和类型。将理解，给定场景可以表示多个模拟对象。与针对模拟对象的编程行为一起，配置参数可以响应于方法的调用(与模拟对象相关联的编程行为、或真实对象处的实际方法)而包括要执行的针对与模拟对象相关联的多种方法中的每种的指令。

[0019] 在一个实现中，场景使用适当的对象创建工具来生成，诸如设计模式或流畅的应用程序接口。可以验证所确定的配置代码，以确保编程关于编程接口是正确的。例如，其可以被查证正确地规定了输入参数、输出参数和返回值，并且核对针对其可以在编译时间进行捕捉的各种错误。场景还可以定义什么信息将在运行时间针对每种方法进行收集，包括特定的输入和输出参数、返回值、呼叫的数目、以及类似的值。

[0020] 在 154 处，被实现为针对相关联的多种方法的无状态动态代理器的模拟对象被注入到集成软件系统中。例如，模拟对象可以是独立的对象或作为环绕系统中的另一对象的方面。这可以例如使用面向方面编程(AOP)框架、使用中间语言或字节代码编排、或代码插装来实现。将理解，模拟对象的无状态性质简化了以与布置、动作、断言(AAA)模型一致的方式将模拟对象注入到系统中，从而允许测试代码摆脱复杂注入的负担，使其更干净且更加可保持和可读。

[0021] 在 158 处，利用所提供的输入数据来调用真实对象和模拟对象之一。如果调用模

拟对象，则该方法使用存储在场景处的配置参数来仿真。在实践中，集成测试可以涉及被测试系统上的用例的执行，并且与模拟对象相关联的方法可以由与它们进行交互的系统中的其他组件来调用。模拟对象经由当前的上下文询问场景如何基于配置参数而继续进行并且相应地动作。作为调用的一部分，与该方法相关联的执行参数可以在结果收集组件处被更新。可以在调用真实或模拟的方法之前或在此期间收集输入数据、被调用的方法或多种方法的输出、返回值、出现的异常、以及其他此类数据中的任何或所有。在 160 处，所收集的数据根据与方法相关联的规则而被查证。例如，规则可以包括预期的输入参数值、预期的输出参数值、预期的返回值、预期的呼叫数目、预期的针对各种异常的故障、以及针对成功终止该方法的核对。在一个实现中，在与测试框架相关联的中央测试服务中的中央数据收集处执行所述查证。

[0022] 将理解的是，由于模拟对象是无状态的，所以给定模拟对象的行为可以通过利用包含不同配置数据的新场景来替换该场景而被完全改变。类似地，通过替换当前的上下文即场景、所收集的数据、要使用真实或模拟的方法的指令、以及所有的预期结果，可能的是在不具有针对重新创建或重新配置任何模拟对象的需要的情况下完全地重置测试环境。这虑及在不需要拆卸现场测试环境的情况下被测试的多种情形。

[0023] 图 5 图示了用于调用复合应用测试框架中的复合对象的方法 170 的一个示例。在 172 处，被提供给复合对象的输入数据被收集并提供给测试代理。所收集的数据可以是细粒的并被调谐成使得收集少于所有输入数据。例如，对于与给定的对象相关联的每种方法而言，可以收集特定的输入参数。通过限制所收集和查证的输入数据量，可以加快测试。在 174 处，可以自动地响应于输入数据或者响应于与输入数据或对象的调用相关联的事件来执行与对象相关联的预调用触发。预调用触发可以作为包裹代理器的一部分而被添加到对象，以表示在模拟对象被编程时所期望的定制行为。

[0024] 在 176 处，确定应调用针对复合对象的真实还是编程的行为。这可以根据与场景相关联的配置数据来确定，并且可以通过替换场景而被动态地改变。如果期望执行真实方法(Y)，则在 178 处其被执行以证明一个或多个输出。如果期望针对模拟对象的编程行为(N)，则在 180 处模拟对象请求来自场景的编程行为并且相应地动作。场景存储针对与模拟对象相关联的多种方法中的每种的编程行为，并且适当的行为可以根据针对特定的模拟类型和方法签名所存储的配置数据而被选择并提供给模拟对象。无论调用哪种方法，如果复合对象使用利用事件注册，那么特定事件的订户和发布者被记录并映射以允许对交叉组件交互的追踪和仿真。

[0025] 在 182 处，输出值(包括由调用的方法所提供的输出参数值、返回值、以及出现的异常中的任何一个)被收集并提供给集中式数据存储库以供查证。像收集输入数据一样，收集输出数据可以是细粒的并被调谐成使得收集少于所有输出数据，以使得对于与给定的模拟对象相关联的每种方法而言，可以收集特定的输出参数、返回值和异常。在 184 处，可以自动地响应于模拟对象的调用或者响应于与对象输出相关联的事件来执行与模拟对象相关联的调用后触发。像预调用触发一样，调用后触发可以被添加到复合对象，以表示在模拟对象被编程时所期望的定制行为。

[0026] 图 6 是图示了能够实现图 1-5 中所公开的方法和系统(诸如在图 1 和 2 中所图示的复合应用测试系统)的示例的硬件组件的示例性系统 200 的示意性框图。系统 200 可以

包括各种系统和子系统。系统 200 可以是个人计算机、膝上型计算机、工作站、计算机系统、器具、专用集成电路(ASIC)、服务器、刀片式服务器中心、服务器群等。

[0027] 系统 200 可以包括系统总线 202、处理单元 204、系统存储器 206、存储器设备 208 和 210、通信接口 212 (例如,网络接口)、通信链接 214、显示器 216 (例如,视频屏幕)、和输入设备 218(例如,键盘和 / 或鼠标)。系统总线 202 可以与处理单元 204 和系统存储器 206 进行通信。附加存储器设备 208 和 210(诸如硬盘驱动器、服务器、独立数据库、或其他非易失性存储器)也可以与系统总线 202 进行通信。系统总线 202 互连处理单元 204、存储器设备 206-210、通信接口 212、显示器 216、以及输入设备 218。在一些示例中,系统总线 202 还互连附加的端口(未示出),诸如通用串行总线(USB)端口。

[0028] 处理单元 204 可以是计算设备,并且可以包括专用集成电路(ASIC)。处理单元 204 执行一组指令来实现本文中所公开的示例的操作。处理单元可以包括处理核。

[0029] 附加的存储器设备 206、208 和 210 可以存储数据、程序、指令、以文本或编译形式的数据库查询、以及对于操作计算机而言可能需要的任何其他信息。存储器 206、208 和 210 可以被实现为计算机可读介质(集成或可移除),诸如存储卡、磁盘驱动器、光盘(CD)、或可通过网络访问的服务器。在某些示例中,存储器 206、208 和 210 可以包括文本、图像、视频、和 / 或音频,其部分可以对不同的人是可用的。

[0030] 另外或替代地,系统 200 可以通过可以与系统总线 202 和通信链接 214 进行通信的通信接口 212 来访问外部数据源或查询源。

[0031] 在操作中,系统 200 可以被用来实现集成软件系统中的一个或多个应用或复合应用测试系统的一个或多个部分,以供评价集成软件系统。根据某些示例,用于实现复合应用测试系统的计算机可执行逻辑驻存在系统存储器 206 和存储器设备 208、210 中的一个或多个上。处理单元 204 执行源自系统存储器 206 以及存储器设备 208 和 210 的一个或多个计算机可执行指令。如本文所使用的,术语“计算机可读介质”指代参与将指令提供给处理单元 204 以供执行的介质。

[0032] 以上已经描述的是示例。当然,不可能描述方法或组件的每个可想到组合,但是本领域普通技术人员将认识到,许多进一步的组合和置换是可能的。因此,本发明旨在涵盖落入本申请(包括所附权利要求)的范围内的所有此类更改、修改和变形。如本文所使用的,术语“包括”意指包括但不限于,术语“包括着”意指包括但不限于。术语“基于”意指至少部分地基于。此外,在公开或权利要求记载“一”、“一个”、“第一”或“另一个”元件或其等价物的地方,其应被解释为包括一个或多于一个此类元件,既不要求也不排除两个或更多此类元件。

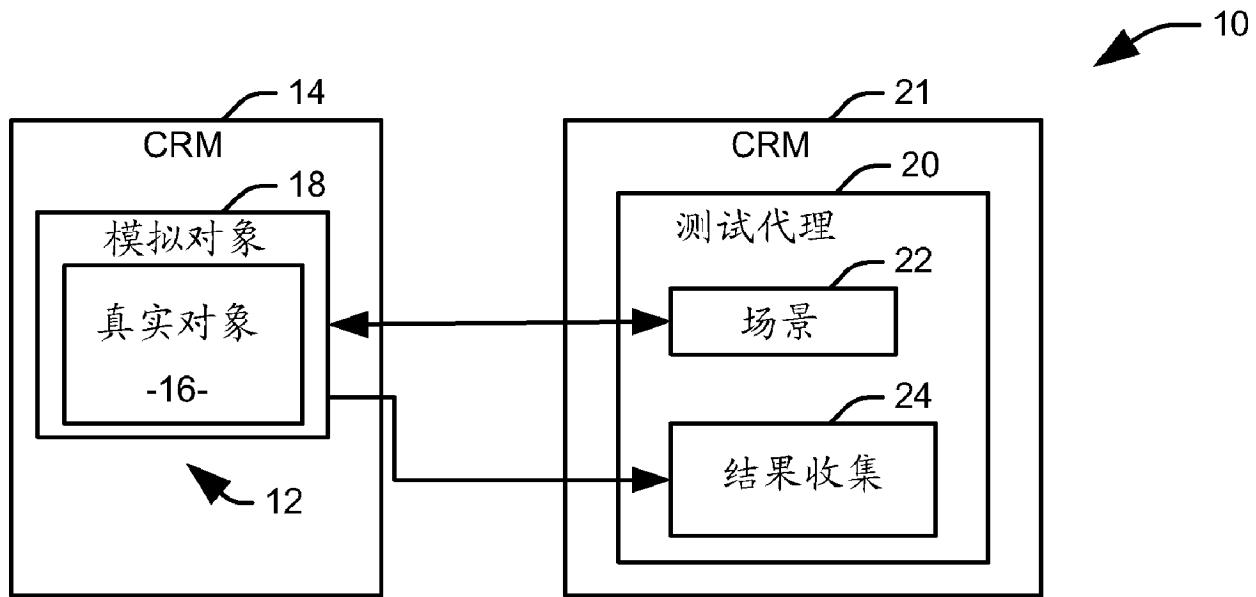


图 1

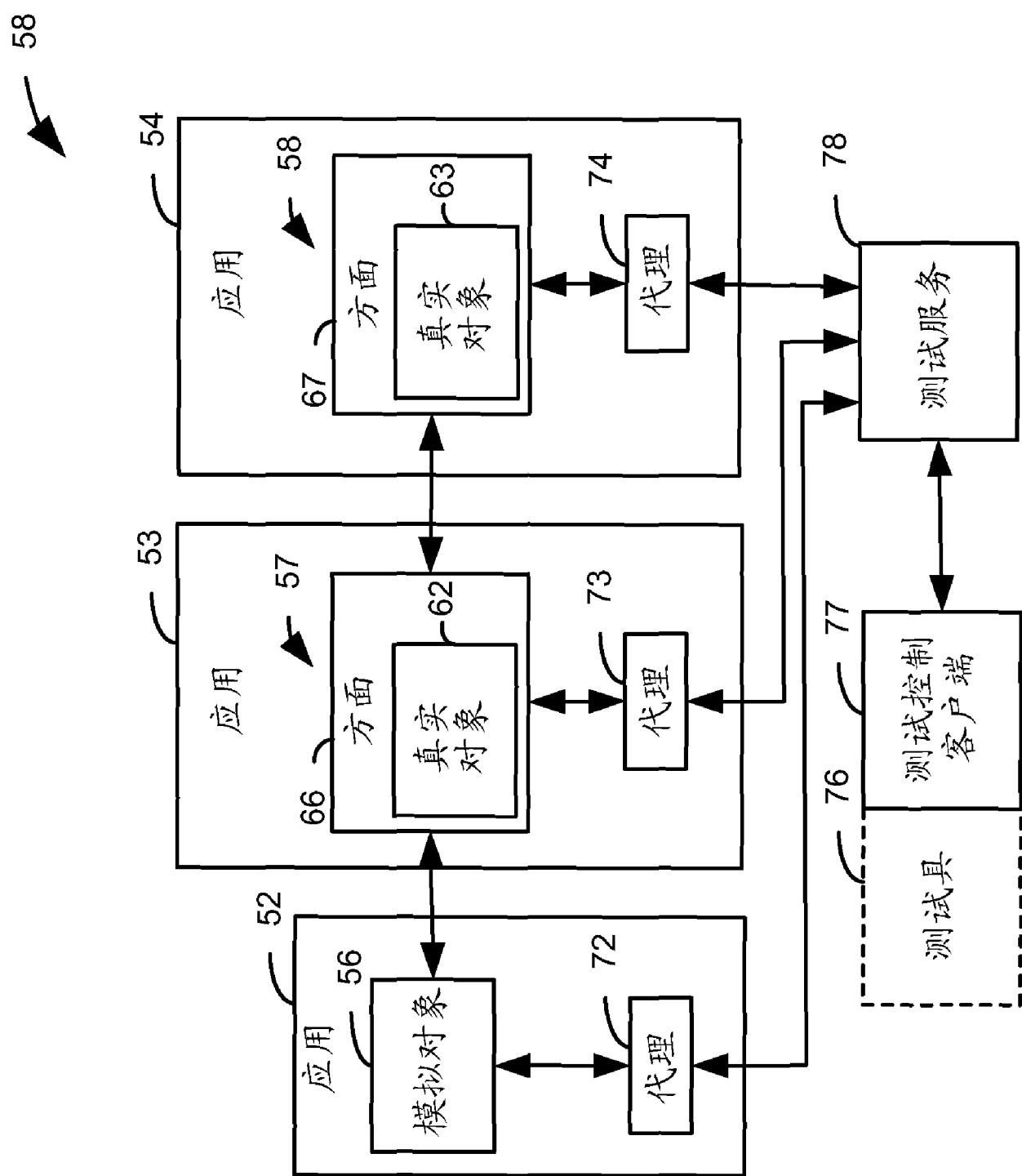


图 2

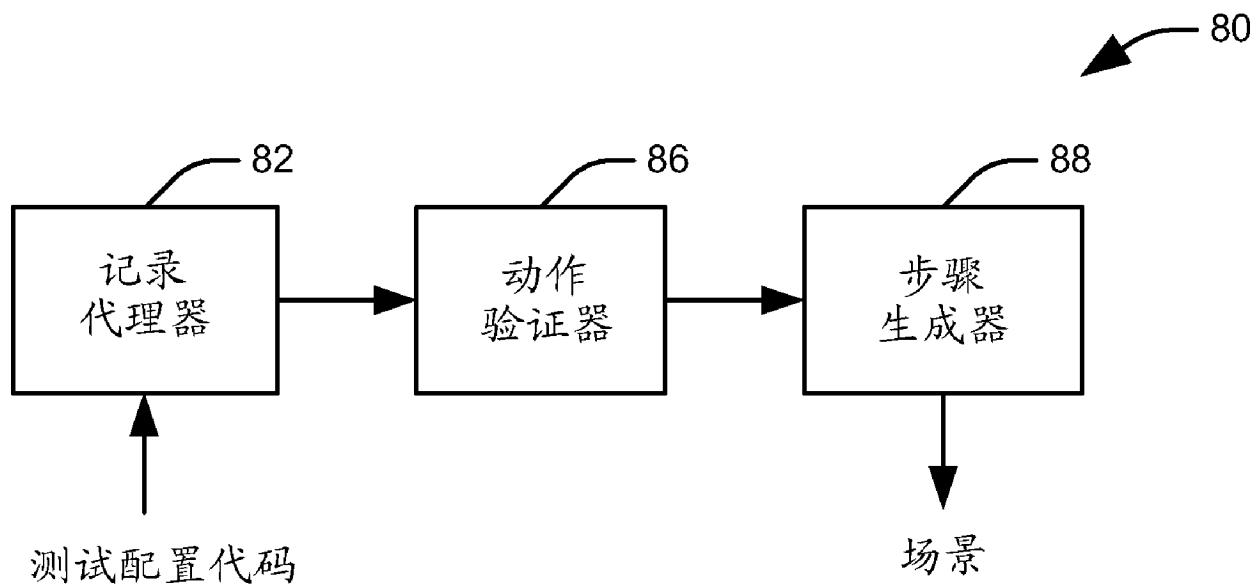


图 3

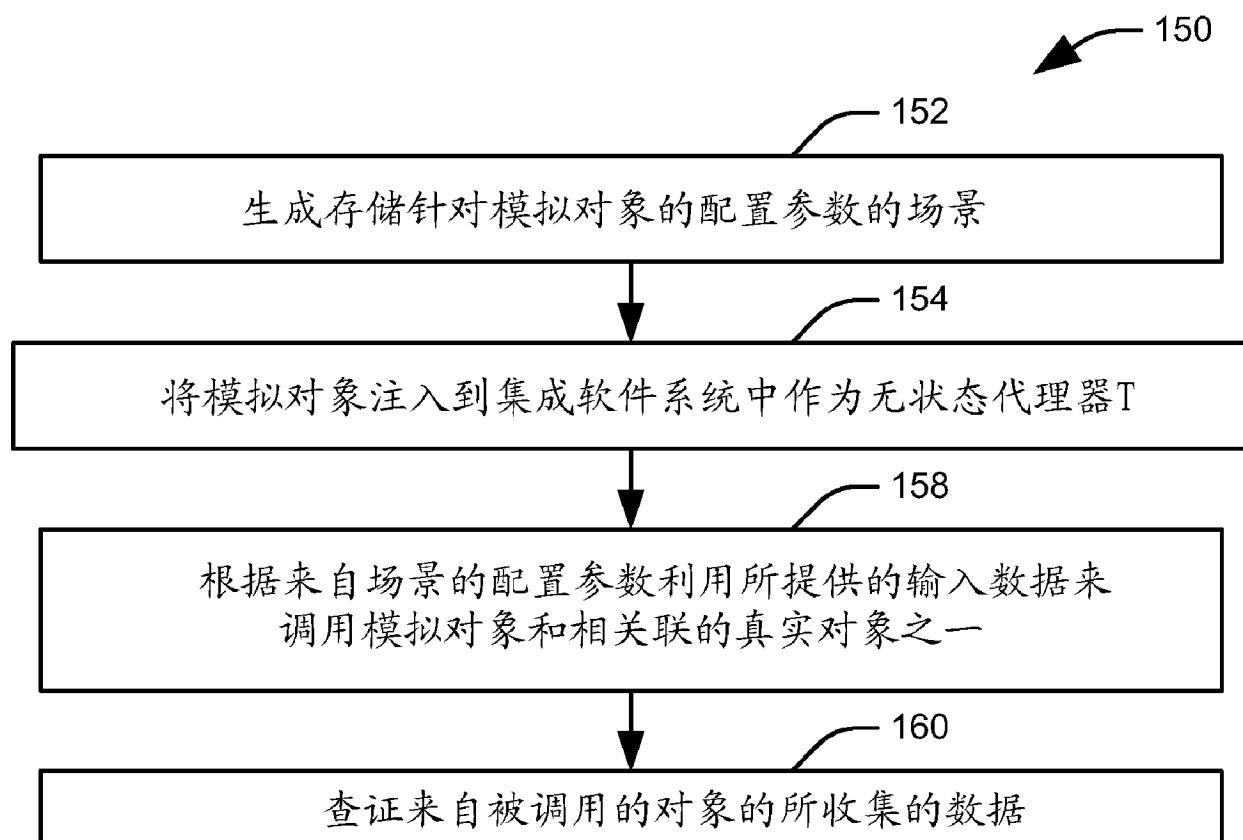


图 4

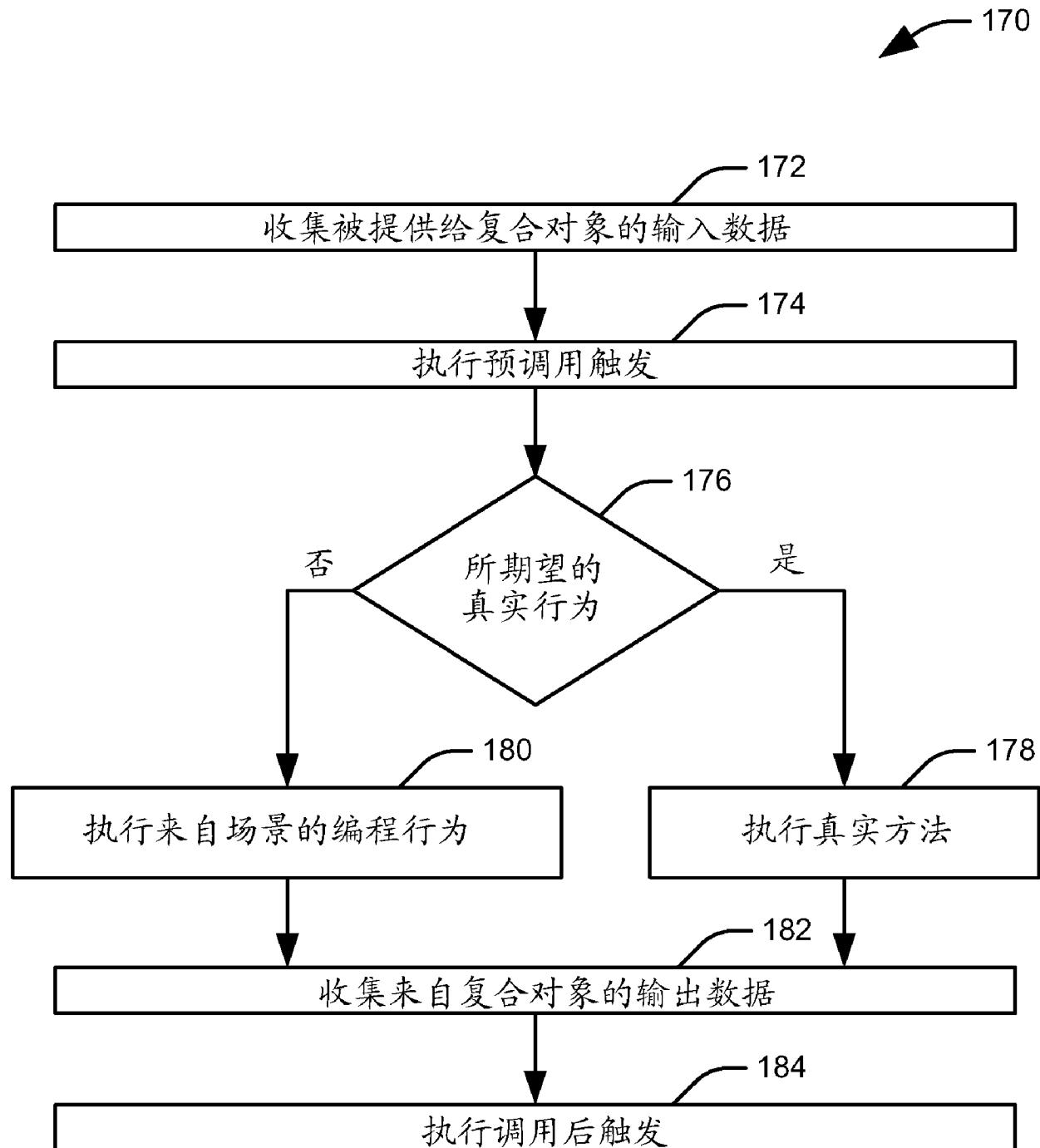


图 5

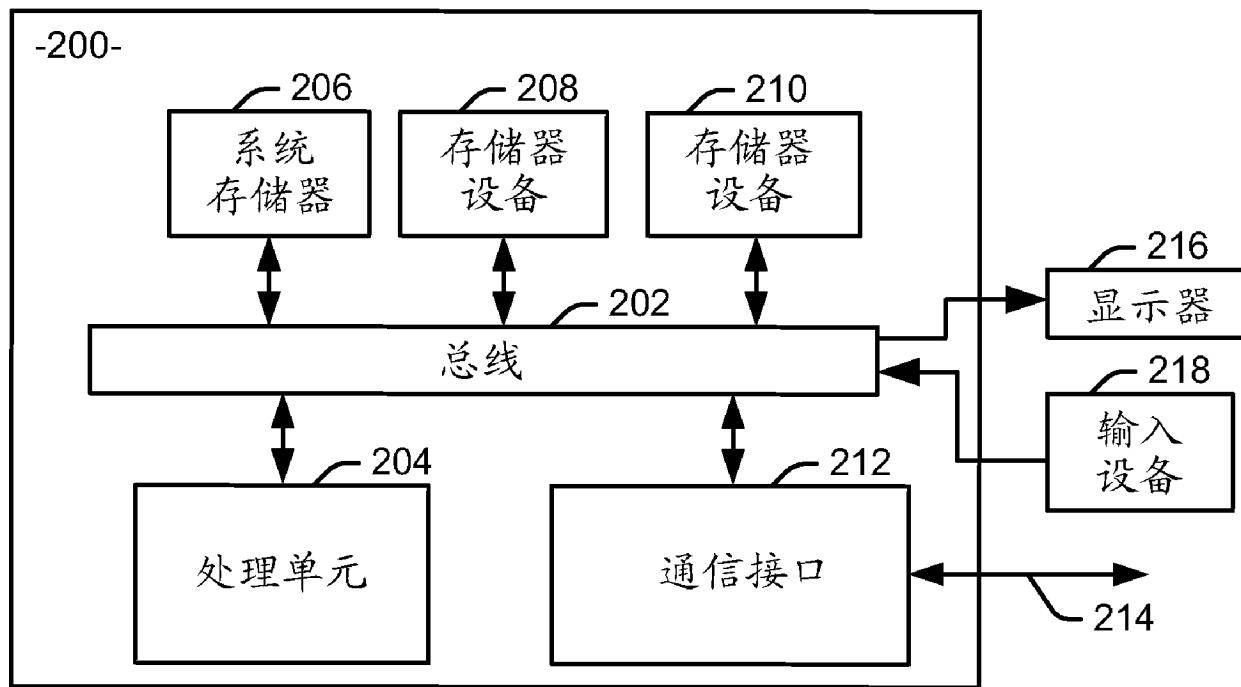


图 6